Enterprise Information Integration

New Approaches to Web Information Extraction

Patricia Jiménez

University of Sevilla, Spain

Doctoral Dissertation
Supervised by Dr. Rafael Corchuelo

September, 2015

First published in September 2015 by
The Distributed Group
ETSI Informática
Avda. de la Reina Mercedes s/n
Sevilla, 41012. SPAIN

Classification (ACM 1998): H.1.2 [User/Machine Systems] Human information processing; H.3.4 [Systems and Software] Performance evaluation; H.3.5 [Online Information Services] Web-based services; H.3.m [Information Storage and Retrieval] Miscellaneous – Data Extraction, Wrapper Generation; I.2.6 [Learning] Induction; I.5.2 [Design Methodology] Pattern analysis; I.7.5 [Document Capture] Document analysis.

Support: Supported by the European Commission (FEDER), the Spanish and the Andalusian R&D&I programmes under grants TIN2007-64119, P07-TIC-2602, P08-TIC-4100, TIN2008-04718-E, TIN2010-21744, TIN2010-09809-E, TIN2010-10811-E, TIN2010-09988-E, TIN2011-15497-E, and TIN2013-40848-R. The University of Bristol (United Kingdom) and the Institute of Information Sciences (University of Southern California, United States of America) also supported part of this work during the research visits that the author paid to these institutions.

University of Sevilla, Spain

The committee in charge of evaluating the dissertation presented by Patricia Jiménez in partial fulfillment of the requirements for the degree of Doctor of Philosophy in Software Engineering, hereby recommends ____________________ of this dissertation and awards the author the grade ____________________________________.

Miguel Toro
Catedrático de Universidad
Universidad de Sevilla

Juan Pavón
Catedrático de Universidad
Universidad Complutense

Juan M. Corchado
Catedrático de Universidad
Universidad de Salamanca

Alberto Pan Bermúdez
Profesor Contratado Doctor
Universidad de A Coruña

Pedro Szekely
Research Associate Professor
Information Science Institute

To put record where necessary, we sign minutes in ____________, ________________.

Extracting information of interest from the World Wide Web, by Luna, aged twelve.

Extracting information of interest from the World Wide Web, by Adriana, aged five.

Dedicated to my parents,
for their encouragement
and support to guide my search through life.

Contents

List of figures

List of tables

Acknowledgements

I wish to seize this opportunity to express my sincere gratitude to the people who directly or indirectly have somehow contributed to help me finish this project. First of all, to my parents, for their unconditional love and care, for encouraging me to be better and for being proud of their daughter. Thanks for letting me make my own decisions through life. To my brother and his family, for supporting me in everything that I do. I simply love them.

To my close friends, they always manage to make me disconnect from work and recharge my batteries. Especially, to my friends from El Rompido, who are like my sisters. I also have big thanks for Virginia, María, Ana, Reme, Estela, and Cristina, for supporting me and cheering me up to keep pushing during my weakest moments. I feel lucky for having you all by my side. You really mean a lot to me.

To my working partners, the TDG family. I am honoured to have worked with you and I hope to keep working with you in future. Really nice memories come up to my mind around the Geozoco room walls. I would also like to thank Arturo, my first office partner at the University of Huelva. He gave me great and valuable pieces of advice. I wish you all the best in everything that you wish to do in life.

I am also grateful for having met Dr. José L. Arjona, who trusted me to carry out this project and guided me in the early stages of this PhD project, and Dr. José L. Álvarez, who also helped me in the early stages. Many thanks, too, to the external reviewers, namely: Dr. Rafael Z. Frantz and Dr. Carlos R. Rivero, who contributed to improve this dissertation through their wise comments.

I also owe thanks to Dr. Peter A. Flach, for his positive and useful feedback during my research visit to the University of Bristol, England, and Claudio, my colleague, flatmate, and closest friend there. I spent a really good time with him, especially during our cooking time and paella Sundays.

I would also like to extend a word of thanks to the ISI staff, where I have spent the past six months. Thanks to the administrative assistants, students, researchers, and professors, for making it possible and making my stay there much more easier. Thanks for creating a great working atmosphere. Especially, I am deeply thankful to Dr. Craig A. Knoblock and Dr. Pedro Szekely, for allowing me to work with them, participate in their projects and for providing me with priceless feedback. It has been a very enriching experience. From them, I have learnt a lot and it is always a pleasure to listen the way and the enthusiasm with which they talk about their research concerns and projects.

Last and not least, I could not forget to thank my boss, Dr. Rafael Corchuelo. He hates when I introduce him as my boss, but I feel serendipitous to have such a great boss and I am proud to say so. I hope that he already knows that he is more than that, he is my friend or even a family member. I really appreciate from the bottom of my heart your generosity, patience, support, encouragement, dedication, and guidance. To not to mention your tireless effort to prepare this manuscript. Simply put, you are a role model, because you always find a way to improve and see the silver linings in every cloud. I could not be writing these lines without your help, and I really appreciate that. Thanks for trusting me.

I beg forgiveness to those who have been with me over the course of the years and whose names I have failed to mention. I feel blessed for having had you all near me, cheering me up. You do not know how important it has been to me. Thanks you all again for making it happen!

Abstract

Information has changed the lives of most people forever thanks to the advent of the Web, which has become the universally accessible distribution channel for data and has then boosted people using the Internet at an ever increasing pace. However, data themselves are not powerful; it is transforming them into information and inferring knowledge from that information using Business Intelligence techniques that makes them valuable. To do that, we need web information extractors, which are tools intended to extract data from the Web and endow them with structure and semantics so as to transform them into information that can be consumed by people or feed automated business processes to exploit them in an intelligent way.

In this dissertation, we focus on developing web information extractors that learn rules to extract information from semi-structured web documents and on how to evaluate different information extraction proposals so as to rank them automatically. We developed two proposals for web information extraction called TANGO and ROLLER; they both are based on an open catalogue of features, which eases evolving them as the Web evolves. We have also devised VENICE, an automated, open, agnostic method to rank information extraction proposals homogeneously, fairly, and stringently. Our results prove that we have advanced the state of the art with several proposals that are intended to help both researchers and practitioners.

Resumen

La manera de entender la información ha cambiado radicalmente en las últimas décadas gracias a la Web, que impulsa a las personas a hacer uso de Internet a un ritmo cada vez más vertiginoso. No es de extrañar, pues, que se haya convertido en uno de los canales de distribución de datos más usados y universalmente accesible. Sin embargo, los datos por sí solos no tienen suficiente valor; es necesario convertirlos en información a partir de la cual se pueda inferir conocimiento útil. Éste es el propósito de la inteligencia de negocio, que involucra un proceso de integración y transformación de datos en información y posterior obtención de conocimiento con el objetivo de llevar a cabo una toma de decisiones eficaz. Para que ese proceso de integración y transformación de datos tenga lugar, es necesario hacer uso de extractores de información, que son las herramientas que permiten extraer datos de la Web y dotarlos de estructura y semántica de modo que puedan ser interpretados por las personas o incorporados en procesos de negocios automáticos con el objetivo de explotarlos de una forma inteligente.

En esta tesis nos centramos en el aprendizaje de reglas para extraer información de documentos web semi-estructurados y en cómo evaluar diferentes propuestas con el objetivo de obtener un ranking de una forma totalmente automática. Nuestras dos propuestas de extracción de información son TANGO y ROLLER; ambas están basadas en un catálogo abierto de características y en técnicas inductivas. Nuestra propuesta para obtener rankings se llama VENICE; proporciona un método automático, abierto y agnóstico que esta basado en técnicas estadísticas. Esperamos que nuestras contribuciones en esta tesis puedan ser de utilidad tanto a investigadores como profesionales y que ayuden a reducir los costes en los proyectos que requieren extraer información de la Web.

Chapter 1

Introduction

This chapter introduces our PhD work, in which we present two web information extractors and an automated ranking method. It is organised as follows: in Section §1.1, we introduce the context of our research work; Section §1.2 presents an overview of the related work; Section §1.3 presents the hypothesis that has motivated our dissertation and states our thesis; Section §1.4 summarises our main contributions; Section §1.5 introduces the collaborations that we have conducted throughout the development of this dissertation; and, finally, we describe its structure in Section §1.6.

1.1 Research context

Over the past two decades, information has changed the lives of most people forever. This has been possible thanks to the advent of the Web [147], which has boosted people using the Internet at an increasing pace. It has become the universally accessible distribution channel for data. However, data themselves are not powerful; the truly revolutionary impact is how we handle them to make them a valuable resource of information that is later transformed into knowledge. Unfortunately, our incapacity as human beings to manage, analyse and exploit this overwhelming, raw, and uncategorised available and growing amount of on-line data has motivated a strong need for technology. Thus, the consumption and analysis of information by computers and computer networks has become a major global industry, which is typically referred to as Web Business Intelligence.

The goal of Web Information Extraction is to extract data from web documents and endow them with semantics so as to transform them into information that can be consumed by people or that can feed computed-based processes, e.g., Predictive Analytic, Decision Making, Data Mining, Enterprise Information Integration, Enterprise Application Integration. This field has dramatically increased our ability to infer knowledge from web documents, which results in great efficiencies for companies since they are thus able to exploit the information on the Web and make value from it. This idea is not new at all. In 1950, Zellig Harris suggested that it would make sense to reduce documents to tabular structures as a means to provide an abstract with relevant facts only. Sager [149] devised one of the earliest materialisations of Harris's ideas in the context of medical documents. With the advent of the Web in the early 90s, the problem attracted an increasing number of researchers, first in the context of the well-known Message Understanding Conference series, or MUC conferences for short, and later in the context of the SIGMOD, WWW, VLDB, and CIKM conferences, to mention a few.

In this dissertation we focus on learning rules to extract information from semi-structured web documents and also on how to evaluate different proposals so as to rank them automatically. Unfortunately, most information extraction proposals in the literature rely on learning procedures that were specifically tailored to learning ad-hoc web information extraction rules; this implies that they cannot benefit from the many research results in the general field of Machine Learning and neither can they evolve as the Web does, which in turn might make them fade away easily. Thus, none of the existing proposals is universally applicable, which has made web information extraction quite an active research field for years [42]. For instance, as of the time of writing this dissertation, Google Scholar reported on roughly 4 190 proposals on web information extraction in the last decade. Unfortunately, most of the papers in the literature regarding Information Extraction do not use a formal automated ranking method, but rather present an experimental analysis that does not disclose many details regarding the experimental environment and how the comparison was carried out.

1.2 Related work

The literature provides many pieces of related work, which can be broadly classified as information extractors, region extractors, verifiers, and repairers. Information extractors focus on helping software engineers extract information from web documents [25, 173]; region extractors focus on identifying the regions of a web document that are most likely to provide the information in which the user is interested [162]; verifiers analyse the information that is extracted in an attempt to identify when an information extractor is broken [27, 104, 105, 116, 123]; in such cases, the information extractor has to be repaired as automatically as possible [30, 126, 140, 142, 178].

In this dissertation, we focus on information extractors, which can be broadly classified into heuristic-based proposals and rule-based proposals. The heuristic-based proposals provide an algorithm that relies on a set of built-in heuristics [4, 48, 75, 154, 160]. They are not specifically tailored to the web documents to which they are applied, but have resulted from studying many web documents and concluding that there are shared patterns that help identify the information to extract. The heuristics can be fine-tuned in some cases, but they are built into the algorithms, which makes them impossible to be replaced without devising a completely new proposal. The rule-based proposals build on a generic algorithm that executes extraction rules that are specific to a web site. Such rules range from regular expressions to context-free grammars, Horn clauses, tree templates, or transducers, to mention a few. They can be handcrafted [7, 37, 71, 76, 127, 141, 150], which is a tedious and error-prone approach, learnt supervisedly [19, 21, 26, 29, 35, 53, 59, 62, 73, 82, 85, 88, 97, 107, 134, 161, 164, 167], which requires the user to provide an annotated learning set in which she or he has labelled the information to extract, or unsupervisedly [6, 9, 28, 38–40, 83, 93, 119, 121, 137, 154, 158, 160, 172, 177, 183, 183, 188], which does not require the learning set to be annotated, but requires a person to interpret the resulting rules.

Our work focuses on learning web information extraction rules. Many authors have devised related techniques that work on the text of the input documents, namely: Kushmerick and others [107] presented a proposal that learns two patterns of tokens that characterise the left and the right context of the information to extract; Hsu and Dung [85] presented a proposal that relies on using automata to model the structure of the information and regular patterns to control the transitions amongst states; Chidlovskii [29] and Muslea and others [134] also explored the idea of learning automata and patterns; Crescenzi and Mecca [38] and Crescenzi and Merialdo [40] explored learning regular expressions to extract information; Chang and Kuo [26] explored a multiple-string alignment technique; Arasu and Garcia-Molina [6] presented other proposals to learn regular expressions; and Sleiman and Corchuelo [160, 162] presented two proposals that are based on multi-string alignment techniques. There are also many authors who have devised techniques that work on the DOM tree representation of the input documents, namely: Hogue and Karger [82] presented a proposal that is based on tree similarity; Park and Barbosa [137] devised a technique that combines tree matching and clustering; Shen and Karger [152] devised a heuristic-based proposal; Álvarez and others [4] devised a proposal that relies on clustering, tree matching, string matching, and string alignment; Su and others [167] presented a proposal that is based on aligning DOM trees using a maximum entropy model; and Kayed and Chang [93] introduced a technique that first learns an information schema and then a context-free grammar using a tree similarity and a tree alignment technique. The previous techniques work on the documents themselves, that is, on their tokens or their nodes. A few authors have explored transforming the tokens or the nodes into vectors of attributive features that are related to others by means of relational features. Such a representation allows to use techniques that got inspiration from inductive logic programming. Soderland [164] and Califf and Mooney [21] pioneered this research path with two proposals that learn ground first-order rules that work on the textual representation of the input documents; Bădică and others [19] presented a technique that learns first-order rules with variables by applying the FOIL system to a first-order tree-based representation of the input documents. The previous techniques rely on quite a limited catalogue of built-in features; Freitag [62], Irmak and Suel [88], and Fernández-Villamor and others [53] worked on proposals that learn first-order rules using open catalogues of features.

The conclusion is that there are many available techniques, so ranking them is an additional problem. Unfortunately, there is not a clear taxonomy regarding methods to rank information extractors. Some authors have used informal methods in an attempt to support the idea that their proposals perform better than others in the literature, but they are very heterogeneous. There exist a few formal methods, but they just provide some foundations and guidelines [33, 81, 87, 112, 113, 115].

1.3 Research rationale

In this section, we present the hypothesis that has motivated our research work and we also state the thesis that we prove in the rest of the dissertation.

1.3.1 Hypothesis

Very frequently, the information that the Web provides is buried into semi-structured web documents. Such documents have become the standard for companies to provide catalogues of products and/or services. They are commonly generated using a template that specifies how the information that is retrieved from a back-end database regarding a user request is rendered in a human-friendly format. This makes it very difficult to extract the information that a typical web document provides automatically, which, in turn, makes it difficult to use it in typical automated business processes. Kim [95] has recently reported on the emerging trends regarding integrating web data for analysis in Business Intelligent Management and van der Meulen and Rivera [175] have highlighted the need for web data preparation as a mayor challenge to face Business Intelligence. We think that more and more companies shall rely on an increasing number of such automated business processes, which shall require more and more web data to be prepared and integrated to support them and to perform Business Intelligence. Furthermore, we think that companies should benefit from an automated method that allows them to compare the existing information extraction proposals to find the most appropriate ones for a particular purpose.

Unfortunately, even though the technologies provided by the Service-Oriented Architecture and the Semantic Web initiatives are helping cut web information integration costs down, a recent report by IBM [108] highlighted that 80% of the information on the Web is not structured, but in semi-structured or unstructured forms. Furthermore, Gartner [95] highlighted the importance of information extraction in the semantic connectivity technology trend. Another recent SIGMOD paper [34] highlighted the high costs involved in developing and maintaining information extractors.

According to the previous argumentation, we formulate this hypothesis:

> *Companies are increasingly interested in extracting information from the Web automatically so that business processes can tap into this information and perform Business Intelligence. Not only need software engineers accurate information extractors that meet their requirements, but also automated ranking methods to evaluate and compare them homogeneously, fairly, and stringently.*

1.3.2 Thesis

Many information extraction techniques that used to perform well a few years ago have faded away as the Web has evolved. The reason is that most of them are ad-hoc, that is, they rely on features of the documents that are not current and the process to analyse them is built-in. Some of them are even specific to a particular kind of layout, e.g., lists, tables or search engine results [5, 125]. Consequently, adapting them boils down to devising completely new proposals.

In the literature, there are some surveys on Information Extraction that made the previous problems evident, and most authors agree in that there is not a universal technique [60, 106, 159]. In this context, we do not think that devising new ad-hoc techniques is the right way. There exist a number of Machine Learning techniques that are applicable to a wide range of learning problems, which includes the Information Extraction field. Amongst them, we would like to highlight inductive logic programming techniques, which can naturally learn first-order rules. They are appealing insofar such rules are very expressive [47, 65, 92, 129], but, unfortunately, their learning processes are costly from a computational point of view. There are also a number of so-called propositio-relational machine-learning techniques that attempt to provide effective and efficient means to learn from relational data using propositional techniques, but they have been seldom explored in the field of web information extraction.

Unfortunately, the literature lacks automated ranking methods that allow to select an Information Extraction proposal out of the existing ones. There exist some methods, but they are not automated, open, or agnostic; neither address they key questions regarding how to set up the experimental environment, how to create evaluation splits, how to compute and cook the experimental data, or how to compute rankings and produce a report.

According to the previous argumentation, we formulate this thesis:

> *It is possible to develop general-purpose extractors that can be adapted as the Web evolves by applying inductive logic programming techniques. Furthermore, we think that it is possible to speed up*

the learning process by applying propositionalisation, which would allow us to achieve both high effectiveness and efficiency on current web documents. Finally, we think that it is also possible to provide an automated ranking method to compare and evaluate existing information extraction proposals. These contributions are expected to simplify and reduce the costs of web information extraction.

1.4 Summary of contributions

Next, we summarise the contributions we have made to prove our thesis.

TANGO: this is a system that learns first-order information extraction rules using an approach that got inspiration from several proposals in the field of inductive logic programming. The extraction rules are based on an open catalogue of features that allow to characterise not only the information that should be extracted but also the information that surrounds it, which has proven to contribute to learn more expressive rules. TANGO also relies on a number of variation points that are intended to configure it so that TANGO can reach the highest effectiveness and efficiency. They both help adapt it easily as the Web evolves.

ROLLER: this is a propositio-relational system that learns information extraction rules that are as precise and with as a high recall as TANGO's, but they are learnt in a fraction of the time required by TANGO. The approach itself is novel and differentiates from others in the literature in that it can explore an unbounded neighbourhood by means of a dynamic flattening technique that does not require any form of aggregation, and it does not explore a node in the neighbourhood unless it is proven to be good enough. Like TANGO, it uses an open catalogue of features and can leverage the continuous advances in the general field of Machine Learning.

VENICE: this is a method to evaluate, compare, and then rank Web Information Extraction proposals. Its salient features are that is it automated so that it reduces the bias that a researcher can introduce in the results; it is open so that it can easily accommodate new performance measures as they are devised and proven to be adequate in our context; it is agnostic in the sense that it does not commit to a particular kind of extractor, but has been designed to rank as many proposals as possible; it provides a clear guideline regarding how to set up the experimental environment, how to create evaluation splits, how to compute the experimental data and how to cook them, how to compute rankings, and how to report on the results.

1.5 Collaborations

During the development of this dissertation, a three-month research visit was organised at the University of Bristol (England). This visit was paid to the Machine Learning Research Group headed by Prof. Dr. Peter A. Flach, who also participated actively as a supervisor. The focus was on studying the state of the art regarding inductive logic programming algorithms, on analysing evaluation measures, and on elaborating a list of features that would help the system learn more effective rules as well as a preliminary version of prospective optimisations to speed up its learning process.

Later, a six-month research visit was organised at the Information Sciences Institute, University of Southern California (USA). This visit was paid to the Research Group headed by Prof. Dr. Craig Knoblock and supervised by Prof. Dr. Pedro Szekely. The goal was twofold: on the one hand, we wished to share our main contributions with them and get feedback from them, which definitely helped us improve the final version of this dissertation; on the other hand, we wanted to explore the field of Open Information Extraction from semi-structured web documents, which is a new, unexplored research field. Regarding the latter goal, we just attempted to put a foundation to find out how to use the techniques that we have developed in this PhD thesis to solve the problem. The idea was to explore new more general features that allow us to learn effective rules from a few web documents from several related web sites.

1.6 Structure of this dissertation

This dissertation is organised as follows:

- The introduction comprises this chapter, in which we motivate our research work and conclude that there exists a need to devise more general, expressive, adaptable, and flexible information extraction proposals that can evolve as the Web does. Furthermore, we also highlight the need for an automated method to compare information extractors.

- Chapter §2 reports on our first information extraction system. We first motivate our work, then describe the details of our proposal, and then provide an exhaustive experimental evaluation to support that we have advanced the state of the art not only conceptually, but also empirically.

- Chapter §3 describes our propositio-relational approach to learn web extraction rules. We present the motivation of our work, describe the details of our proposal, and provide an exhaustive experimental evaluation to support that we have advanced the state of the art not only conceptually, but also empirically.

- Chapter §4 reports on our automated ranking method, which encompasses a number of steps to make decisions regarding what the best information extractor is out of a number of alternatives. We first introduce the problem to solve, then describe the details of our proposal, and finally show how the method works in practice.

- Chapter §5 concludes this dissertation. It summarises our key findings and sketches some future work towards unsupervised Open Information Extraction.

Chapter 2

TANGO: an inductive logic learner

Here, we describe TANGO, which is an inductive logic programming approach to learn web information extraction rules. It is organised as follows: Section §2.1 presents our motivation and sketches our system; Section §2.2 describes the details of our proposal; Section §2.3 reports on how we have configured it so that it can achieve its best results; then, the results of our experimental analysis are presented in Section §2.4; Section §2.5 presents the related work and a detailed comparison with our proposal; Section §2.6 summarises our conclusions. Appendices §A and §B report, respectively, on our experimental environment and the performance measures that we have used.

2.1 Introduction

Most of the information extraction proposals in the literature have faded away or are fading away as the Web is evolving. The reason is that they were designed around some features of the input documents that are analysed using a number of procedures that implement ad-hoc Machine Learning techniques. The problem with such approaches is that they start failing when the Web evolves since the assumptions on which they rely are then likely to get broken. For instance, some years ago it was quite usual to use a variety of HTML tags with many style attributes; nowadays, the most common tags are div or span and the styles are injected by means of CSS rules; this implies that a not-very-old technique that heavily relies on the plain HTML representation of web documents is very likely to fail nowadays. Unfortunately, existing proposals are not flexible enough to be adapted easily.

The solution is simple to state: use an open catalogue of features, that is, a catalogue in which the features are black boxes that can be easily replaced, and identify a number of variation points, that is, the procedures that implement heuristics for which several alternatives that may have an impact on performance and/or effectiveness exists. A system that relies on an open catalogue of features and variation points is inherently easier to adapt than a system that relies on a closed catalogue of features and built-in procedures that implement ad-hoc Machine Learning techniques. Unfortunately, only a few authors have explored the idea of using an open catalogue of features [19, 53, 59, 62, 88] and no-one has ever explored the idea of identifying the variation points.

In this chapter, we present TANGO, which is a new proposal to learn information extraction rules. Its salient features are that it relies on an open catalogue of features and a number of variation points. The catalogue being open means that our proposal does not rely on built-in features to represent the input web documents, but they are provided by the user and can then change and evolve as needed; our current catalogue includes HTML, DOM, rendering, user-defined, and relational features that have proven to characterise well current web documents, but it can be easily replaced because nothing in our proposal depends on the catalogue providing a specific feature. The variation points are procedures that encapsulate heuristics that are intended to guide the search for good rules as effectively and efficiently as possible; we have devised several alternatives for each heuristic, but our proposal is open to try different ones if required. We have devised a method that allows to assess which configuration of the variation points is the best regarding both effectiveness and efficiency; we have used that method to configure our proposal so that it can beat others in the literature. Note that the previous characteristics clearly make TANGO deviate from other proposals in the literature, most of which are closed in the sense that they rely on a closed catalogue of features and built-in procedures to analyse them. We have conducted an extensive experimental analysis that proves that our proposal outperforms other state-of-the-art proposals regarding effectiveness. Regarding efficiency, it has proven to be practical, but not the most efficient one. The conclusions that we have drawn from our experimental analysis have been confirmed using standard statistical hypothesis tests in the literature.

2.2 Description of our proposal

TANGO works on a set of documents that are represented using DOM trees and an annotation. The documents provide examples of how the information to extract is encoded and the annotation assigns each of their nodes to a slot that classifies the information that it provides. (The nodes to be extracted are referred as positive examples, and the nodes to be ignored as negative examples.) The documents are assumed to provide information on a given topic and to have regularities that help learn the rule. TANGO creates a learning set that consists of a ground first-order representation of the input documents and the annotation. It then uses a top-down covering algorithm that learns a rule set for each slot. It starts with an overly-general rule that matches every node in the learning set and then extends it by adding conditions that constraint the subset of nodes that it matches; when a rule that matches positive examples only is found, it is considered a solution; the positive examples that it matches are then removed from the learning set and the procedure is re-started until no positive example remains in the learning set or it is not possible to find a rule, which is very unlikely in practice. Our proposal also manages a set of savepoints to which it can backtrack if the current search path is not good enough.

In the following subsections, we first present some preliminaries, then introduce the main procedure, and then present the procedures to learn a rule set and to learn a rule; we also describe some ancillary procedures that deal with computing the conditions that can possibly be used to extend a rule and with managing savepoints. To make a distinction amongst the procedures for which TANGO provides a unique implementation and the variation points for which there are several choices, we typeset the names of the latter using SMALL CAPITALS.

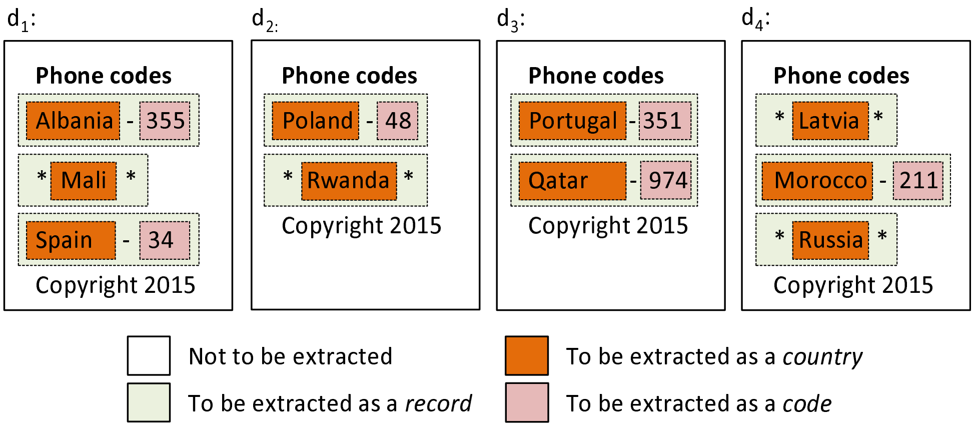


Figure 2.1: *Sample documents.*

2.2.1 Preliminaries

Next, we present the mathematical notation that we use and then define and illustrate the concepts on which our proposal relies.

Definition 2.1 (Mathematical notation) *We use the standard mathematical notation to represent variables, sets, logical formulae, and the like. We would like to highlight only a few pieces of notation for which we have not found a standard in the literature, namely: given a set of elements* $\{x_1, x_2, \ldots, x_n\}$*, then* $\langle x_1, x_2, \ldots, x_n \rangle$ *denotes a sequence of them; given a sequence* s*, we denote its number of elements as* $|s|$*; given two sequences* s_1 *and* s_2*, we denote their concatenation as* $s_1 \oplus s_2$*.*

Definition 2.2 (Documents) *A document is a character string that adheres to the HTML syntax and can then be represented as the root node of the corresponding DOM tree [80, 176].*

Example 2.1 *Figure §2.1 shows a collection of four documents that provide listings of phone codes for several countries, if available; countries for which the system does not have a phone code are starred. Figure §2.2 shows document* d_1 *as a DOM tree whose root is node* n_1*.*

Definition 2.3 (Features) *Features are functions that map nodes onto values or other nodes. The former are referred to as attributive features, and*

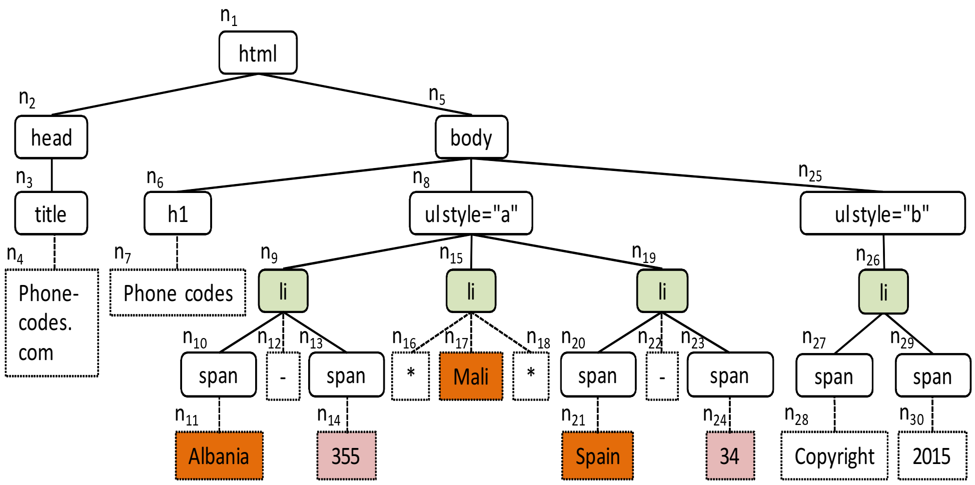


Figure 2.2: *Sample DOM tree.*

they can be based on HTML attributes [80], DOM attributes [176], rendering attributes [17], or user-defined functions; the latter are relational features and they build on the usual relationships amongst the nodes of a DOM tree, e.g., parents, children, siblings, and the like. Note that we do not expect every feature to be instantiatable on every node.

Example 2.2 *Table §2.1 illustrates the instantiation of some of the features of the nodes of which the document in Figure §2.2 is composed. Column* node *represents the node being examined; columns* tag *and* style *represent its HTML tag and its CSS style, respectively; columns* depth *and* children *represent its depth and the number of children it has in the DOM tree, respectively; columns* ypos *and* xpos *represent the ordinate and the abscissa of the corresponding rendering box, respectively; columns* len *and* isnumber *represent the number of tokens in the text that is associated with the node and whether it is a number or not, respectively; columns* parent *and* left *represent the corresponding relationships amongst nodes in the DOM tree. A blank cell means that the corresponding feature cannot be instantiated on the corresponding node. For instance, node* n_1 *does not have a parent and node* n_6 *is an* h1 *node that does not have an explicit CSS style.*

Definition 2.4 (Slots and annotations) *A slot is a label that provides a meaning to the information that is contained in a node. An annotation is a function that maps a subset of nodes onto a set of slots. We assume that the slots may*

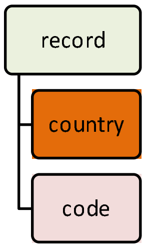
	Attributive features								Relational features	
	HTML		DOM		Rendering		User-defined			
node	tag	style	depth	children	ypos	xpos	len	isnumber	parent	left
n_1	html		1	2	0	0	18	false		
n_2	head		2	1	0	0	5	false	n_1	
n_3	title		3	1	0	0	5	false	n_2	
n_4			4	0	0	0	5	false	n_3	
n_5	body		2	3	0	0	13	false	n_1	n_2
n_6	h1		3	1	0	0	2	false	n_5	
n_7			4	0	0	0	2	false	n_6	
n_8	ul	a	3	3	16	0	9	false	n_5	n_6
n_9	li		4	3	16	0	3	false	n_8	
n_{10}	span		5	1	16	0	1	false	n_9	
n_{11}			6	0	16	0	1	false	n_{10}	
n_{12}			5	0	16	32	1	false	n_9	n_{10}
n_{13}	span		5	1	16	36	1	true	n_9	n_{12}
n_{14}			6	0	16	36	1	true	n_{13}	
n_{15}	li		4	3	32	0	3	false	n_8	n_9
n_{16}			5	0	32	0	1	false	n_{15}	
n_{17}			5	0	32	20	1	false	n_{15}	n_{16}
n_{18}			5	0	32	40	1	false	n_{15}	n_{17}
n_{19}	li		4	3	48	0	3	false	n_8	n_{15}
n_{20}	span		5	1	48	0	1	false	n_{19}	
n_{21}			6	0	48	0	1	false	n_{20}	
n_{22}			5	0	48	32	1	false	n_{19}	n_{20}
n_{23}	span		5	1	48	36	1	true	n_{19}	n_{22}
n_{24}			6	0	48	36	1	true	n_{23}	
n_{25}	ul	b	3	1	64	0	2	false	n_5	n_8
n_{26}	li		4	2	64	0	2	false	n_{25}	
n_{27}	span		5	1	64	0	1	false	n_{26}	
n_{28}			6	0	64	0	1	false	n_{27}	
n_{29}	span		5	1	64	36	1	true	n_{26}	n_{27}
n_{30}			6	0	64	36	1	true	n_{29}	

Table 2.1: *Sample feature instantiation.*

be organised hierarchically so that there is a first-level slot that contains some nested slots. We do not require every node to be mapped by an annotation; intuitively, nodes that are not mapped by an annotation are not expected to be extracted. (It is common to use term slot to refer to either a label or a node that is extracted with that label, but this should not be a problem.)

Example 2.3 *Table §2.2 illustrates the annotation of the document in Figure §2.2. We use the following slots:* `record`*, which refers to the records to be*

Structure of slots



Annotation		Annotation		Annotation	
node	slot	node	slot	node	slot
n_1		n_{11}	country	n_{21}	country
n_2		n_{12}		n_{22}	
n_3		n_{13}		n_{23}	
n_4		n_{14}	code	n_{24}	code
n_5		n_{15}	record	n_{25}	
n_6		n_{16}		n_{26}	
n_7		n_{17}	country	n_{27}	
n_8		n_{18}		n_{28}	
n_9	record	n_{19}	record	n_{29}	
n_{10}		n_{20}		n_{30}	

Table 2.2: *Sample annotation.*

% Annotation

record(n_9).
record(n_{15}).
record(n_{19}).
country(n_{11}).
country(n_{17}).
country(n_{21}).
code(n_{14}).
code(n_{24}).

% Attributive features

tag(n_1, 'html').
tag(n_2, 'head').
...
tag(n_{27}, 'span').
tag(n_{29}, 'span').

depth(n_1, 1).
depth(n_2, 2).
...
depth(n_{29}, 5).
depth(n_{30}, 6).

children(n_1, 2).
children(n_2, 1).
...
children(n_{29}, 1).
children(n_{30}, 0).

style(n_8, 'a').
style(n_{25}, 'b').

ypos(n_1, 0).
ypos(n_2, 0).
...
ypos(n_{29}, 64).
ypos(n_{30}, 64).

xpos(n_1, 0).
xpos(n_2, 0).
...
xpos(n_{29}, 36).
xpos(n_{30}, 36).

len(n_1, 18).
len(n_2, 5).
...
len(n_{29}, 1).
len(n_{30}, 1).

isnumber(n_{13}).
isnumber(n_{14}).
isnumber(n_{23}).
isnumber(n_{24}).
isnumber(n_{29}).
isnumber(n_{30}).

% Relational features

parent(n_2, n_1).
parent(n_3, n_2).
...
parent(n_{29}, n_{26}).
parent(n_{30}, n_{29}).

left(n_5, n_2).
left(n_8, n_6).
...
left(n_{25}, n_8).
left(n_{29}, n_{27}).

Table 2.3: *Sample dataset.*

extracted, country, *which refers to the names of the countries, and* code, *which refers to their phone codes (if available); slots* country *and* phone *are hierarchically nested into first-level slot* record. *Note that some cells are blank, which means that the corresponding nodes are not intended to be extracted.*

Definition 2.5 (Datasets) *A dataset is a ground first-order representation of an annotation and the instantiation of a catalogue of features on a set of documents. The datasets that are used to learn rules are referred to as learning sets and the datasets that are used to test rules are referred to as test sets. Note that there is not a structural difference between them; the difference is regarding how they are used.*

Example 2.4 *Table §2.3 shows an excerpt of the dataset that corresponds to the annotation in Table §2.2 and the feature instantiation in Table §2.1. It is organised into three sections, namely: the first one is a representation of our sample annotation; then comes the representation of the instantiation of the attributive features; finally, there is the representation of the relational features. Note that Boolean features are represented compactly. For instance, we use fact* $\mathtt{isnumber}(n_{14})$ *to indicate that node* n_{14} *is a number, instead of a fact of the form* $\mathtt{isnumber}(n_{14}, \mathtt{true})$*; thanks to this compact notation, it is not necessary to make it explicit the cases in which the feature returns* `false`*.*

Definition 2.6 (Rules and conditions) *A rule consists in a number of conditions that characterise the nodes that provide the information to be extracted as accurately as possible. We represent the rules using Horn clauses of the form* $h :- b_1, b_2, \ldots, b_n$ $(n \geq 0)$*, where the head is a slot instantiator and the body consists of feature instantiators, comparators, and/or further slot instantiators (if recursion is allowed). A slot instantiator is a condition of the form* $s(N)$*, where* s *denotes the kind of slot that we wish to extract and* N *is a variable that can be bound to every node in the input documents. A feature instantiator is a condition that binds the value of a feature on a node to a constant or a variable; feature instantiators can be negated, in which case the condition is satisfied if the corresponding feature cannot be instantiated. A comparator is a condition that compares a variable to another variable or a constant using the usual relational operators. A condition is said to be determinate in the context of a rule if it is a feature instantiator, it can be instantiated exactly once on every positive example that is matched by the rule, at most once on every negative example, and does not return the same value on every example. Note that neither negated conditions, nor Boolean feature instantiators, nor comparators are considered determinate; note, too, that a condition is not determinate or indeterminate per se, but in the context of a rule. In our algorithms, we represent a rule of the form* $h :- b_1, b_2, \ldots, b_n$ *as a sequence of conditions* $\langle h, b_1, b_2, \ldots, b_n \rangle$ $(n \geq 0)$*.*

Example 2.5 *Below, we present a very simple rule that extracts nodes that belong to slot* `record`*:*

$$\langle \mathtt{record}(N_0), \mathtt{tag}(N_0, A_1), A_1 = \text{'li'}, \mathtt{parent}(N_0, N_1), \mathtt{style}(N_1, A_2), A_2 = \text{'a'} \rangle$$

Simply put, the head is a slot instantiator of the form $\mathtt{record}(N_0)$ *that indicates that it is a rule to extract record slots. Variable* N_0 *can be bound to any node in the input document that fulfils the conditions in the body. The first two conditions state that the nodes to which* N_0 *is bound must have tag* $\mathtt{li}$*; the last three conditions state that they must also have a parent with style* $\mathtt{a}$*.*

All of the feature instantiators were found to be determinate when they were added to the rule. For instance, condition $\mathtt{tag}(N_0, A_1)$ *was added in the context of the initial rule* $\langle\mathtt{record}(N_0)\rangle$*; in this rule, variable* N_0 *can be bound to any of the nodes in the input documents; furthermore* $\mathtt{tag}(N_0, A_1)$ *can be instantiated only once on the nodes that correspond to positive examples because they are element nodes and they have a unique tag; contrarily, it can be instantiated at most once on the remaining nodes because text nodes do not have a tag; furthermore, not every node has the same tag. A similar reasoning can be straightforwardly applied to* $\mathtt{parent}(N_0, N_1)$ *and* $\mathtt{style}(N_1, A_2)$ *to prove that they were determinate in the context of the rule to which they were added.*

Definition 2.7 (Scores and gains) *We require a rule scorer to assesses how good a rule is. Intuitively, it must return high scores for rules that are close to be a solution and low scores for the others. Since our proposal learns rules by adding conditions incrementally, it is also necessary to compute the gain that adding a specific condition achieves. If* r *represents the current rule and* r' *represents the rule that results from adding a given condition* c *to* r*, that is,* $r' = r \oplus \langle c\rangle$*, then we compute the gain of condition* c *as* $p'(s' - s)$*, where* p' *denotes the number of positive examples matched by rule* r'*,* s' *is the score of rule* r'*, and* s *is the score of rule* r*. Realise that we weight the difference of scores with the number of positive examples matched by* r'*, which helps make a difference that rewards the conditions that match the largest possible number of positive examples.*

Example 2.6 *In the sequel, we use a rule scorer that is based on the well-known Information Content function [139]. This function is defined as* $-\log_2 P$*, where* P *denotes the precision of the rule on which the function is computed; it then ranges in interval* $[0.00, +\infty)$ *so that the closer to* 0.00*, the better. To use it as a rule scorer within the context of our proposal, we simply have to negate it, so that high scores correspond to good rules and low scores correspond to bad rules.*

As an example, consider the following initial rule:

$\langle\mathtt{record}(N_0)\rangle$

```
method $\mathrm{TANGO}(documents, annotation)$
    – Step 1: initialisation.
    $result = \emptyset$
    $dataset$ = create a dataset from $documents$ and $annotation$
    – Step 2: learn a rule set for every slot.
    for each different $slot$ in $annotation$ do
        $learningSet$ = create a learning set for $slot$ from $dataset$
        $learningSet = \textsc{preProcessLearningSet}(learningSet, slot)$
        $ruleset = learnRuleSet(learningSet, slot)$
        $result = result \cup \{ruleSet\}$
    end
return $result$
```

Figure 2.3: *TANGO's main procedure.*

It matches the 3 *positive examples and the* 27 *negative examples in the DOM tree in Figure* §2.2*; thus its score is* $\log_2 3/(3+27) = -3.32$*. If condition* $tag(N_0, A_1)$ *is added to the rule, then it becomes the following one:*

$\langle record(N_0), tag(N_0, A_1)\rangle$

This rule matches the 3 *positive examples in our running example, because feature* tag *can be instantiated on any of our positive examples, and* 14 *negative examples, because there are* 27 *such examples, but this feature cannot be instantiated on any of the* 13 *text nodes. Thus, it scores at* $\log_2 3/(3+14) = -2.50$*. The gain that adding condition* $tag(N_0, A_1)$ *to the initial rule is then computed as* $3\,(-2.50 + 3.32) = 2.46$*.*

2.2.2 The main procedure

Figure §2.3 shows TANGO's main procedure, which works on a set of documents and an annotation; it returns a set of rule sets, each of which is specifically tailored to extracting information that belongs to a given slot.

The first step consists in initialising the result to an empty set and then creating a dataset from the input documents and the annotation. Basically, we have to loop through a user-provided catalogue of features and try to instantiate them on every node of the input documents. This procedure is intricate from a technical point of view since it requires to parse the input documents, to render them, and then compute the features, but it is very simple from a conceptual point of view, which is the reason why we do not delve into additional details.

The second step iterates through the set of slots used in the annotation of the input documents. For each slot, it first creates a learning set from the previous dataset, pre-processes it, and then invokes the procedure to learn a rule set; the result is stored in the result variable, which is returned when the loop finishes. Creating the learning set amounts to creating a new dataset in which the positive examples are the nodes that belong to the slot that is being analysed and the negative examples are the remaining nodes. In order to reduce the computational effort, the learning sets that correspond to first-level slots have information about every node in the input documents; contrarily, the learning sets that correspond to the nested slots have information about the nodes in the enclosing slots only. This makes sense because we need to make a global decision regarding every node in the input documents that corresponds to a first-level slot; when they are extracted, the rules to extract their nested slots must be applied to the DOM sub-trees that are rooted in the nodes that have been extracted. The resulting learning set must be pre-processed using a variation point called PREPROCESSLEARNINGSET. The reason is that there are some alternatives to transform them that might result in equivalent datasets from which learning is more effective or efficient. Unfortunately, it is not clear if these pre-processing steps are worth or not, which justifies implementing them as a variation point.

Example 2.7 *To illustrate TANGO's main procedure, we focus on the sample document whose DOM tree is shown in Figure §2.2. The instantiation of the sample feature catalogue that we are going to use to illustrate our proposal is presented in Table §2.1 and the corresponding annotation is presented in Table §2.2; an excerpt of the dataset from which the learning sets are created is presented in Table §2.3.*

Regarding slot `record`*, TANGO first creates a learning set that is partially illustrated in Table §2.4. Note that the difference with the dataset is that we keep the nodes that belong to slot* `record` *as positive examples and make it explicit that the other nodes are negative examples; the instantiation of the features remains the same. The reason why we have to make the negative examples explicit is that rule scorers assess how good a rule is building on confusion matrices, which cannot be computed unless the negative examples are made explicit. In this simple example it is not actually necessary to perform any additional pre-processing because the total number of nodes and features is very small. From this learning set, TANGO can learn the following rule set, which specifies that node* N_0 *must be extracted as a record if it has tag* `li` *and its parent has style* `a`*:*

% Positive examples

record(n_9).
record(n_{15}).
record(n_{19}).

% Negative examples

¬ record(n_1).
...
¬ record(n_8).
¬ record(n_{10}).
...
¬ record(n_{14}).
¬ record(n_{16}).
...
¬ record(n_{18}).
¬ record(n_{20}).
...
¬ record(n_{30}).

% Attributive features

tag(n_1, 'html').
tag(n_2, 'head').
...
tag(n_{27}, 'span').
tag(n_{29}, 'span').

depth(n_1, 1).
depth(n_2, 2).
...
depth(n_{29}, 5).
depth(n_{30}, 6).

children(n_1, 2).
children(n_2, 1).
...
children(n_{29}, 1).
children(n_{30}, 0).

style(n_8, 'a').
style(n_{25}, 'b').

ypos(n_1, 0).
ypos(n_2, 0).
...
ypos(n_{29}, 64).
ypos(n_{30}, 64).

xpos(n_1, 0).
xpos(n_2, 0).
...
xpos(n_{29}, 36).
xpos(n_{30}, 36).

len(n_1, 18).
len(n_2, 5).
...
len(n_{29}, 1).
len(n_{30}, 1).

isnumber(n_{13}).
isnumber(n_{14}).
isnumber(n_{23}).
isnumber(n_{24}).
isnumber(n_{29}).
isnumber(n_{30}).

% Relational features

parent(n_2, n_1).
parent(n_3, n_2).
...
parent(n_{29}, n_{26}).
parent(n_{30}, n_{29}).

left(n_5, n_2).
left(n_8, n_6).
...
left(n_{25}, n_8).
left(n_{29}, n_{27}).

Table 2.4: *Sample learning set for slot* record.

$$\{\langle \mathrm{record}(N_0), \mathrm{tag}(N_0, \text{'li'}), \mathrm{parent}(N_0, N_1), \mathrm{style}(N_1, \text{'a'})\rangle\}$$

To learn a rule set to extract nodes that belong to slot country*, TANGO first creates the learning set that is illustrated in Table §2.5. Note that it is similar to the learning set for slot* record*, but the positive and negative examples are related to slot* country *only and the features have been instantiated on the nodes that are involved in the enclosing record slots only; for instance, there is not any information regarding nodes* n_{29} *or* n_{30} *because they are out of the scope of the records within which the* country *slots are contained. The rules that TANGO learns from this learning set are the following:*

$$\{\langle \mathrm{country}(N_0), \mathrm{xpos}(N_0, A_1), A_1 \leq 20, \neg\mathrm{left}(\_, N_0)\rangle,$$
$$\langle \mathrm{country}(N_0), \mathrm{xpos}(N_0, 20)\rangle\}$$

The former states that node N_0 *must be extracted as a country if it is rendered horizontally at no more than 20 pixels and does not have a right sibling; that is, it extracts countries for which the system provides a phone code.*

```
% Positive examples

country(n_{11}).
country(n_{17}).
country(n_{21}).

% Negative examples

¬ country(n_{10}).
¬ country(n_{12}).
¬ country(n_{13}).
¬ country(n_{14}).
¬ country(n_{16}).
¬ country(n_{18}).
¬ country(n_{20}).
¬ country(n_{23}).
¬ country(n_{24}).

% Attributive features

tag(n_{10}, 'span').
tag(n_{13}, 'span').
tag(n_{20}, 'span').
tag(n_{23}, 'span').

depth(n_{10}, 5).
depth(n_{11}, 6).
...
depth(n_{23}, 5).
depth(n_{24}, 6).

children(n_{10}, 5).
children(n_{11}, 6).
...
children(n_{23}, 5).
children(n_{24}, 6).

ypos(n_{10}, 16).
ypos(n_{11}, 16).
...
ypos(n_{23}, 48).
ypos(n_{24}, 48).
xpos(n_{10}, 0).
xpos(n_{11}, 0).
...
xpos(n_{23}, 36).
xpos(n_{24}, 36).

len(n_{10}, 1).
len(n_{11}, 1).
...
len(n_{23}, 1).
len(n_{24}, 1).

isnumber(n_{13}).
isnumber(n_{14}).
isnumber(n_{23}).
isnumber(n_{24}).

% Relational features

parent(n_{11}, n_{10}).
parent(n_{14}, n_{13}).
parent(n_{21}, n_{20}).
parent(n_{24}, n_{23}).

left(n_{12}, n_{10}).
left(n_{13}, n_{12}).
left(n_{17}, n_{16}).
left(n_{18}, n_{17}).
left(n_{22}, n_{20}).
left(n_{23}, n_{22}).
```

Table 2.5: *Sample learning set for slot* $\mathtt{country}$.

The latter requires the nodes to be extracted to be rendered horizontally at exactly 20 pixels; that is, it extracts the countries for which the system does not provide a phone code.

Regarding slot $\mathtt{code}$*, TANGO first creates a learning set that is specifically tailored to this slot and then learns the following rule set:*

$$\{\langle \mathtt{code}(N_0), \neg \mathtt{tag}(N_0, \_), \mathtt{isnumber}(N_0)\rangle\}$$

This rule states that node N_0 *must be extracted as a code if it does not have a tag and it is a number. Note that both the text nodes that contain the codes and their parents are numbers, so the first condition just prevents the rule from extracting the parents. Note, too, that the node that contains the copyright year does not have a tag and it is also a number, but it does not matter because phone codes are extracted within the context of a record, so there is no room for confusion.*

2.2.3 Learning a rule set

Figure §2.4 presents the procedure to learn a rule set, which works on a learning set and a slot; it returns a set of rules that are specifically tailored to

```
method $\mathtt{learnRuleSet}(\mathtt{learningSet}, \mathtt{slot})$
   – Step 1: initialisation.
   $\mathtt{ruleSet} = \emptyset$
   – Step 2: learn rules.
   repeat
      $\mathtt{rule} = \mathtt{learnRule}(\mathtt{learningSet}, \mathtt{slot})$
      if $\mathtt{rule} \neq \mathtt{null}$ then
         $\mathtt{ruleSet} = \mathtt{ruleSet} \cup \{\mathtt{rule}\}$
         $\mathtt{learningSet} = \mathtt{learningSet} \setminus ($positive examples matched by $\mathtt{rule})$
      end
   until $\mathtt{rule} = \mathtt{null} \vee$ there are no positive examples in $\mathtt{learningSet}$
   – Step 3: post-process the rule set.
   $\mathtt{ruleSet} = \textsc{postProcessRuleSet}(\mathtt{ruleSet})$
return $\mathtt{ruleSet}$
```

Figure 2.4: *Procedure to learn a rule set.*

extracting information that belongs to that slot.

As usual, the first step is an initialisation step that simply sets the resulting rule set to the empty set.

The second step is a loop that iterates until no new rule is found or no positive example remains in the learning set. In each iteration, the procedure to learn a rule is invoked using the current learning set and the input slot as parameters. If this procedure returns null, then it means that it has not been able to find a rule that matches the positive examples in the learning set; otherwise, it returns a rule that matches some positive examples and no negative one, that is, a solution. If a rule is returned, then the resulting rule set is updated, the learning set is subtracted the positive examples that are matched by that rule, and it loops again if the learning set is not empty.

The third step post-processes the rule set learnt in an attempt to simplify it so that the resulting rules can be applied as efficiently as possible. We implemented this procedure as a variation point called POSTPROCESSRULESET because there are several post-processing alternatives and it is not clear beforehand which one is the best one.

Example 2.8 *To illustrate the procedure to learn a rule set, we focus on slot* country*. In the first iteration of Step 2, it invokes the procedure to learn a rule, which returns the following one:*

$$\langle \mathtt{country}(N_0), \mathtt{depth}(N_0, A_2), \mathtt{children}(N_0, A_3), \mathtt{ypos}(N_0, A_4),$$
$$\mathtt{xpos}(N_0, A_5), A_5 \leq 20, \neg\mathtt{left}(N_4, N_0)\rangle$$

Intuitively, the previous rule means that a node must be extracted as belonging to slot country *if features* depth*,* children*,* ypos*, and* xpos *can be instantiated on it, it is rendered horizontally at a maximum of 20 pixels, and it does not have a right sibling. In other words, this rule extracts the nodes that correspond to countries for which the system provides a phone code.*

The procedure then removes these positive examples matched by the previous rule from the learning set and invokes the procedure to learn a rule again. Now, it returns the following rule:

$$\langle \mathtt{country}(N_0), \mathtt{depth}(N_0, A_2), \mathtt{children}(N_0, A_3), \mathtt{ypos}(N_0, A_4),$$
$$\mathtt{xpos}(N_0, A_5), A_5 = 20\rangle$$

It is similar to the previous one, but requires the node to be rendered at an abscissa of exactly 20 pixels, which matches the nodes that correspond to countries for which the system does not provide a phone code.

Since the previous rule completes the rule set because it matches every remaining positive examples in the learning set, we then can proceed to post-processing both rules, which results in the following rule set:

$$\{\langle \mathtt{country}(N_0), \mathtt{xpos}(N_0, A_1), A_1 \leq 20, \neg\mathtt{left}(\_, N_0)\rangle,$$
$$\langle \mathtt{country}(N_0), \mathtt{xpos}(N_0, 20)\rangle\}$$

Realise that we always rename the variables in the original rule, that singleton variables are replaced by anonymous variables, and that constants are embedded in feature instantiators if possible. In this example, we have also removed conditions $\mathtt{depth}(N_0, A_2)$*,* $\mathtt{children}(N_0, A_3)$*, and* $\mathtt{ypos}(N_0, A_4)$ *because they are useless. The reason why rules may have useless conditions is that the procedure to learn them is based on a number of heuristics that are intended to guide the search for conditions as effectively and efficiently as possible, but there are situations in which some conditions that seem very promising at the early stages of the search finally turn out to be useless and can then be removed.*

2.2.4 Learning a rule

Figure §2.5 presents the procedure to learn a rule, which works on a learning set and a slot. It returns a solution if possible, that is, a rule that matches

```
method $learnRule(learningSet, slot)$
   – Step 1: initialisation.
   $savepoints = \emptyset$
   $var$ = generate a fresh variable
   $rule = \langle slot(var) \rangle$
   $score = \textsc{RuleScorer}(rule, learningSet)$
   – Step 2: extend the current rule.
   repeat
      – Step 2.1: compute and select candidates.
      $candidates = computeCandidates(rule, score, learningSet)$
      $(bestCandidates, saveCandidates) =$
            $\textsc{selectCandidates}(rule, score, candidates, learningSet)$
      – Step 2.2: update savepoints and current rule.
      $savepoints = updateSavepoints($
            $savepoints, rule, saveCandidates, learningSet)$
      $rule = rule \oplus \langle$conditions in $bestCandidates\rangle$
      $score = \textsc{RuleScorer}(rule, learningSet)$
      – Step 2.3: check for a replacement.
      if $bestCandidates = \emptyset \vee \textsc{isTooComplex}(rule, score, learningSet)$ then
         $(rule, savepoints) = findBestSavepoint(savepoints)$
      end
   until $rule = null \vee isSolution(rule, learningSet)$
   – Step 3: check for better savepoints.
   if $rule \neq null$ then
      $rule = findBetterSavepoint(savepoints, rule, score, learningSet)$
   end
return $rule$
```

Figure 2.5: *Procedure to learn a rule.*

at least a positive example, but no negative one; in cases in which a solution cannot be found, it returns a $null$ value.

The first step consists in initialising a set of savepoints and the rule that is going to be learnt. The savepoints are initialised to an empty set; during the learning process, this set stores some promising rules that might be used to backtrack if the current search path is not good enough. The rule is initialised to $\langle slot(var) \rangle$, where $slot$ denotes the slot for which the procedure is learning a rule and var denotes a fresh variable; in our examples we use N_0 to denote that variable. This rule trivially matches every example in the learning set because no conditions have been added to its body yet. Note that we need to compute a score to assess how good the rule is; since this computation can be accomplished in a variety of different ways, we have implemented this procedure as a variation point called RULESCORER.

The second step is a loop that extends the initial rule with new conditions and updates the savepoints. It consists of three sub-steps. The first one computes a set of candidates. Then, it selects a subset of them to extend the current rule and another subset to update the savepoints, which is implemented using a variation point called SELECTCANDIDATES because there are several alternatives available. The second sub-step first calls a procedure to update the savepoints, then extends the current rule, and finally re-computes its score. The third sub-step checks for a replacement of the current rule, which is a savepoint to which the procedure can backtrack in cases in which no candidate is selected to extend the current rule or cases in which it is too complex. Note that the learning process might explore arbitrarily complex rules, which does not make sense in practice. This calls for a mechanism to check whether a rule is complex enough not to explore it. We have implemented this mechanism using a variation point called ISTOOCOMPLEX because there are several choices available.

The third step attempts to substitute the rule learnt in the previous step by a better savepoint. Assume, for instance, that $r_1 = \langle c_1, c_2 \rangle$ is the current rule and that it can be extended as $r_2 = \langle c_1, c_2, c_3 \rangle$ or $r_3 = \langle c_1, c_2, c_4 \rangle$. Assume, too, that r_2 is a solution that matches 10 positive examples and no negative one, whereas r_3 matches 30 positive examples and one negative example. Even though rule r_2 is a solution, rule r_3 might be considered to provide more gain because it matches more positive examples and thus might lead to a smaller rule set. In such cases, the search should stick with r_3 and r_2 should be kept as a savepoint. The problem is that, in the following iterations, rule r_3 might lead to a solution that matches less than 10 positive examples; in such cases, which are not frequent but happen in practice, it is necessary to check if there is a better savepoint, in which case, it must obviously be returned.

Example 2.9 *Assume that we have to learn a rule to extract nodes that belong to slot* $\mathsf{country}$. *Our procedure starts working on the following initial rule:*

$\langle \mathsf{country}(N_0) \rangle$,

which matches every positive and negative example in the learning set. This rule scores at -2.12, *which simply confirms that it is not very good. The procedure then computes the following candidates, which are represented as triples in which the first component is a condition that might possibly be* *added to the current rule, the second one is its corresponding gain, and the third one indicates whether the condition is determinate.*

$$\{(\neg\mathsf{tag}(N_0, A_1), 1.59, \mathsf{false}), (\mathsf{depth}(N_0, A_2), 0.00, \mathsf{true}),\\ (\mathsf{children}(N_0, A_3), 0.00, \mathsf{true}), (\mathsf{ypos}(N_0, A_4), 0.00, \mathsf{true}),\\ (\mathsf{xpos}(N_0, A_5), 0.00, \mathsf{true})\}$$

It then has to select the candidates to extend the current rule. In our example, we use the following heuristic to illustrate how to select them: if it exists, we select the candidate with the maximum gain as long as it is at least 80% *the maximum gain that a condition can achieve on the current rule; otherwise, we select every determinate condition; if no such condition is a candidate, we then select the one with the highest gain. The idea behind this heuristic is that we must first explore candidates that provide a high gain, as long as it is close to the maximum gain that a condition might achieve on the current rule. If no such candidate exists, we then add determinate conditions to the rule; typically, these conditions provide little or no gain at all, but note that they are feature instantiators, which means that they allow to explore new features that may lead to better conditions in the succeeding iterations. Note that several determinate conditions can be added at a time because they do not constraint the positive examples matched at all; the reason is that they are based on features that are guaranteed to be instantiatable once on every positive example, that is, they simply put a foundation so that other conditions can be added to the rule. If no high-gain conditions or determinate conditions are available, then the best condition found is selected to extend the current rule as a best effort.*

Since the current rule scores at -2.12*, the maximum gain that a condition can achieve on it is* 6.35*, which happens when that condition preserves the positive examples matched by the current rule but does not match any negative examples. In other words, the minimum gain that a condition must achieve so that it can be selected to extend the current rule is* 80% *that score, that is,* 5.08*. None of the previous candidates achieves this minimum gain, which means that we have to resort to determinate conditions to extend the current rule. That is, we extend the current rule as follows:*

$$\langle\mathsf{country}(N_0), \mathsf{depth}(N_0, A_2), \mathsf{children}(N_0, A_3), \mathsf{ypos}(N_0, A_4),\\ \mathsf{xpos}(N_0, A_5)\rangle$$

This rule scores at -2.12*, as expected, because the determinate conditions that we have selected do not result in any gain. This is very common on the first iteration because it is necessary to have some feature instantiators before a comparator or some other feature instantiators regarding their neighbours can be added to the rule.*

We also have to select some candidates to update the set of savepoints, which is currently empty. In this example, we use the following heuristic: we select the best candidate that results in a solution, if any; if the current rule is extended with a non-determinate condition, then we select the candidates that achieve a gain of at least 80% the gain of that condition. In our example, none of the candidates results in a solution and the current rule was extended using determinate conditions; thus, none of the previous conditions is selected and the set of savepoints remains empty.

Branching the current rule results in the following candidates:

$$\begin{aligned}&\{(\neg\mathtt{tag}(N_0, A_1), 1.59, \mathtt{false}), (\neg\mathtt{isnumber}(N_0), 1.59, \mathtt{false}),\\&(\mathtt{parent}(N_0, N_1), 2.23, \mathtt{false}), (A_2 \neq 5, 2.23, \mathtt{false}),\\&(A_2 > 5, 2.23, \mathtt{false}), (A_2 = 6, 2.23, \mathtt{false}), (A_2 \geq 6, 2.23, \mathtt{false}),\\&(A_5 \leq 20, 3.35, \mathtt{false}), (A_5 < 32, 3.35, \mathtt{false})\}\end{aligned}$$

Note that condition $\mathtt{parent}(N_0, N_1)$ *was determinate before, but not now. Recall that a condition being determinate or indeterminate depends on the current rule; in this case, condition* $\mathtt{parent}(N_0, N_1)$ *is analysed in the context of a rule to extract country slots, which are nested within record slots. That means that a positive example that represents a country for which the system does not provide a phone code does not have a parent in the learning set, which is the reason why this condition is not determinate.*

In our example, none of the previous candidates achieve the minimum gain required to be selected to extend the current rule, which means that we would have to resort to the determinate conditions; unfortunately, none of the candidates is a determinate condition, which implies that we have to add the condition that provides more gain, that is, $A_5 \leq 20$ *or* $A_5 < 32$*; since there is a tie, we break it arbitrarily and select the first condition to extend the current rule, which then becomes the following one:*

$$\begin{aligned}&\langle\mathtt{country}(N_0), \mathtt{depth}(N_0, A_2), \mathtt{children}(N_0, A_3), \mathtt{ypos}(N_0, A_4),\\&\mathtt{xpos}(N_0, A_5), A_5 \leq 20\rangle\end{aligned}$$

This rule scores at -1.32 *and the maximum possible gain that a condition may achieve on it is* 2.64*, which implies that a condition must achieve a gain of at least* 2.11 *so that it can be selected to create a savepoint. That is, any of the candidates might be selected to update the savepoints, except for* $\neg\mathtt{tag}(N_0, A_1)$ *and* $\neg\mathtt{isnumber}(N_0)$*. Backtracking might be useful in some cases, but it is not generally required for our proposal to work well. Thus, we do not keep every possible savepoint, but a small subset. For instance, if we decide to keep only two savepoints, then the set would be updated as follows:*

$$\{(\langle \mathsf{country}(N_0), \mathsf{depth}(N_0, A_2), \mathsf{children}(N_0, A_3), \mathsf{ypos}(N_0, A_4), \mathsf{xpos}(N_0, A_5), \mathsf{parent}(N_0, N_1)\rangle, -1.00), (\langle \mathsf{country}(N_0), \mathsf{depth}(N_0, A_2), \mathsf{children}(N_0, A_3), \mathsf{ypos}(N_0, A_4), \mathsf{xpos}(N_0, A_5), A_5 < 32\rangle, -1.32)\}$$

Note that the savepoints are represented as tuples of the form (r, s)*, where* r *denotes a rule to which our proposal can backtrack if necessary and* s *denotes its corresponding score. Note, too, that ties are broken arbitrarily in cases in which two different savepoints might be added to the set of savepoints but there is not enough room.*

Branching the current rule results in the following candidates:

$$\{(\neg\mathsf{tag}(N_0, A_1), 1.47, \mathsf{false}), (\neg\mathsf{parent}(N_2, N_0), 1.47, \mathsf{false}), (\neg\mathsf{left}(N_4, N_0), 2.64, \mathsf{false})\}$$

Note that condition $\neg\mathsf{left}(N_4, N_0)$ *has gain 2.64, which is the maximum possible gain that can be achieved on the current rule. That means that adding this condition to the current rule results in a solution, namely:*

$$\langle \mathsf{country}(N_0), \mathsf{depth}(N_0, A_2), \mathsf{children}(N_0, A_3), \mathsf{ypos}(N_0, A_4), \mathsf{xpos}(N_0, A_5), A_5 \leq 20, \neg\mathsf{left}(N_4, N_0)\rangle$$

Intuitively, this rule matches the countries for which the system provides a phone code. These positive examples are then removed from the learning set and the process is restarted in order to learn a new rule that matches the countries for which the system does not provide a phone code.

2.2.5 Computing candidates

Figure §2.6 presents the procedure to compute the candidates that can possibly be used to extend a rule or to create new savepoints. It works on the current rule, its score, and a learning set; it returns a set of candidates. The candidates are represented as tuples of the form (c, g, d), where c denotes a condition, g the gain that is achieved when condition c is added to the input rule, and d is a Boolean value that indicates whether c is a determinate condition or not.

The first step branches the input rule, which consists in generating a sequence of conditions that can be used to extend it. There can be many approaches to generating such conditions, which is the reason why we have implemented it using a variation point called BRANCH.

```
method $computeCandidates(rule, score, learningSet)$
  – Step 1: branch the rule.
  $conditions = \textsc{Branch}(rule, learningSet)$
  – Step 2: bound the candidate conditions.
  $candidates = \emptyset$
  $stop = false$
  for each $condition$ in $conditions$ while $\neg stop$ do
    $newRule = rule \oplus \langle condition \rangle$
    $newScore = \textsc{RuleScorer}(newRule, learningSet)$
    $candidate = \textsc{Bound}(rule, score, newRule, newScore)$
    if $candidate \neq null$ then
      $candidates = candidates \cup \{candidate\}$
      $stop = \textsc{isPromisingCandidate}(rule, score, newRule, newScore)$
    end
  end
return $candidates$
```

Figure 2.6: *Procedure to compute candidates.*

The second step is a loop that bounds the conditions that have been generated in the previous step. Bounding a condition means that we assess its gain and decide on whether it is bad enough to prune it. Since there are several choices, we have implemented the bounding procedure using a variation point called BOUND. The search can be stopped at any moment, when a promising condition is found, that is, when a condition is considered so good that it is not necessary to continue exploring the others. Since there are also several alternatives to implement this stopping criterion and it is not clear which one is the best one, we have implemented it using a variation point called ISPROMISINGCANDIDATE.

Example 2.10 *Assume that we have to branch the following initial rule:*

$\langle country(N_0) \rangle$

A simple approach consists of generating every possible feature instantiator regarding node N_0*, which would result in the following conditions:*

$\langle tag(N_0, A_1), \neg tag(N_0, A_1),$
$depth(N_0, A_2), \neg depth(N_0, A_2), children(N_0, A_3), \neg children(N_0, A_3),$
$ypos(N_0, A_4), \neg ypos(N_0, A_4), xpos(N_0, A_5), \neg xpos(N_0, A_5),$

$\mathtt{len}(N_0, A_6), \neg\mathtt{len}(N_0, A_6), \mathtt{isnumber}(N_0), \neg\mathtt{isnumber}(N_0),$
$\mathtt{parent}(N_0, N_1), \neg\mathtt{parent}(N_0, N_1), \mathtt{parent}(N_2, N_0), \neg\mathtt{parent}(N_2, N_0),$
$\mathtt{left}(N_0, N_3), \neg\mathtt{left}(N_0, N_3), \mathtt{left}(N_4, N_0), \neg\mathtt{left}(N_4, N_0)\rangle$

For instance, condition $\mathtt{tag}(N_0, A_1)$ *binds the value computed for the* tag *feature on node* N_0 *to variable* A_1*. Note that we also generate negated conditions that allow to check if a feature cannot be instantiated on a node. For instance, condition* $\neg\mathtt{parent}(N_0, N_1)$ *checks that node* N_0 *does not have a parent. Note, too, that argument permutations can be explored in the case of relational features. For instance, this allows to explore the right siblings of node* N_0 *using condition* $\mathtt{left}(N_4, N_0)$*. In this example, recursion is not considered because it would result in a non-sense rule. Finally, note that feature* style *is not involved in any of the previous conditions because it cannot be instantiated on any of the examples in the corresponding learning set, cf. Table §2.5.*

Assume now that we have to branch the following rule:

$\langle\mathtt{country}(N_0), \mathtt{depth}(N_0, A_2)\rangle$

This rule has a feature instantiator that is intended to bind the values of feature depth *to variable* A_2*. In this case, the following additional conditions are generated:*

$\langle A_2 = 5, A_2 \neq 5, A_2 > 5, A_2 < 5, A_2 \geq 5, A_2 \leq 5,$
$A_2 = 6, A_2 \neq 6, A_2 > 6, A_2 < 6, A_2 \geq 6, A_2 \leq 6\rangle$

That is, when a variable can be instantiated to the value of a feature, we generate additional conditions using standard comparators and the values of the feature that we have found in the learning set.

Example 2.11 *To bound the previous conditions, we examine them one after the other and we use the following heuristic: if adding a condition does not result in a gain that is at least* 80% *the gain of the best condition found so far, then we prune it unless it is a determinate condition; if a determinate condition is found, then the pruning threshold is changed to* 80% *the maximum gain that a condition can achieve on the current rule. Note that determinate conditions help expand the search space in cases in which no better condition is found, which is the reason why they are not pruned, but make the bounding criterion more demanding.*

Below, we show the set of all possible candidates regarding the initial rule $\langle\mathtt{country}(N_0)\rangle$*:*

$$\begin{array}{ll}
\{(\mathsf{tag}(N_0, A_1), ?, \mathsf{false}), & (\neg\mathsf{tag}(N_0, A_1), 1.59, \mathsf{false}) \\
\;(\mathsf{depth}(N_0, A_2), 0.00, \mathsf{true}), & (\neg\mathsf{depth}(N_0, A_2), ?, \mathsf{false}) \\
\;(\mathsf{children}(N_0, A_3), 0.00, \mathsf{true}), & (\neg\mathsf{children}(N_0, A_3), ?, \mathsf{false}) \\
\;(\mathsf{ypos}(N_0, A_4), 0.00, \mathsf{true}), & (\neg\mathsf{ypos}(N_0, A_4), ?, \mathsf{false}) \\
\;(\mathsf{xpos}(N_0, A_5), 0.00, \mathsf{true}), & (\neg\mathsf{xpos}(N_0, A_5), ?, \mathsf{false}) \\
\;(\mathsf{len}(N_0, A_6), 0.00, \mathsf{false}), & (\neg\mathsf{len}(N_0, A_6), ?, \mathsf{false}) \\
\;(\mathsf{isnumber}(N_0), ?, \mathsf{false}), & (\neg\mathsf{isnumber}(N_0), 1.59, \mathsf{false}) \\
\;(\mathsf{parent}(N_0, N_1), 2.23, \mathsf{false}), & (\neg\mathsf{parent}(N_0, N_1), -1.05, \mathsf{false}) \\
\;(\mathsf{parent}(N_2, N_0), ?, \mathsf{false}), & (\neg\mathsf{parent}(N_2, N_0), 1.59, \mathsf{false}) \\
\;(\mathsf{left}(N_0, N_3), -0.47, \mathsf{false}), & (\neg\mathsf{left}(N_0, N_3), 0.62, \mathsf{false}) \\
\;(\mathsf{left}(N_4, N_0), -0.47, \mathsf{false}), & (\neg\mathsf{left}(N_4, N_0), 0.62, \mathsf{false})\}
\end{array}$$

Note that there are some cases in which the gain is denoted with a "?", which correspond to cases in which the rule score is indeterminate. For instance, rule

$$\langle \mathsf{country}(N_0), \mathsf{tag}(N_0, A_1) \rangle$$

does not match any positive examples because the country nodes are text nodes and then do not have a tag; this leads to an indetermination when its score is computed. Conditions for which a score cannot be computed can be trivially pruned. Note, too, that there are some conditions whose gain is negative, which means that adding them to the current rule would be counter-productive since it would result in a worse rule. For instance, condition $\mathsf{left}(N_0, N_3)$ *would result in a rule whose score is* -2.58*, which leads to negative gain because it reduces the number of positive examples matched very significantly.*

To bound the candidates, we first examine condition $\mathsf{tag}(N_0, A_1)$*, which is pruned because it leads to an indetermination; it then examines* $\neg\mathsf{tag}(N_0, A_1)$*, whose gain is* 1.59*; this is the best condition found so far, so the pruning threshold is set to* 80% *that gain, that is* 1.27*; this means that the following conditions shall be pruned unless they can achieve this minimum gain or they are determinate. Then, condition* $\mathsf{depth}(N_0, A_2)$ *is examined; note that it does not achieve any gain at all, but it is determinate because feature* depth *can be instantiated on the positive and negatives examples matched by the current rule exactly once; this means that we have already found a condition that can expand the search space and possibly lead to a better rule in the next iteration of our algorithm. The maximum gain that a condition can achieve on the current rule is* 6.35*, so we can set the new pruning threshold to* 80% *this maximum, that is,* 5.08 *instead of* 1.27*.*

The process would then continue and would result in the following candidates:

$\{(\neg\mathsf{tag}(N_0, A_1), 1.59, \mathsf{false}), (\mathsf{depth}(N_0, A_2), 0.00, \mathsf{true}),$
$(\mathsf{children}(N_0, A_3), 0.00, \mathsf{true}), (\mathsf{ypos}(N_0, A_4), 0.00, \mathsf{true}),$
$(\mathsf{xpos}(N_0, A_5), 0.00, \mathsf{true})\}$

Note that the bounding heuristic that we have presented is very demanding. A condition like $\mathsf{parent}(N_0, N_1)$ *clearly provides more gain than* $\neg\mathsf{tag}(N_0, A_1)$*, but it is bound because it is not determinate in the context of the current rule and we have already found some determinate conditions before, which has increased the selection threshold. Building on our experience, we never prune determinate conditions because they introduce feature instantiators that allow to explore new conditions in the following iterations, which typically helps find better rules. Note that pruning* $\mathsf{parent}(N_0, N_1)$ *in this iteration does not entail that it is discarded forever; it can be a candidate in a forthcoming iteration if it proves to be good enough in the context of the corresponding rule.*

2.2.6 Managing savepoints

TANGO maintains a set of savepoints, which are rules to which it can backtrack if the current search path does not lead to any good candidates or the rule being learnt is too complex. In practice, we have found that backtracking is not very common in practice when our proposal is configured properly, but there might be cases in which it could be useful, which is the reason why we implemented it. We have devised three procedures to manage the savepoints, namely: $\mathsf{updateSavepoints}$, to update them, $\mathsf{findBestSavepoint}$, to find the best one, if any, and $\mathsf{findBetterSavepoint}$, to find one that is better than a given solution, if any. Next, we provide additional explanations on each procedure.

Updating savepoints. Figure §2.7 shows the procedure to update the savepoints. It works on the current set of savepoints, the current rule (before it is extended with the best candidates), the set of candidates that have been selected to update the savepoints, and the learning set. It returns an updated set of savepoints that fulfils two properties, namely: a) it has k savepoints at most, where k is a user-defined parameter that we recommend to set to 20 (our experience proves that it is not common to backtrack if TANGO is properly configured, so we do not usually wish to keep a large set of savepoints that are commonly useless); b) and there is at most a savepoint that is a solution (note that if TANGO backtracks to a savepoint that is a solution, then it returns it immediately, so it suffices to keep the best solution found as a savepoint).

```
method $updateSavepoints(savepoints, rule, candidates, learningSet)$
    for each candidate $(c, s, d)$ in $candidates$ do
        – Step 1: extend the current rule.
        $newRule = rule \oplus \langle c \rangle$
        $newScore = \text{RULESCORER}(newRule, learningSet)$
        – Step 2: find a savepoint to replace.
        $(r, s) = (null, -\infty)$
        if $isSolution(newRule, learningSet)$ then
            $(r, s) =$ find a savepoint $(r, s)$ in $savepoints$ such that
                  $isSolution(r, learningSet)$
        end
        if $r = null \wedge |savepoints| = k$ then
            $(r, s) =$ find savepoint $(r, s)$ in $savepoints$ such that
                  $s$ is the minimum score
        end
        – Step 3: update the set of savepoints
        if $r \neq null \wedge newScore > s$ then
            $savepoints = savepoints \setminus \{(r, s)\} \cup \{(newRule, newScore)\}$
        elsif $r = null$ then
            $savepoints = savepoints \cup \{(newRule, newScore)\}$
        end
    end
return $savepoints$
```

Figure 2.7: *Procedure to update the savepoints.*

The procedure iterates over the set of candidates and proceeds in three steps. In the first step, it creates a new rule by adding the condition in the current candidate and computes its score. In the second step, it searches for a savepoint to replace, namely: if the new rule is a solution, then it searches for the only savepoint that is a solution, if any; if it is not a solution or it is a solution but there is not a savepoint that is a solution, it then retrieves the savepoint with the minimum score if the set of savepoints is full (since, otherwise, there is no need to replace any savepoints). We assume that if the search for a savepoint fails, then the rule returned is $null$ and the corresponding score is $-\infty$. The third step updates the savepoints as follows: if there is a savepoint to replace and its score is smaller than the score of the new rule, then it is replaced; otherwise, if there is not a savepoint to replace, the new rule is added to the set of savepoints. This guarantees that there are no more than k savepoints, of which only one can be a solution.

```
method $\mathit{findBestSavepoint}(\mathit{savepoints})$
    $(r, s) =$ select a savepoint $(r, s)$ with maximum score from $\mathit{savepoints}$
    $sp = \mathit{savepoints} \setminus \{(r, s)\}$
return $(r, sp)$
```

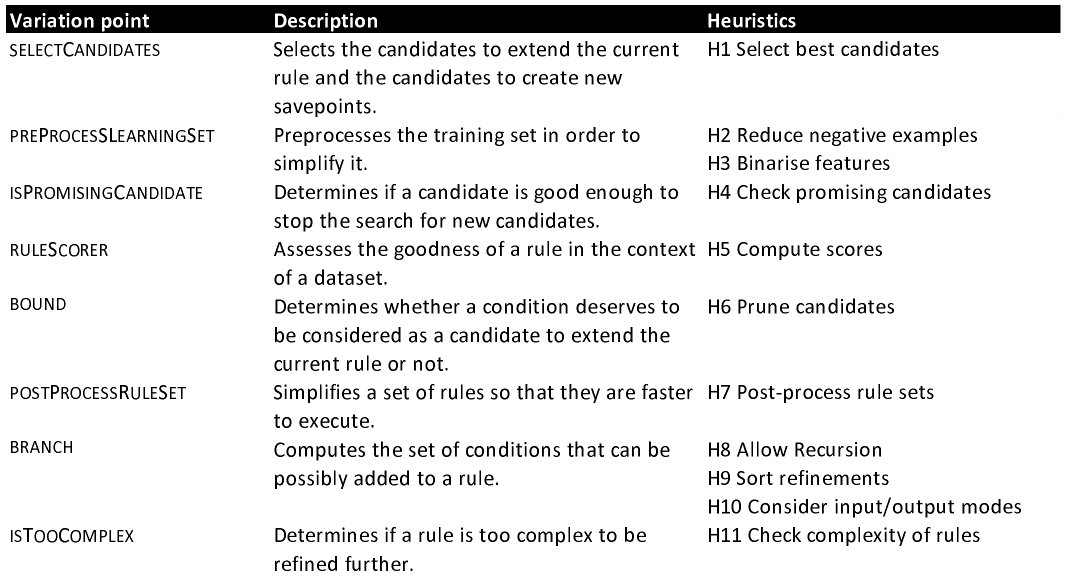
Figure 2.8: *Procedure to find the best savepoint.*

```
method $\mathit{findBetterSavepoint}(\mathit{savepoints}, \mathit{rule}, \mathit{score}, \mathit{learningSet})$
    $(r, s) =$ select a savepoint $(r, s)$ from $\mathit{savepoints}$ such that
            $\mathit{isSolution}(r, \mathit{learningSet})$
    if $r \neq \mathit{null} \wedge s > \mathit{score}$ then
        $\mathit{rule} = r$
    end
return $\mathit{rule}$
```

Figure 2.9: *Procedure to find a better savepoint.*

The procedure iterates over the set of candidates and proceeds in three steps. In the first step, it creates a new rule by adding the condition in the current candidate and computes its score. In the second step, it searches for a savepoint to replace, namely: if the new rule is a solution, then it searches for the only savepoint that is a solution, if any; if it is not a solution, then it retrieves the savepoint with the minimum score if the set of savepoints is full (since, otherwise, there is no need to replace any savepoint). We assume that if the search for a savepoint fails, then the rule returned is null and the corresponding score is $-\infty$. The third step updates the savepoints as follows: if there is a savepoint to replace and its score is smaller than the score of the new rule, then it is replaced; otherwise, if there is not a savepoint to replace, the new rule is added to the set of savepoints. This guarantees that there are no more than k savepoints, of which only one can be a solution.

Finding the best savepoint. Figure §2.8 shows the procedure to find the best savepoint. It works on the current set of savepoints and returns a tuple of the form (r, sp), where r denotes the rule associated with the savepoint that has the maximum score or null if no such savepoint exists, and sp denotes the updated set of savepoints.



Variation point	Description	Heuristics
SELECTCANDIDATES	Selects the candidates to extend the current rule and the candidates to create new savepoints.	H1 Select best candidates
PREPROCESSLEARNINGSET	Preprocesses the training set in order to simplify it.	H2 Reduce negative examples H3 Binarise features
ISPROMISINGCANDIDATE	Determines if a candidate is good enough to stop the search for new candidates.	H4 Check promising candidates
RULESCORER	Assesses the goodness of a rule in the context of a dataset.	H5 Compute scores
BOUND	Determines whether a condition deserves to be considered as a candidate to extend the current rule or not.	H6 Prune candidates
POSTPROCESSRULESET	Simplifies a set of rules so that they are faster to execute.	H7 Post-process rule sets
BRANCH	Computes the set of conditions that can be possibly added to a rule.	H8 Allow Recursion H9 Sort refinements H10 Consider input/output modes
ISTOOCOMPLEX	Determines if a rule is too complex to be refined further.	H11 Check complexity of rules

Table 2.6: *Summary of variation points.*

Finding a better savepoint. Figure §2.9 shows the procedure to find a better savepoint, as long as it is a solution. It works on the current set of savepoints, a rule that is a solution, its corresponding score, and the learning set. It first searches for the only savepoint that is a solution; if it exists, then it replaces the current rule by the rule that is associated with the savepoint if it has a better score; otherwise, it returns the input rule.

2.3 Configuring our proposal

TANGO relies on a number of variation points, each of which is expected to implement one or more heuristics for which there are several alternatives. In the sequel, we refer to a specific combination of alternatives as a configuration. TANGO was first implemented using a combination of default alternatives, which were the simplest ones of which we could think; our goal was to study whether replacing them with more sophisticated ones might have a positive impact on its overall performance.

We first analysed every heuristic in isolation and guessed which ones might contribute the most to improving the overall performance of our system; this allowed us to arrange them (and consequently the variation points) in a list that we explored sequentially, cf. Table §2.6. In order to make a decision regarding whether replacing a default alternative was appropriate or not, we setup the corresponding configuration, run the resulting system on a collection of datasets, and computed the usual performance measures: precision (P), recall (R), and the F_1 score (F_1), as effectiveness measures, and learning time (LT), and extraction time (ET), as efficiency measures. We then used a method to compute a rank for each of the resulting configurations and made a decision. We set the weight of F_1 score to 50%, the weight of LT to 30%, and the weight of ET to 20%. These figures highlight that effectiveness is very important, since the goal is to achieve rules that are very precise and have high recall, and the learning time is a little more important than the extraction time. We then used a method to compute a rank for each of the resulting configurations so that we could make a decision. For further details, please, consult Appendices §A and §B.

In the following subsections, we provide additional details on our analysis of the many configurations that we have explored. In each case, we first provide an overall picture of the variation point and then report on the heuristic/s on which it relies, on the alternatives that we have taken into account, and then present our experimental results and discuss on them to make a decision regarding which of the alternatives is the best one.

2.3.1 Variation point SELECTCANDIDATES

The goal of this variation point is twofold: select the most promising candidates to expand the current rule and some of the remaining ones to create savepoints. This variation point consists of Heuristic H1 only. The goal is to customise TANGO so that it efficiently learns effective extraction rules whose efficiency does not degrade significantly when they are executed.

Heuristic H1: select best candidates. Given the current rule, it is generally possible to extend it using many different conditions. Selecting the best candidates has an impact on both the effectiveness and the efficiency of our system: it may lead to rules that are learnt faster, which has a positive impact on efficiency, but might not be general enough, which has a negative impact on effectiveness; contrarily, searching for very general rules may have a positive impact on effectiveness, but might lead to rules that take longer to be learnt. Thus, the right selection heuristic should find a balance between effectiveness and efficiency. There are cases in which wrong conditions are selected, so that it might be necessary to backtrack. This implies that we also need to select the candidates that are the most appropriate to create new savepoints to which TANGO can backtrack if necessary. Obviously, saving every possible candidate is not a choice because this would require much memory to store them and much time to explore them.

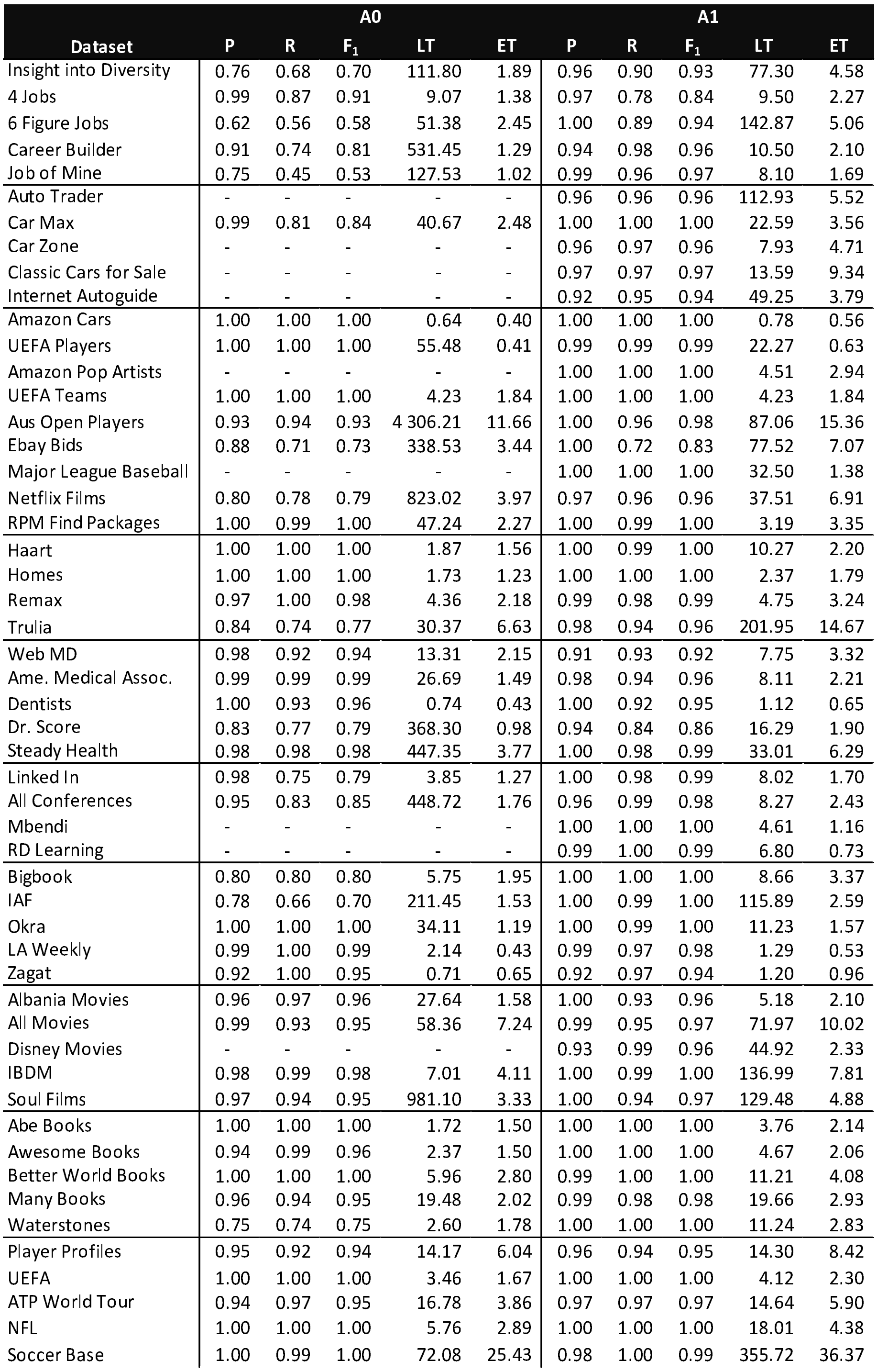
The alternatives that we have devised to implement this heuristic are the following: A0) Select the candidate that provides the maximum gain to extend the current rule and then select the following k candidates to create savepoints. A1) Select the candidate with the maximum gain as long as it is at least 80% the maximum gain that a condition can achieve on the current rule; otherwise, select every determinate condition; if no such condition is a candidate, then select the one with the highest gain. To create the savepoints from the remaining candidates that were not selected to expand the current rule, we select the candidate with the highest gain out of the candidates that result in a solution, if any; if the best candidate to expand the rule corresponds to a non-determinate condition, then we also select the candidates whose gain is at least 80% the gain of that condition to create the savepoints; otherwise, no more candidates are selected.

The first alternative is a simple approach that has been used many times in the literature. Although it is very simple and might work well in some cases, our intuition was that there are cases in which such candidates do not help learn good rules because they can easily lead to local maxima. In cases in which backtracking should be performed, the alternative considers that if the top candidate proves not to be adequate, then the next to the top candidate must be explored, and so forth. Although this approach might work in some cases, our intuition was the same: the savepoints could also lead to local maxima if only the maximum gain was considered. The rationale behind the second alternative is that the candidate with the maximum gain can be added to the current rule as long as its gain is high enough with regard to the maximum gain that a condition may achieve; otherwise, it is better to add determinate conditions to the current rule because they help get out of local maxima; recall that such conditions are feature instantiators that allow to explore new comparators and neighbour nodes in the succeeding iterations. If no determinate conditions exists, then we have to resort to the condition that provides the maximum gain, like in alternative A0. Note that there might be still cases in which we need to perform backtracking. Thus, we think that it makes more sense to select the candidates that result in a solution when they are added to the current rule; if that candidate is not selected to extend the current rule, then it means that there is another candidate that provides more gain because it leads to a rule that matches more positive examples and is then more promising because it is more general; but if it finally results in a bad decision, backtracking to a savepoint that is a solution helps stop the search process immediately. Now, if the candidates selected to extend the current rule are not determinate, we then select the candidates that achieve a gain that is high enough in comparison with that candidate; otherwise, if the selected candidates were determinate, no more candidates are selected to create savepoints. The rationale behind this idea is that if determinate conditions are selected, it then means that there were no conditions whose gain exceeded 80% the maximum gain since, otherwise, they would have been selected before determinate conditions. Thus, in this case, we are not interested in using conditions that do not provide enough gain to create savepoints. If a non-determinate condition is selected to expand the rule, we can be less demanding regarding the candidates that should be selected to create new savepoints so that we only select the k best candidates as long as their gain exceeds 80% the gain of the non-determinate condition selected to extend the current rule.

Discussion. The empirical results are shown in Table §2.7. The columns represent the alternatives for the heuristic being evaluated; sub-columns represent the performance measures that we have studied. The rows correspond to the datasets on which we performed our experiments. A dash in a cell means that the corresponding alternative was not able to learn a rule for a given dataset, be it because it failed to learn it or because it ran out of memory.

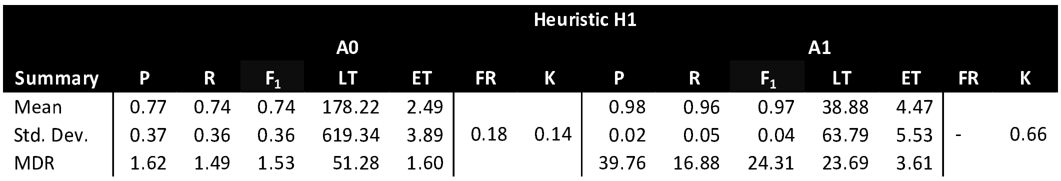
Our conclusion regarding effectiveness is that alternative A1 produces better results since precision, recall, and the F_1 score increase considerably with regard to alternative A0. In average, the precision of alternative A1 is 0.21 ± 0.39 higher, its recall is 0.23 ± 0.42 higher, and its F_1 score is 0.23 ± 0.40 higher than the corresponding measures regarding alternative A0. Furthermore, the standard deviation of every effectiveness measure in alternative A1 is smaller, which means that it is generally more stable than alternative A0, that is, it does not generally produce rules whose effectiveness largely deviates from the average.

Our conclusion regarding efficiency is that alternative A1 seems to be faster when learning rules since it is 139.34 ± 683.13 minutes faster than alternative A0, which is a significant improvement; however, alternative A1 is slower when executing the rules that it learns, but we must take into account that there are many datasets in which alternative A0 could not learn any rules and that, sometimes, the resulting rule sets were unable to match all of the positive examples, and therefore, this is very likely the reason why the rules produced by alternative A0 are 1.98 ± 9.42 minutes faster. Backtracking was not performed in many cases, so we cannot provide solid arguments



	A0					A1				
Dataset	P	R	F_1	LT	ET	P	R	F_1	LT	ET
Insight into Diversity	0.76	0.68	0.70	111.80	1.89	0.96	0.90	0.93	77.30	4.58
4 Jobs	0.99	0.87	0.91	9.07	1.38	0.97	0.78	0.84	9.50	2.27
6 Figure Jobs	0.62	0.56	0.58	51.38	2.45	1.00	0.89	0.94	142.87	5.06
Career Builder	0.91	0.74	0.81	531.45	1.29	0.94	0.98	0.96	10.50	2.10
Job of Mine	0.75	0.45	0.53	127.53	1.02	0.99	0.96	0.97	8.10	1.69
Auto Trader	-	-	-	-	-	0.96	0.96	0.96	112.93	5.52
Car Max	0.99	0.81	0.84	40.67	2.48	1.00	1.00	1.00	22.59	3.56
Car Zone	-	-	-	-	-	0.96	0.97	0.96	7.93	4.71
Classic Cars for Sale	-	-	-	-	-	0.97	0.97	0.97	13.59	9.34
Internet Autoguide	-	-	-	-	-	0.92	0.95	0.94	49.25	3.79
Amazon Cars	1.00	1.00	1.00	0.64	0.40	1.00	1.00	1.00	0.78	0.56
UEFA Players	1.00	1.00	1.00	55.48	0.41	0.99	0.99	0.99	22.27	0.63
Amazon Pop Artists	-	-	-	-	-	1.00	1.00	1.00	4.51	2.94
UEFA Teams	1.00	1.00	1.00	4.23	1.84	1.00	1.00	1.00	4.23	1.84
Aus Open Players	0.93	0.94	0.93	4 306.21	11.66	1.00	0.96	0.98	87.06	15.36
Ebay Bids	0.88	0.71	0.73	338.53	3.44	1.00	0.72	0.83	77.52	7.07
Major League Baseball	-	-	-	-	-	1.00	1.00	1.00	32.50	1.38
Netflix Films	0.80	0.78	0.79	823.02	3.97	0.97	0.96	0.96	37.51	6.91
RPM Find Packages	1.00	0.99	1.00	47.24	2.27	1.00	0.99	1.00	3.19	3.35
Haart	1.00	1.00	1.00	1.87	1.56	1.00	0.99	1.00	10.27	2.20
Homes	1.00	1.00	1.00	1.73	1.23	1.00	1.00	1.00	2.37	1.79
Remax	0.97	1.00	0.98	4.36	2.18	0.99	0.98	0.99	4.75	3.24
Trulia	0.84	0.74	0.77	30.37	6.63	0.98	0.94	0.96	201.95	14.67
Web MD	0.98	0.92	0.94	13.31	2.15	0.91	0.93	0.92	7.75	3.32
Ame. Medical Assoc.	0.99	0.99	0.99	26.69	1.49	0.98	0.94	0.96	8.11	2.21
Dentists	1.00	0.93	0.96	0.74	0.43	1.00	0.92	0.95	1.12	0.65
Dr. Score	0.83	0.77	0.79	368.30	0.98	0.94	0.84	0.86	16.29	1.90
Steady Health	0.98	0.98	0.98	447.35	3.77	1.00	0.98	0.99	33.01	6.29
Linked In	0.98	0.75	0.79	3.85	1.27	1.00	0.98	0.99	8.02	1.70
All Conferences	0.95	0.83	0.85	448.72	1.76	0.96	0.99	0.98	8.27	2.43
Mbendi	-	-	-	-	-	1.00	1.00	1.00	4.61	1.16
RD Learning	-	-	-	-	-	0.99	1.00	0.99	6.80	0.73
Bigbook	0.80	0.80	0.80	5.75	1.95	1.00	1.00	1.00	8.66	3.37
IAF	0.78	0.66	0.70	211.45	1.53	1.00	0.99	1.00	115.89	2.59
Okra	1.00	1.00	1.00	34.11	1.19	1.00	0.99	1.00	11.23	1.57
LA Weekly	0.99	1.00	0.99	2.14	0.43	0.99	0.97	0.98	1.29	0.53
Zagat	0.92	1.00	0.95	0.71	0.65	0.92	0.97	0.94	1.20	0.96
Albania Movies	0.96	0.97	0.96	27.64	1.58	1.00	0.93	0.96	5.18	2.10
All Movies	0.99	0.93	0.95	58.36	7.24	0.99	0.95	0.97	71.97	10.02
Disney Movies	-	-	-	-	-	0.93	0.99	0.96	44.92	2.33
IBDM	0.98	0.99	0.98	7.01	4.11	1.00	0.99	1.00	136.99	7.81
Soul Films	0.97	0.94	0.95	981.10	3.33	1.00	0.94	0.97	129.48	4.88
Abe Books	1.00	1.00	1.00	1.72	1.50	1.00	1.00	1.00	3.76	2.14
Awesome Books	0.94	0.99	0.96	2.37	1.50	1.00	1.00	1.00	4.67	2.06
Better World Books	1.00	1.00	1.00	5.96	2.80	0.99	1.00	1.00	11.21	4.08
Many Books	0.96	0.94	0.95	19.48	2.02	0.99	0.98	0.98	19.66	2.93
Waterstones	0.75	0.74	0.75	2.60	1.78	1.00	1.00	1.00	11.24	2.83
Player Profiles	0.95	0.92	0.94	14.17	6.04	0.96	0.94	0.95	14.30	8.42
UEFA	1.00	1.00	1.00	3.46	1.67	1.00	1.00	1.00	4.12	2.30
ATP World Tour	0.94	0.97	0.95	16.78	3.86	0.97	0.97	0.97	14.64	5.90
NFL	1.00	1.00	1.00	5.76	2.89	1.00	1.00	1.00	18.01	4.38
Soccer Base	1.00	0.99	1.00	72.08	25.43	0.98	1.00	0.99	355.72	36.37

Table 2.7: *Experimental results regarding Heuristic H1 .*



Heuristic H1														
	A0							A1						
Summary	P	R	F_1	LT	ET	FR	K	P	R	F_1	LT	ET	FR	K
Mean	0.77	0.74	0.74	178.22	2.49	0.18	0.14	0.98	0.96	0.97	38.88	4.47	-	0.66
Std. Dev.	0.37	0.36	0.36	619.34	3.89			0.02	0.05	0.04	63.79	5.53		
MDR	1.62	1.49	1.53	51.28	1.60			39.76	16.88	24.31	23.69	3.61		

Table 2.8: *Ranking of alternatives regarding Heuristic H1 .*

to support that alternative A1 is better or worse than alternative A0 regarding the selection of candidates to create savepoints.

Table §2.8 shows the ranks that we computed. We report on the mean and the standard deviation of each measure, its mean-to-deviation ratio (MDR), the failure ratio of each proposal (FR), and its rank (K). The rank of alternative A1 is 0.66, which is much better than the rank of alternative A0, which is 0.14. Note also that the failure ratio of alternative A1 is exactly zero, which means that it was able to learn rules for every dataset, whereas alternative A0 was not. Therefore, our conclusion is that alternative A1 is the best one.

2.3.2 Variation point PREPROCESSLEARNINGSET

This variation point deals with simplifying a learning set. It consists of two heuristics, namely: Heuristic H2 , which helps reduce the number of negative examples, and Heuristic H3 , which binarises the features. The goal is to customise TANGO so that it learns extraction rules that improve on effectiveness, efficiency, or both.

Heuristic H2: reduce negative examples. Web documents are typically composed of a large number of nodes, but only a few are positive examples. This means that there is not usually a balance between positive and negative examples. The effectiveness of our system is not affected by this characteristic of our learning sets, but it has an impact on its efficiency because checking which negative examples are matched by a rule requires time. Thus, the more negative examples, the more inefficient this process.

The alternatives that we have devised to implement this heuristic are the following: A0) Work with the whole learning set, that is, no reduction of negative examples is performed. A1) Select a subset of negative examples that are in the neighbourhood of every positive example. A2) Select a subset of negative examples that are in the neighbourhood of every positive example plus a random subset of the remaining negative examples. A3) Select a subset with the most similar negative examples that correspond to every positive example. A5) Select a subset with the most similar negative examples that correspond to every positive example plus a random subset of the remaining negative examples.

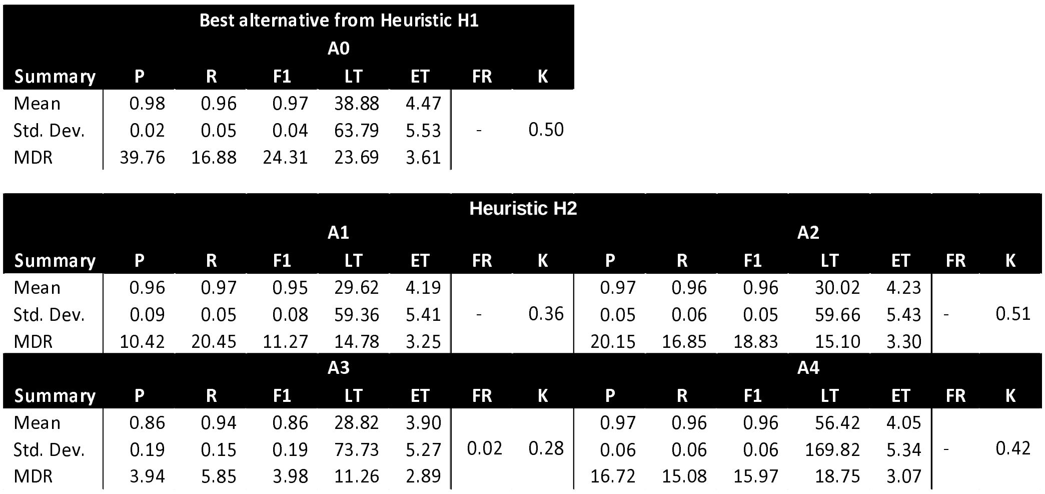
The first alternative is a simple approach in which every negative example is considered in the learning process. Our intuition was that this would not be efficient because there are typically many negative examples. Our hypothesis was that it would be possible to discard many such negative examples from the learning set without a negative impact on the effectiveness of the resulting rules, as long as the negative examples that are kept are still representative of the whole set of negative examples. The problem is how to find that subset. The other alternatives were intended to find them. The rationale behind alternative A1 is to discard the negative examples that are not in the neighbourhood of the nodes that correspond to the positive examples. By neighbourhood, we refer to the nodes that can be reached within a radius of each positive example by applying relational features transitively. We experimented with a radius $r = 10$ when computing the neighbourhood of a given positive example. Alternative A2 is based on A1 but it also includes a set of negative examples that are selected randomly; we set the radius to compute the neighbours to $r = 10$ and the percentage of nodes selected randomly to $p = 10\%$. Alternative A3 searches for the most similar negative examples and discards the remaining ones. We measured the similarity between any two nodes using the well-known Euclidean distance on the attributive features; in the case of non-numeric features, we computed the difference between two different values as 1.00 and the difference between equal values as 0.00. We experimented with the $k = 50$ most similar negative examples to each positive example. Alternative A4 explores the k most similar negative examples for each positive example but it also includes a small percentage of negative examples selected randomly from the remaining ones; we selected the $k = 50$ most similar negative examples to each positive example and $p = 50\%$ of the remaining negative examples. (We experimented with many different values for the radius and the percentage of negatives, but we cannot report on all of the results due to space constraints. This is the reason why we decided to report only on the best combinations that we found.)

Discussion. The empirical results are shown in Table §2.9 and the ranking is shown in Table §2.10. (Note that the rank of the baseline needs to be re-computed because it depends on the other alternatives.) Our conclusion is that the best alternative is A2. Selecting the closest neighbours in a

	A1					A2					A3					A4				
Dataset	P	R	F_1	LT	ET	P	R	F_1	LT	ET	P	R	F_1	LT	ET	P	R	F_1	LT	ET
Insight into Diversity	0.96	0.90	0.93	76.58	4.37	0.96	0.90	0.93	76.67	4.41	0.96	0.90	0.93	76.20	4.30	0.96	0.90	0.93	76.62	4.36
4 Jobs	0.97	0.78	0.84	8.93	1.92	0.97	0.78	0.84	8.54	2.01	0.97	0.78	0.84	8.24	1.91	0.97	0.78	0.84	8.68	1.99
6 Figure Jobs	1.00	0.89	0.94	100.93	5.40	0.83	0.89	0.81	102.63	5.35	0.78	0.89	0.74	20.20	3.66	1.00	0.89	0.94	42.62	4.26
Career Builder	0.94	0.98	0.96	9.55	1.83	0.94	0.98	0.96	9.33	1.91	0.94	0.98	0.96	8.83	1.75	0.94	0.98	0.96	9.69	1.85
Job of Mine	0.99	0.96	0.97	5.76	1.60	0.99	0.96	0.97	5.59	1.60	0.99	0.96	0.97	4.17	1.45	0.99	0.96	0.97	4.50	1.52
Auto Trader	0.96	0.96	0.96	27.14	4.98	0.96	0.96	0.96	30.27	5.07	0.87	0.96	0.89	13.81	4.81	0.96	0.96	0.96	55.91	5.08
Car Max	1.00	1.00	1.00	13.97	3.31	1.00	1.00	1.00	13.00	3.40	0.88	1.00	0.88	9.05	3.19	1.00	1.00	1.00	10.70	3.32
Car Zone	0.96	0.97	0.96	8.58	4.66	0.96	0.97	0.96	8.86	4.67	0.96	0.97	0.96	5.78	4.43	0.96	0.97	0.96	6.51	4.53
Classic Cars for Sale	0.97	0.97	0.97	7.76	7.87	0.97	0.97	0.97	8.08	7.94	0.97	0.97	0.97	7.68	7.74	0.97	0.97	0.97	19.29	8.19
Internet Autoguide	0.92	0.95	0.94	11.78	3.41	0.87	0.95	0.90	11.60	3.43	0.83	0.95	0.86	9.89	3.36	0.92	0.95	0.94	18.51	3.50
Amazon Cars	1.00	1.00	1.00	0.51	0.49	1.00	1.00	1.00	0.54	0.50	0.84	1.00	0.88	0.24	0.31	0.89	1.00	0.93	0.43	0.48
UEFA Players	0.43	0.99	0.48	15.46	0.54	0.91	0.99	0.95	15.91	0.56	0.91	1.00	0.95	5.22	0.51	0.93	0.98	0.95	506.86	0.69
Amazon Pop Artists	1.00	1.00	1.00	4.07	2.85	1.00	1.00	1.00	3.81	2.85	-	-	-	-	-	1.00	1.00	1.00	6.86	2.04
UEFA Teams	1.00	1.00	1.00	3.99	1.71	0.93	1.00	0.94	4.55	1.76	0.20	0.97	0.26	0.36	1.14	1.00	1.00	1.00	2.28	1.50
Aus Open Players	1.00	0.96	0.98	86.73	15.25	1.00	0.96	0.98	86.84	15.27	0.94	0.96	0.94	85.01	14.83	1.00	0.96	0.98	85.85	15.01
Ebay Bids	0.91	0.86	0.86	12.03	6.15	1.00	0.72	0.82	17.44	6.28	0.52	0.56	0.39	2.82	3.93	0.70	0.65	0.66	26.70	4.70
Major League Baseball	0.90	1.00	0.94	18.75	1.16	0.89	1.00	0.93	19.44	1.20	0.97	1.00	0.99	17.80	1.00	1.00	1.00	1.00	16.52	1.14
Netflix Films	0.97	0.96	0.96	36.61	6.75	0.97	0.96	0.96	37.96	6.79	0.78	0.96	0.79	34.48	6.48	0.97	0.96	0.96	35.50	6.62
RPM Find Packages	1.00	0.99	1.00	3.05	3.30	1.00	0.99	1.00	3.35	3.30	0.99	0.99	0.99	2.21	3.17	1.00	0.99	1.00	4.99	3.24
Haart	1.00	0.99	1.00	2.27	1.96	1.00	0.99	1.00	2.84	1.99	1.00	0.99	1.00	0.67	1.85	1.00	0.99	1.00	1.68	1.98
Homes	1.00	1.00	1.00	2.25	1.77	1.00	1.00	1.00	1.88	1.78	0.84	1.00	0.85	0.31	1.55	1.00	1.00	1.00	1.15	1.64
Remax	0.99	0.98	0.99	1.88	2.69	0.99	0.98	0.99	2.42	2.90	0.99	0.98	0.99	1.95	2.70	0.99	0.98	0.99	2.69	2.90
Trulia	0.98	0.94	0.96	180.07	11.74	0.98	0.94	0.96	185.67	11.95	0.98	0.94	0.96	175.96	10.92	0.98	0.94	0.96	195.63	11.48
Web MD	0.79	0.93	0.82	5.74	2.93	0.91	0.93	0.92	6.19	2.98	0.79	0.93	0.82	4.67	2.76	0.92	0.93	0.92	5.95	2.94
Ame. Medical Assoc.	0.98	0.94	0.96	6.37	2.08	0.98	0.94	0.96	6.23	2.08	0.80	0.94	0.79	5.05	2.00	0.98	0.94	0.96	5.89	2.08
Dentists	1.00	0.92	0.95	0.84	0.57	1.00	0.92	0.95	0.96	0.59	0.86	0.92	0.86	0.49	0.55	1.00	0.92	0.95	0.75	0.59
Dr. Score	0.94	0.84	0.86	16.06	1.87	0.94	0.84	0.86	16.11	1.87	0.87	0.84	0.82	15.52	1.78	0.94	0.84	0.86	15.87	1.82
Steady Health	1.00	1.00	1.00	32.08	6.16	1.00	1.00	1.00	32.02	6.18	0.78	1.00	0.80	30.27	5.23	1.00	1.00	1.00	31.15	5.96
Linked In	1.00	0.98	0.99	7.99	1.68	1.00	0.98	0.99	8.07	1.68	1.00	0.98	0.99	7.46	1.61	1.00	0.98	0.99	8.48	1.63
All Conferences	0.96	0.99	0.98	9.21	2.37	0.96	0.99	0.98	8.73	2.37	0.83	0.99	0.88	4.26	2.49	0.96	0.99	0.98	4.97	2.33
Mbendi	0.90	1.00	0.93	2.85	1.14	0.90	1.00	0.93	3.10	1.15	0.81	1.00	0.82	2.12	1.11	1.00	1.00	1.00	2.83	1.13
RD Learning	0.99	1.00	0.99	6.26	0.73	0.99	1.00	0.99	6.79	0.72	0.82	1.00	0.85	5.36	0.70	0.99	1.00	0.99	5.72	0.71
Bigbook	1.00	1.00	1.00	8.56	3.30	1.00	1.00	1.00	8.52	3.31	1.00	1.00	1.00	7.61	2.84	0.99	1.00	1.00	8.21	3.20
IAF	0.94	0.99	0.96	105.45	2.34	0.94	0.95	0.94	104.51	2.38	0.65	0.92	0.68	366.43	8.15	0.83	0.96	0.87	1 084.67	1.94
Okra	1.00	0.99	1.00	11.06	1.51	1.00	0.99	1.00	11.14	1.51	1.00	0.99	1.00	8.17	1.09	1.00	0.99	1.00	10.14	1.33
LA Weekly	0.99	0.97	0.98	1.31	0.50	0.99	0.97	0.98	1.30	0.51	0.69	0.95	0.78	0.49	0.47	0.98	0.96	0.97	4.37	0.61
Zagat	1.00	1.00	1.00	0.81	0.86	1.00	1.00	1.00	0.83	0.87	1.00	1.00	1.00	0.49	0.79	1.00	1.00	1.00	0.92	0.85
Albania Movies	0.98	0.96	0.97	4.71	2.03	1.00	0.96	0.98	4.78	2.05	1.00	0.96	0.98	3.99	1.95	1.00	0.96	0.98	4.90	2.00
All Movies	0.91	0.96	0.92	19.27	9.64	0.99	0.95	0.97	21.95	9.77	0.91	0.95	0.92	13.29	9.61	0.99	0.95	0.97	26.21	9.76
Disney Movies	0.78	0.99	0.84	15.94	2.06	0.78	0.99	0.84	16.53	2.04	0.74	0.99	0.77	2.00	1.52	0.82	0.99	0.88	8.17	1.70
IBDM	1.00	0.99	1.00	96.30	6.99	1.00	0.99	1.00	90.26	7.19	0.84	0.99	0.84	6.85	4.90	0.89	0.99	0.91	18.78	5.16
Soul Films	1.00	0.94	0.97	129.71	4.75	1.00	0.94	0.97	132.61	4.78	1.00	0.94	0.97	127.12	4.57	1.00	0.94	0.97	128.41	4.63
Abe Books	1.00	1.00	1.00	3.04	2.06	1.00	1.00	1.00	5.85	2.08	1.00	1.00	1.00	1.30	1.85	1.00	1.00	1.00	2.62	1.93
Awesome Books	1.00	1.00	1.00	3.53	1.98	1.00	1.00	1.00	3.87	2.00	1.00	1.00	1.00	2.57	1.87	1.00	1.00	1.00	3.56	1.92
Better World Books	0.98	1.00	0.99	2.70	3.55	0.99	1.00	1.00	2.86	3.58	0.99	1.00	1.00	1.50	3.45	1.00	1.00	1.00	2.84	3.66
Many Books	0.99	0.98	0.98	20.84	2.91	0.99	0.98	0.98	17.76	2.90	0.99	0.98	0.98	16.51	2.75	0.99	0.98	0.98	17.87	2.80
Waterstones	1.00	1.00	1.00	7.52	2.64	1.00	1.00	1.00	11.13	2.68	1.00	1.00	1.00	5.54	2.39	1.00	1.00	1.00	8.71	2.54
Player Profiles	0.96	0.94	0.95	14.22	8.38	0.96	0.94	0.95	14.22	8.40	0.80	0.94	0.79	13.26	8.12	0.96	0.94	0.95	13.67	8.25
UEFA	1.00	1.00	1.00	1.43	1.87	1.00	1.00	1.00	1.64	1.95	0.82	1.00	0.84	0.57	1.63	1.00	1.00	1.00	1.73	1.97
ATP World Tour	0.97	0.97	0.97	11.00	5.38	0.97	0.97	0.97	8.59	5.39	0.82	0.97	0.83	5.84	4.46	0.97	0.97	0.97	9.77	5.21
NFL	1.00	1.00	1.00	2.79	3.77	1.00	1.00	1.00	2.28	3.90	0.86	1.00	0.86	0.72	2.40	1.00	1.00	1.00	5.86	3.88
Soccer Base	0.98	1.00	0.99	353.98	36.25	0.98	1.00	0.99	354.91	36.34	0.98	1.00	0.99	348.11	35.00	0.98	1.00	0.99	350.72	35.86

Table 2.9: *Experimental results regarding Heuristic H2 .*

radius has proved to help TANGO learn rules that discern well amongst positive and negative examples that are very near in the DOM tree. Furthermore, selecting a small percentage of the remaining negative examples helps it produce rules that are general enough to make a difference amongst the positive examples and others that are very far away in the DOM tree. This alternative is a bit worse regarding precision and the F_1 score than the baseline; it behaves similarly in terms of recall and extraction time, but improves very much in terms of learning time since it is 8.86 ± 123.45 minutes faster. Alternative A4 got a rank that is close to the rank of the baseline but it is still a bit worse. Neither A1 nor A3 produced better results than the baseline.



Best alternative from Heuristic H1							
	A0						
Summary	P	R	F1	LT	ET	FR	K
Mean	0.98	0.96	0.97	38.88	4.47	-	0.50
Std. Dev.	0.02	0.05	0.04	63.79	5.53		
MDR	39.76	16.88	24.31	23.69	3.61		

Heuristic H2														
	A1							A2						
Summary	P	R	F1	LT	ET	FR	K	P	R	F1	LT	ET	FR	K
Mean	0.96	0.97	0.95	29.62	4.19	-	0.36	0.97	0.96	0.96	30.02	4.23	-	0.51
Std. Dev.	0.09	0.05	0.08	59.36	5.41			0.05	0.06	0.05	59.66	5.43		
MDR	10.42	20.45	11.27	14.78	3.25			20.15	16.85	18.83	15.10	3.30		
	A3							A4						
Summary	P	R	F1	LT	ET	FR	K	P	R	F1	LT	ET	FR	K
Mean	0.86	0.94	0.86	28.82	3.90	0.02	0.28	0.97	0.96	0.96	56.42	4.05	-	0.42
Std. Dev.	0.19	0.15	0.19	73.73	5.27			0.06	0.06	0.06	169.82	5.34		
MDR	3.94	5.85	3.98	11.26	2.89			16.72	15.08	15.97	18.75	3.07		

Table 2.10: *Ranking of alternatives regarding Heuristic H2 .*

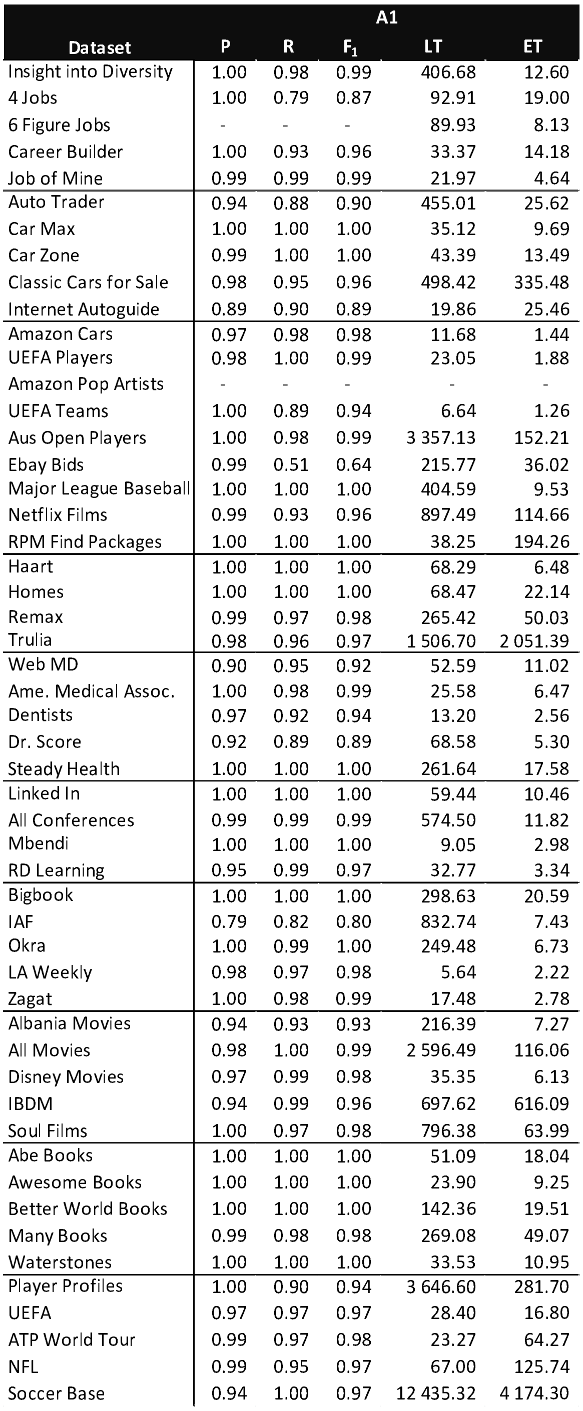
Heuristic H3: binarise features. When TANGO selects a feature instantiator to extend the current rule, it basically makes a blind decision: thus far, that instantiator is the best condition that can be added to the rule, but the value of the feature is not constrained at all; if necessary, it can be constrained later by adding a comparator to the rule. In other words, constraining the value of a feature is a two-step procedure. Binarising features is a process by means of which a single step suffices to instantiate a feature and constraint its value. Obviously, only attributive features can be binarised because they are the only ones that provide values that can be constrained by means of comparators. The binarisation process works as follows: a discrete feature f that ranges over the set of values $\{v_1, v_2, \dots, v_n\}$ is transformed into a collection of new features of the form $f\_v_1, f\_v_2, \dots, f\_v_n$; simply put, $f\_v_i(N)$ is satisfied as long as $f(N, A), A = v_i$ is satisfied, where N denotes a variable that ranges over the set of nodes, A is a variable that ranges over the set of values of feature f, and i ranges in interval $1 \mathinner{..} n$, where n is the number of different values that feature f can take. A numeric feature f that ranges over the set of values $\{v_1, v_2, \dots, v_n\}$ is transformed into a collection of new features of the form $f\_\theta\_v_1, f\_\theta\_v_2, \dots, f\_\theta\_v_n$, where θ represents a comparison operator; simply put, $f\_\theta\_v_i(N)$ is satisfied as long as $f(N, A), A\,\theta\,v_i$ is satisfied, where N denotes a variable that ranges over the set of nodes, A is a variable that ranges over the set of values of feature f, θ is a comparison operator, and i ranges in interval $1 \mathinner{..} n$, where n is the number of different values that feature f can take. Recall from Section §2.2.1 that Boolean features are represented in a compact form that binarises them by default.

The alternatives that we have considered regarding this heuristic are the following: A0) Work with the original features. A1) Binarise them.

Discussion. The empirical results are provided in Table §2.11 and the ranks are shown in Table §2.12. To our surprise, the ranks prove that the best alternative is the baseline, that is, not binarising the features, since it has proven to be effective and efficient. Regarding effectiveness, the precision of the baseline is 0.03 ± 0.24 higher, its recall is 0.04 ± 0.26 higher, and its F_1 score is 0.03 ± 0.25 higher. Regarding efficiency, the differences are more significant: alternative A1 is 587.76 ± 1 889.52 minutes slower with regard to alternative A0 when learning rules and 165.00 ± 644.64 minutes slower when the rules are executed. We found several explanations for that behaviour, namely: a) the number of features to be considered grows dramatically, which leads to extremely large learning sets that are very costly to process; b) when alternative A0 is used, TANGO typically selects several determinate conditions in the first iteration, that is, several feature instantiators, and the corresponding feature values are typically constrained in the forthcoming iterations by means of comparators. This means that the number of comparators that are explored and evaluated depends on the number of feature instantiators that were added to the rule in the previous iteration. When binarisation is used, a feature instantiator both instantiates a feature and constrains its values at the same time, which means that many features that are not promising at all must be considered in each iteration. Consequently, the number of candidates to explore and evaluate in each iteration grows significantly when binarisation is used. Furthermore, the standard deviation of the performance measures is smaller regarding alternative A0, which means that it is generally more stable than alternative A1. Furthermore, note that alternative A1 failed in a few cases because it was unable to find a proper set of rules. Thus, our conclusion is that the intuition behind binarising features fails in our context and that the best alternative is to use the features as they are provided in the corresponding catalogue.

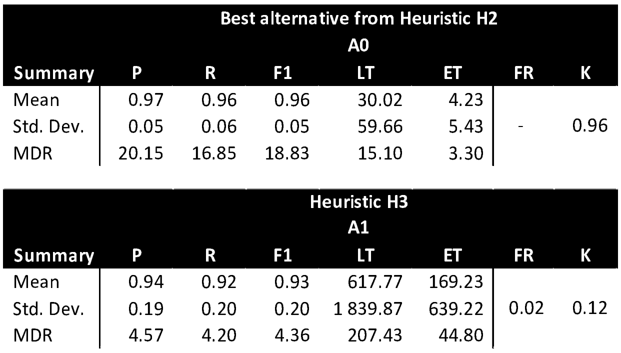
2.3.3 Variation point isPromisingCandidate

This variation point deals with determining if a candidate is good enough to stop the search for new candidates in each iteration. It consists of Heuristic H4 only. The goal is to customise TANGO so that it can learn rules more efficiently without degrading their effectiveness or increasing the time required to execute them.



	A1				
Dataset	P	R	F_1	LT	ET
Insight into Diversity	1.00	0.98	0.99	406.68	12.60
4 Jobs	1.00	0.79	0.87	92.91	19.00
6 Figure Jobs	-	-	-	89.93	8.13
Career Builder	1.00	0.93	0.96	33.37	14.18
Job of Mine	0.99	0.99	0.99	21.97	4.64
Auto Trader	0.94	0.88	0.90	455.01	25.62
Car Max	1.00	1.00	1.00	35.12	9.69
Car Zone	0.99	1.00	1.00	43.39	13.49
Classic Cars for Sale	0.98	0.95	0.96	498.42	335.48
Internet Autoguide	0.89	0.90	0.89	19.86	25.46
Amazon Cars	0.97	0.98	0.98	11.68	1.44
UEFA Players	0.98	1.00	0.99	23.05	1.88
Amazon Pop Artists	-	-	-	-	-
UEFA Teams	1.00	0.89	0.94	6.64	1.26
Aus Open Players	1.00	0.98	0.99	3 357.13	152.21
Ebay Bids	0.99	0.51	0.64	215.77	36.02
Major League Baseball	1.00	1.00	1.00	404.59	9.53
Netflix Films	0.99	0.93	0.96	897.49	114.66
RPM Find Packages	1.00	1.00	1.00	38.25	194.26
Haart	1.00	1.00	1.00	68.29	6.48
Homes	1.00	1.00	1.00	68.47	22.14
Remax	0.99	0.97	0.98	265.42	50.03
Trulia	0.98	0.96	0.97	1 506.70	2 051.39
Web MD	0.90	0.95	0.92	52.59	11.02
Ame. Medical Assoc.	1.00	0.98	0.99	25.58	6.47
Dentists	0.97	0.92	0.94	13.20	2.56
Dr. Score	0.92	0.89	0.89	68.58	5.30
Steady Health	1.00	1.00	1.00	261.64	17.58
Linked In	1.00	1.00	1.00	59.44	10.46
All Conferences	0.99	0.99	0.99	574.50	11.82
Mbendi	1.00	1.00	1.00	9.05	2.98
RD Learning	0.95	0.99	0.97	32.77	3.34
Bigbook	1.00	1.00	1.00	298.63	20.59
IAF	0.79	0.82	0.80	832.74	7.43
Okra	1.00	0.99	1.00	249.48	6.73
LA Weekly	0.98	0.97	0.98	5.64	2.22
Zagat	1.00	0.98	0.99	17.48	2.78
Albania Movies	0.94	0.93	0.93	216.39	7.27
All Movies	0.98	1.00	0.99	2 596.49	116.06
Disney Movies	0.97	0.99	0.98	35.35	6.13
IBDM	0.94	0.99	0.96	697.62	616.09
Soul Films	1.00	0.97	0.98	796.38	63.99
Abe Books	1.00	1.00	1.00	51.09	18.04
Awesome Books	1.00	1.00	1.00	23.90	9.25
Better World Books	1.00	1.00	1.00	142.36	19.51
Many Books	0.99	0.98	0.98	269.08	49.07
Waterstones	1.00	1.00	1.00	33.53	10.95
Player Profiles	1.00	0.90	0.94	3 646.60	281.70
UEFA	0.97	0.97	0.97	28.40	16.80
ATP World Tour	0.99	0.97	0.98	23.27	64.27
NFL	0.99	0.95	0.97	67.00	125.74
Soccer Base	0.94	1.00	0.97	12 435.32	4 174.30

Table 2.11: *Experimental results regarding Heuristic H3 .*



Best alternative from Heuristic H2							
A0							
Summary	P	R	F1	LT	ET	FR	K
Mean	0.97	0.96	0.96	30.02	4.23	-	0.96
Std. Dev.	0.05	0.06	0.05	59.66	5.43		
MDR	20.15	16.85	18.83	15.10	3.30		

Heuristic H3							
A1							
Summary	P	R	F1	LT	ET	FR	K
Mean	0.94	0.92	0.93	617.77	169.23	0.02	0.12
Std. Dev.	0.19	0.20	0.20	1 839.87	639.22		
MDR	4.57	4.20	4.36	207.43	44.80		

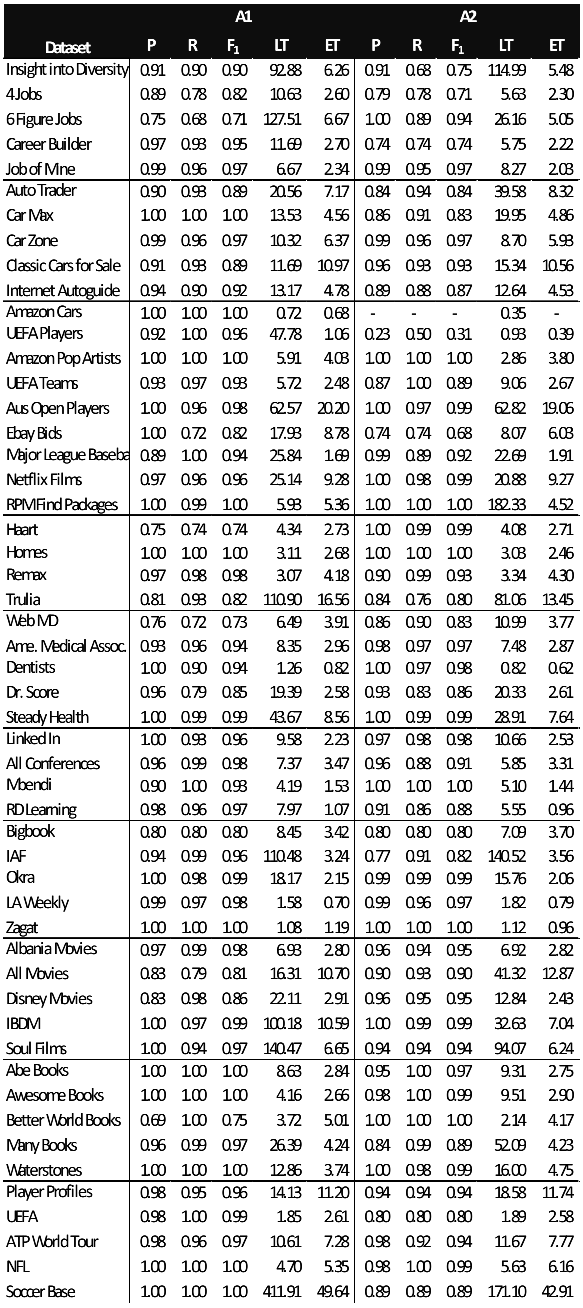
Table 2.12: *Ranking of alternatives regarding Heuristic H3 .*

Heuristic H4: check promising candidates. TANGO always branches the current rule to compute a collection of candidate conditions. Then, it iterates on it to compute the gain that each condition achieves and whether it is determinate or not, which requires time. If there is a heuristic that allows to stop the computation of candidates when a very promising condition is found, then it might help save much time.

The alternatives to implement this heuristic are the following: A0) Stop the search only if we find a new condition that leads to a rule that matches all of the positive examples matched by the current rule, but discards all of the negative examples that it matches. A1) Stop the search when the gain achieved by a condition is at least 80% the maximum possible gain on the current rule. A2) Like alternative A1, but we also require the resulting rule to be a solution.

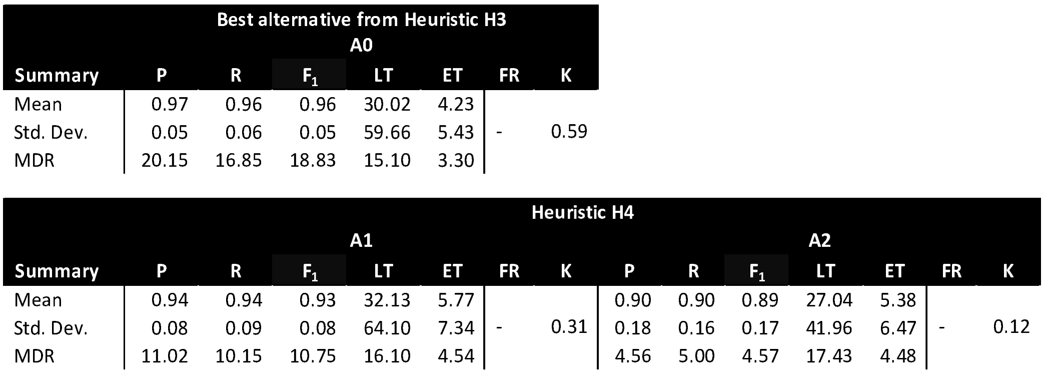
Alternative A0 considers almost every condition as a candidate to extend the current rule or to create a savepoint; it only stops when the best solution is found. The rationale behind alternative A1 is that Heuristic H1 selects a candidate whose gain is at least 80% the maximum possible gain on the current rule; thus, we might stop the search for candidates when such a candidate is found; the only problem would be that if that candidate proved not to be good enough in the forthcoming iterations, then we would miss some better candidates that would have been evaluated later in the same iteration. Alternative A2 is a bit more demanding since it requires a candidate that result in a solution so that the search can be stopped.

Discussion. The empirical results are presented in Table §2.13 and the ranks are shown in Table §2.14. They suggest that the best alternative to implement



	A1					A2				
Dataset	P	R	F_1	LT	ET	P	R	F_1	LT	ET
Insight into Diversity	0.91	0.90	0.90	92.88	6.26	0.91	0.68	0.75	114.99	5.48
4 Jobs	0.89	0.78	0.82	10.63	2.60	0.79	0.78	0.71	5.63	2.30
6 Figure Jobs	0.75	0.68	0.71	127.51	6.67	1.00	0.89	0.94	26.16	5.05
Career Builder	0.97	0.93	0.95	11.69	2.70	0.74	0.74	0.74	5.75	2.22
Job of Mne	0.99	0.96	0.97	6.67	2.34	0.99	0.95	0.97	8.27	2.03
Auto Trader	0.90	0.93	0.89	20.56	7.17	0.84	0.94	0.84	39.58	8.32
Car Max	1.00	1.00	1.00	13.53	4.56	0.86	0.91	0.83	19.95	4.86
Car Zone	0.99	0.96	0.97	10.32	6.37	0.99	0.96	0.97	8.70	5.93
Classic Cars for Sale	0.91	0.93	0.89	11.69	10.97	0.96	0.93	0.93	15.34	10.56
Internet Autoguide	0.94	0.90	0.92	13.17	4.78	0.89	0.88	0.87	12.64	4.53
Amazon Cars	1.00	1.00	1.00	0.72	0.68	-	-	-	0.35	-
UEFA Players	0.92	1.00	0.96	47.78	1.06	0.23	0.50	0.31	0.93	0.39
Amazon Pop Artists	1.00	1.00	1.00	5.91	4.03	1.00	1.00	1.00	2.86	3.80
UEFA Teams	0.93	0.97	0.93	5.72	2.48	0.87	1.00	0.89	9.06	2.67
Aus Open Players	1.00	0.96	0.98	62.57	20.20	1.00	0.97	0.99	62.82	19.06
Ebay Bids	1.00	0.72	0.82	17.93	8.78	0.74	0.74	0.68	8.07	6.03
Major League Baseba	0.89	1.00	0.94	25.84	1.69	0.99	0.89	0.92	22.69	1.91
Netflix Films	0.97	0.96	0.96	25.14	9.28	1.00	0.98	0.99	20.88	9.27
RPMFind Packages	1.00	0.99	1.00	5.93	5.36	1.00	1.00	1.00	182.33	4.52
Haart	0.75	0.74	0.74	4.34	2.73	1.00	0.99	0.99	4.08	2.71
Homes	1.00	1.00	1.00	3.11	2.68	1.00	1.00	1.00	3.03	2.46
Remax	0.97	0.98	0.98	3.07	4.18	0.90	0.99	0.93	3.34	4.30
Trulia	0.81	0.93	0.82	110.90	16.56	0.84	0.76	0.80	81.06	13.45
Web MD	0.76	0.72	0.73	6.49	3.91	0.86	0.90	0.83	10.99	3.77
Ame. Medical Assoc.	0.93	0.96	0.94	8.35	2.96	0.98	0.97	0.97	7.48	2.87
Dentists	1.00	0.90	0.94	1.26	0.82	1.00	0.97	0.98	0.82	0.62
Dr. Score	0.96	0.79	0.85	19.39	2.58	0.93	0.83	0.86	20.33	2.61
Steady Health	1.00	0.99	0.99	43.67	8.56	1.00	0.99	0.99	28.91	7.64
Linked In	1.00	0.93	0.96	9.58	2.23	0.97	0.98	0.98	10.66	2.53
All Conferences	0.96	0.99	0.98	7.37	3.47	0.96	0.88	0.91	5.85	3.31
Mbendi	0.90	1.00	0.93	4.19	1.53	1.00	1.00	1.00	5.10	1.44
RD Learning	0.98	0.96	0.97	7.97	1.07	0.91	0.86	0.88	5.55	0.96
Bigbook	0.80	0.80	0.80	8.45	3.42	0.80	0.80	0.80	7.09	3.70
IAF	0.94	0.99	0.96	110.48	3.24	0.77	0.91	0.82	140.52	3.56
Okra	1.00	0.98	0.99	18.17	2.15	0.99	0.99	0.99	15.76	2.06
LA Weekly	0.99	0.97	0.98	1.58	0.70	0.99	0.96	0.97	1.82	0.79
Zagat	1.00	1.00	1.00	1.08	1.19	1.00	1.00	1.00	1.12	0.96
Albania Movies	0.97	0.99	0.98	6.93	2.80	0.96	0.94	0.95	6.92	2.82
All Movies	0.83	0.79	0.81	16.31	10.70	0.90	0.93	0.90	41.32	12.87
Disney Movies	0.83	0.98	0.86	22.11	2.91	0.96	0.95	0.95	12.84	2.43
IBDM	1.00	0.97	0.99	100.18	10.59	1.00	0.99	0.99	32.63	7.04
Soul Films	1.00	0.94	0.97	140.47	6.65	0.94	0.94	0.94	94.07	6.24
Abe Books	1.00	1.00	1.00	8.63	2.84	0.95	1.00	0.97	9.31	2.75
Awesome Books	1.00	1.00	1.00	4.16	2.66	0.98	1.00	0.99	9.51	2.90
Better World Books	0.69	1.00	0.75	3.72	5.01	1.00	1.00	1.00	2.14	4.17
Many Books	0.96	0.99	0.97	26.39	4.24	0.84	0.99	0.89	52.09	4.23
Waterstones	1.00	1.00	1.00	12.86	3.74	1.00	0.98	0.99	16.00	4.75
Player Profiles	0.98	0.95	0.96	14.13	11.20	0.94	0.94	0.94	18.58	11.74
UEFA	0.98	1.00	0.99	1.85	2.61	0.80	0.80	0.80	1.89	2.58
ATP World Tour	0.98	0.96	0.97	10.61	7.28	0.98	0.92	0.94	11.67	7.77
NFL	1.00	1.00	1.00	4.70	5.35	0.98	1.00	0.99	5.63	6.16
Soccer Base	1.00	1.00	1.00	411.91	49.64	0.89	0.89	0.89	171.10	42.91

Table 2.13: *Experimental results regarding Heuristic H4 .*



	Best alternative from Heuristic H3						
	A0						
Summary	P	R	F_1	LT	ET	FR	K
Mean	0.97	0.96	0.96	30.02	4.23		
Std. Dev.	0.05	0.06	0.05	59.66	5.43	-	0.59
MDR	20.15	16.85	18.83	15.10	3.30		

	Heuristic H4													
	A1							A2						
Summary	P	R	F_1	LT	ET	FR	K	P	R	F_1	LT	ET	FR	K
Mean	0.94	0.94	0.93	32.13	5.77			0.90	0.90	0.89	27.04	5.38		
Std. Dev.	0.08	0.09	0.08	64.10	7.34	-	0.31	0.18	0.16	0.17	41.96	6.47	-	0.12
MDR	11.02	10.15	10.75	16.10	4.54			4.56	5.00	4.57	17.43	4.48		

Table 2.14: *Ranking of alternatives regarding Heuristic H4 .*

this heuristic is the default one. Alternative A1 has 0.03 ± 0.13 less precision, 0.03 ± 0.14 less recall, and 0.03 ± 0.13 less F_1 score than alternative A0; furthermore, its learning time is 2.11 ± 123.76 minutes worse and the extraction time is 1.54 ± 12.77 minutes worse. We found out that, typically, the first candidate that exceeds the selected threshold is not actually the best one, and that it is common that some candidates that might have been explored later provide more gain and result in better rules; unfortunately, this alternative prevents TANGO from finding them. In other words, the number of candidates that are explored in each iteration is smaller, but the total number of iterations increases; this contributes to increasing the learning time and producing more specific rules that are not likely to work well with new unseen documents. Neither did alternative A2 perform better: it was able to learn solutions faster than alternative A0, exactly 2.98 ± 101.62 minutes faster; unfortunately, the resulting rule sets were larger since the individual rules learnt were more specific, which worsened the extraction time by 1.15 ± 11.89 minutes. Notice, too, that alternative A0 is the most stable one since it achieves the lowest deviations. As a conclusion, alternative A0 is the best one to implement this heuristic.

We think that the more stringent the criterion to select a promising candidate, the better the effectiveness. However, we did not manage to find a criterion that could improve on the baseline because if it is very stringent, then the behaviour of TANGO is similar to the baseline, i.e., it tends to perform an exhaustive search.

2.3.4 Variation point RULESCORER

This variation point deals with assessing rules. It consists of Heuristic H5 , which implements a function that returns a real number; generally speaking, the smaller the score, the worse the rule and the higher the score, the better the rule. The goal is to customise TANGO so that it learns extraction rules that are more effective and efficient to execute.

Heuristic H5: compute scores. The heuristics that TANGO uses to select which candidates must be used to extend the current rule or to create new savepoints are based on the gain provided by the corresponding condition to the current rule. The gain is computed using a standard formula cf. Definition §2.7 that relies on a scoring function that assesses how good a rule is in the context of a given learning set. Typical methods to compute the score of a rule are based on confusion matrices of the form $(\mathrm{tp}, \mathrm{fp}, \mathrm{tn}, \mathrm{fn})$, where tp denotes the number of true positives, fp the number of false positives, tn the number of true negatives, and fn the number of false negatives.

The alternatives that we have tried to implement this heuristic are the following, where $p = \mathrm{tp} + \mathrm{fn}$, $n = \mathrm{tn} + \mathrm{fp}$, and $N = \mathrm{tp} + \mathrm{fp} + \mathrm{tn} + \mathrm{fn}$: A0) Information Content, which computes the score as $\log \frac{\mathrm{tp}}{\mathrm{tp}+\mathrm{fp}}$. A1) Accuracy-based Information Content, which computes it as $\log \frac{\mathrm{tp}+\mathrm{tn}}{\mathrm{tp}+\mathrm{fp}+\mathrm{fn}+\mathrm{tn}}$. A2) Satisfaction, which computes it as $\frac{\frac{\mathrm{tp}}{\mathrm{tp}+\mathrm{fp}} - \frac{\mathrm{tp}+\mathrm{fn}}{N}}{1 - \frac{\mathrm{tp}+\mathrm{fn}}{N}}$. A3) Laplace Estimate, which computes it as $\frac{\mathrm{tp}+1}{\mathrm{tp}+\mathrm{fp}+2}$. A4) Piatetski-Shapiro's measure, which computes it as $\frac{\mathrm{tp}\,\mathrm{tn} - \mathrm{fp}\,\mathrm{fn}}{N^2}$.

All of these alternatives are well-known functions that have been taken from the literature. Alternative A0 computes the score as the logarithm of precision, which heavily penalises the conditions that result in a large loss of positive examples matched. Alternative A1 is similar, but it computes the logarithm of accuracy and then takes the number of false negatives and true negatives into account. Intuitively, the higher the number of true positive and true negatives, the better; contrarily, the higher the number of false positive and false negatives, the worse. Satisfaction reaches its maximum when precision is close to 1.00 and decreases steadily as the number of positive examples matched decreases. Alternative A3 penalises rules that match few positive examples; if a rule does not match any examples, then the result is 0.50, which is as effective as a random guess; contrarily, if it matches many examples, it tends to the precision. Finally, A4, tends to give higher scores when the number of true positives and true negatives is higher than the number of false negatives and false positives, respectively.

	A1					A2					A3					A4				
Dataset	P	R	F_1	LT	ET	P	R	F_1	LT	ET	P	R	F_1	LT	ET	P	R	F_1	LT	ET
Insight into Diversity	0.90	0.79	0.84	50.39	4.44	0.96	0.77	0.84	119.30	7.46	0.98	0.90	0.93	100.27	6.28	0.75	0.69	0.72	28.61	2.58
4 Jobs	0.93	0.78	0.83	4.77	2.02	0.88	0.78	0.82	12.20	2.94	0.89	0.84	0.85	11.71	2.99	0.97	0.76	0.85	3.18	2.62
6 Figure Jobs	0.97	0.90	0.93	12.52	5.76	0.76	0.89	0.70	158.16	8.14	1.00	0.89	0.94	153.04	7.87	1.00	0.89	0.94	43.04	5.02
Career Builder	0.85	0.85	0.84	8.22	2.67	0.93	0.93	0.93	22.39	2.92	0.82	0.80	0.81	20.14	2.92	0.41	0.43	0.42	3.28	2.19
Job of Mine	0.99	0.86	0.91	10.26	2.32	0.99	0.98	0.98	7.30	2.29	0.99	0.98	0.98	9.78	2.27	-	-	-	0.82	-
Auto Trader	0.89	0.94	0.88	13.36	7.49	0.89	0.84	0.85	105.35	10.41	0.95	0.94	0.94	82.75	9.72	-	-	-	2.00	-
Car Max	0.94	1.00	0.96	11.02	4.62	1.00	1.00	1.00	19.67	5.09	1.00	1.00	1.00	22.62	5.19	-	-	-	1.18	-
Car Zone	0.99	0.98	0.98	5.33	5.57	0.97	0.97	0.97	11.25	6.59	0.95	0.96	0.95	10.57	6.61	-	-	-	1.54	-
Classic Cars for Sale	1.00	0.97	0.98	14.51	11.31	0.98	0.94	0.96	16.15	11.96	0.92	0.97	0.92	17.83	11.83	0.95	0.83	0.87	8.86	12.04
Internet Autoguide	0.88	0.93	0.90	15.29	5.04	0.89	0.92	0.88	29.04	5.63	0.94	0.94	0.94	27.03	5.46	0.88	0.93	0.90	7.70	4.72
Amazon Cars	1.00	1.00	1.00	0.63	0.50	1.00	1.00	1.00	0.81	0.72	1.00	1.00	1.00	0.83	0.72	1.00	1.00	1.00	0.63	0.51
UEFA Players	0.91	1.00	0.95	14.88	0.73	0.91	1.00	0.95	24.15	0.80	0.91	0.99	0.94	22.42	0.80	0.91	1.00	0.95	9.79	0.75
Amazon Pop Artists	1.00	1.00	1.00	5.61	4.18	1.00	1.00	1.00	5.66	4.19	1.00	1.00	1.00	7.60	4.14	1.00	1.00	1.00	19.26	4.43
UEFA Teams	1.00	1.00	1.00	5.60	2.43	0.93	1.00	0.94	6.23	2.50	0.93	1.00	0.94	6.30	2.53	1.00	1.00	1.00	4.92	2.41
Aus Open Players	0.99	0.96	0.98	48.39	21.32	0.99	0.97	0.98	137.46	22.95	0.99	0.97	0.98	129.25	22.23	1.00	0.95	0.97	118.84	21.39
Ebay Bids	0.87	0.85	0.82	6.39	5.92	1.00	0.72	0.82	32.95	11.32	1.00	0.72	0.82	36.16	11.70	0.57	0.83	0.64	8.96	5.96
Major League Baseball	0.99	0.88	0.91	27.27	1.72	0.90	1.00	0.94	36.07	1.69	0.89	0.87	0.85	29.97	1.73	1.00	0.87	0.91	21.27	1.52
Netflix Films	0.92	0.95	0.92	103.40	9.18	1.00	0.96	0.98	53.45	9.72	1.00	0.96	0.98	54.15	9.72	1.00	0.91	0.95	48.53	9.05
RPM Find Packages	1.00	0.99	1.00	2.98	3.89	1.00	0.99	1.00	5.95	4.89	1.00	0.99	1.00	6.22	5.04	0.75	0.74	0.75	4.62	3.90
Haart	0.83	1.00	0.88	1.79	2.72	1.00	0.99	0.99	5.09	2.84	1.00	0.99	0.99	3.67	2.89	-	-	-	0.77	-
Homes	0.94	1.00	0.97	2.49	2.51	1.00	1.00	1.00	4.65	2.60	1.00	1.00	1.00	2.79	2.64	1.00	1.00	1.00	4.02	2.61
Remax	0.92	1.00	0.94	3.49	4.37	0.99	0.98	0.99	3.47	4.19	0.99	0.98	0.99	3.44	4.12	0.58	0.65	0.60	2.19	3.98
Trulia	0.94	0.90	0.92	16.11	10.27	0.96	0.93	0.94	129.52	19.17	0.82	0.91	0.79	117.69	18.98	0.96	0.94	0.95	75.82	15.14
Web MD	0.98	0.93	0.95	5.30	4.12	0.80	0.93	0.83	7.24	4.34	0.84	0.93	0.86	7.47	4.29	-	-	-	1.11	-
Ame. Medical Assoc.	1.00	0.98	0.99	3.49	2.68	1.00	0.96	0.98	10.34	3.43	1.00	0.96	0.98	9.90	3.49	0.80	0.77	0.78	6.31	2.59
Dentists	1.00	1.00	1.00	1.04	0.79	1.00	0.92	0.95	1.25	0.84	1.00	0.92	0.95	1.22	0.87	1.00	0.92	0.95	1.24	0.89
Dr. Score	0.88	0.73	0.76	14.92	2.41	0.97	0.85	0.89	21.90	2.73	0.95	0.84	0.87	21.19	2.68	0.89	0.74	0.78	15.76	2.44
Steady Health	1.00	1.00	1.00	25.85	8.48	1.00	1.00	1.00	41.88	8.54	1.00	1.00	1.00	46.60	8.51	0.50	0.50	0.50	20.30	4.83
Linked In	1.00	0.98	0.99	4.84	2.42	1.00	0.99	1.00	14.20	2.71	0.99	0.99	0.99	13.76	2.78	1.00	0.99	1.00	4.16	2.39
All Conferences	0.96	0.99	0.98	7.35	3.22	0.96	0.99	0.98	9.89	3.51	0.96	0.99	0.98	10.67	3.44	0.96	0.98	0.97	16.23	2.99
Mbendi	0.85	1.00	0.88	3.71	1.45	0.90	1.00	0.93	4.15	1.67	0.90	1.00	0.93	4.92	1.63	-	-	-	0.28	-
RD Learning	0.98	1.00	0.99	7.88	0.99	0.98	1.00	0.99	8.93	1.01	0.98	1.00	0.99	7.59	1.00	-	-	-	0.22	-
Bigbook	1.00	1.00	1.00	6.72	4.14	1.00	1.00	1.00	10.75	4.65	1.00	1.00	1.00	12.55	4.55	0.80	0.80	0.80	10.11	3.38
IAF	0.95	0.99	0.97	65.71	3.30	0.94	0.99	0.96	155.21	3.70	0.94	0.99	0.96	146.21	3.43	0.83	0.82	0.83	71.65	2.64
Okra	1.00	1.00	1.00	5.58	1.99	0.99	0.99	0.99	31.72	2.21	1.00	0.99	0.99	31.95	2.22	0.67	0.66	0.66	15.80	1.70
LA Weekly	0.99	0.97	0.98	1.43	0.68	0.99	0.97	0.98	1.63	0.75	0.99	0.97	0.98	1.59	0.74	0.66	0.67	0.66	1.56	0.42
Zagat	1.00	1.00	1.00	0.95	1.20	1.00	1.00	1.00	1.22	1.28	1.00	1.00	1.00	1.20	1.28	0.70	0.75	0.72	0.88	0.90
Albania Movies	0.96	1.00	0.98	3.85	2.88	0.98	0.97	0.97	26.88	3.21	1.00	0.95	0.97	24.13	3.10	0.83	0.80	0.81	8.43	2.39
All Movies	1.00	0.97	0.99	15.64	12.28	0.84	0.94	0.85	65.01	15.13	1.00	0.93	0.96	65.34	14.52	-	-	-	0.68	-
Disney Movies	0.98	0.99	0.98	15.63	2.62	0.80	0.99	0.85	22.29	2.90	0.83	0.99	0.87	24.60	2.93	-	-	-	0.61	-
IBDM	0.94	0.99	0.96	13.66	7.62	0.97	0.99	0.98	151.34	11.11	0.94	0.99	0.96	114.56	10.85	0.94	0.99	0.96	87.21	7.54
Soul Films	1.00	0.97	0.98	309.97	6.02	1.00	0.94	0.97	179.30	7.16	1.00	0.94	0.97	176.35	6.85	1.00	0.98	0.99	102.98	6.39
Abe Books	1.00	1.00	1.00	2.95	2.68	1.00	1.00	1.00	4.75	3.01	1.00	1.00	1.00	8.22	3.02	-	-	-	1.36	-
Awesome Books	0.99	1.00	1.00	13.38	2.67	1.00	1.00	1.00	4.21	2.87	1.00	1.00	1.00	4.64	2.88	-	-	-	0.90	-
Better World Books	1.00	1.00	1.00	5.66	4.84	0.98	1.00	0.99	4.48	5.33	0.98	1.00	0.99	4.71	5.32	1.00	1.00	1.00	5.12	3.71
Many Books	0.98	0.98	0.98	8.69	3.16	0.98	0.99	0.98	28.99	4.60	0.97	0.99	0.98	29.35	4.52	0.73	0.73	0.73	11.54	3.08
Waterstones	1.00	1.00	1.00	4.83	3.53	1.00	1.00	1.00	9.40	3.85	1.00	0.99	1.00	12.56	3.93	1.00	0.99	1.00	17.30	3.96
Player Profiles	1.00	1.00	1.00	14.64	10.25	0.96	0.94	0.94	20.49	13.04	0.96	0.94	0.94	20.58	12.20	0.99	0.98	0.99	21.17	11.85
UEFA	1.00	1.00	1.00	1.60	1.94	1.00	1.00	1.00	2.19	2.83	1.00	1.00	1.00	1.86	2.72	0.99	1.00	1.00	1.57	2.01
ATP World Tour	0.99	0.99	0.99	12.86	5.84	0.96	0.97	0.97	17.07	7.90	0.94	0.97	0.96	15.93	7.59	0.90	0.97	0.93	17.18	7.66
NFL	0.92	1.00	0.94	2.24	5.65	1.00	1.00	1.00	3.50	5.72	0.96	1.00	0.98	4.06	5.59	1.00	1.00	1.00	2.28	5.66
Soccer Base	0.97	1.00	0.98	96.01	42.97	0.87	0.89	0.88	567.75	55.51	0.75	0.77	0.76	572.08	48.32	0.78	0.77	0.78	283.94	39.50

Table 2.15: *Experimental results regarding Heuristic H5 .*

Discussion. The empirical results are presented in Table §2.15 and the ranks are shown in Table §2.16. Regarding effectiveness, none of the alternatives that we have analysed can beat the baseline, but some of them achieve results that are very similar. This means that using the logarithm of the precision of a rule contributes positively to the overall performance of our system. Accuracy-based Information Content is the one that achieved the best results below the baseline, but the baseline provides 0.01 ± 0.10 more precision and 0.01 ± 0.11 more F_1 score. Regarding efficiency, alternative A1 proved to learn 9.61 ± 105.80 minutes faster, but it was 1.03 ± 11.91 minutes slower when executing the rules. Laplace ranks at the third place, which makes sense, since it

Best alternative from Heuristic H4							
	A0						
Summary	P	R	F_1	LT	ET	FR	K
Mean	0.97	0.96	0.96	30.02	4.23		
Std. Dev.	0.05	0.06	0.05	59.66	5.43	-	0.68
MDR	20.15	16.85	18.83	15.10	3.30		

Heuristic H5														
	A1							A2						
Summary	P	R	F_1	LT	ET	FR	K	P	R	F_1	LT	ET	FR	K
Mean	0.96	0.96	0.95	20.40	5.27			0.96	0.96	0.95	45.66	6.47		
Std. Dev.	0.05	0.06	0.06	46.14	6.48	-	0.64	0.06	0.06	0.07	88.39	8.35	-	0.37
MDR	19.05	14.45	15.70	9.02	4.28			15.31	14.31	13.62	23.58	5.02		
	A3							A4						
Summary	P	R	F_1	LT	ET	FR	K	P	R	F_1	LT	ET	FR	K
Mean	0.96	0.95	0.95	43.69	6.22			0.67	0.66	0.66	22.07	4.23		
Std. Dev.	0.06	0.06	0.06	86.80	7.47	-	0.42	0.40	0.39	0.39	45.89	6.51	-	0.29
MDR	15.78	14.44	14.98	21.99	5.19			1.13	1.12	1.12	10.62	2.74		

Table 2.16: *Ranking of alternatives regarding Heuristic H5 .*

is also based on precision, like Information Content. However, one can easily realise the advantage of using the logarithmic function, chiefly in both learning and extraction times. Satisfaction is not bad regarding effectiveness, but it is similar to the baseline regarding effectiveness and it is 15.64 ± 105.54 minutes slower regarding learning time and 2.24 ± 11.94 minutes slower regarding extraction time. The Piatetski Shapiro's alternative is of little interest because it is clearly the worst one.

Our conclusion is that, except for the last alternative, all of the alternatives reach good results regarding effectiveness. However, the differences in efficiency are more remarkable. According to our intuition and our K rank, it makes sense to select Information Content as our rule scorer since it seems to be the best alternative, followed by Accuracy-based Information Content. There is not a clear reason why some rule scorers performed better than the others, but, in most cases, they all were able to guide the search properly. That is, the system was able to find perfect rules that matched the whole set of positive examples in the learning sets. However, it seems that the choice of some conditions during the learning phase had an impact on the testing phase and some of the candidates selected by some alternatives were not general enough to extract the information from the testing sets. This caused a penalty to precision and/or recall, which made a difference in some datasets because the system was unable to extract all of the positive examples or it extracted some more negative examples.

2.3.5 Variation point BOUND

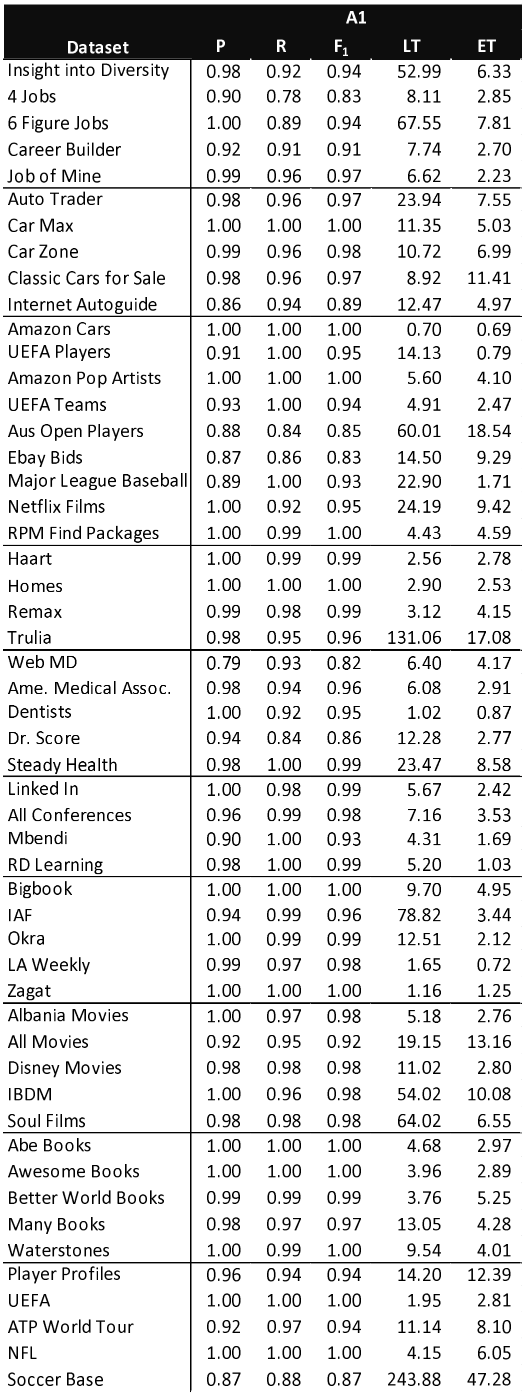
This variation point deals with determining whether a condition deserves to be selected as a candidate to extend the current rule or to create a savepoint. It consists of a single heuristic to which we refer to as Heuristic H6 . The goal is to customise TANGO so that it learns extraction rules more efficiently without degrading their effectiveness or increasing the time required to execute them.

Heuristic H6: prune candidates. Typically, branching the current rule results in many conditions. TANGO has to select a subset to extend the current rule and a subset to create new savepoints. Having a heuristic that prunes some conditions so that they do not have to be analysed makes sense since we guessed that this would save some time.

The alternatives that we have devised to implement this heuristic are the following: A0) Do not prune at all, that is, every possible condition that results from branching a rule is considered as a candidate. A1) Prune every condition that does not result in a gain that is at least 80% the gain of the best condition found so far, unless it is a determinate condition; if a determinate condition is found, then the pruning threshold is changed to 80% the maximum gain that a condition can achieve on the current rule.

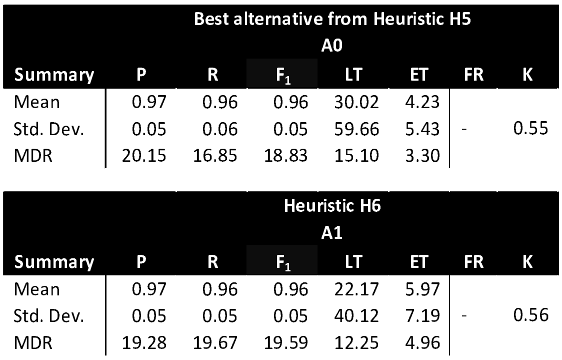
Alternative A0 is the safest one because it does not prune any of the conditions that result from branching the current rule; that is, the search is exhaustive. The second alternative prunes the conditions that do not achieve a high enough gain with regard to the best condition found so far; intuitively, one might think that every condition that does not exceed the gain of the first condition might be trivially discarded, but we guessed that this would be too stringent, not to mention that we need to keep a few ones to create savepoints later if they are proven to be promising. Note, however, that determinate conditions are never pruned because we already know that they help expand the search space and avoid local maxima. The key is that when such a condition is found, we know that there is at least a condition that can be used to extend the current rule. Actually, Heuristic H1 is not going to select any candidates that do not achieve 80% the maximum gain once a determinate condition is found, so we can make our heuristic much more stringent safely, and prune every non-determinate condition that cannot achieve the minimum gain.

Discussion. The empirical results are presented in Table §2.17 and the ranks are shown in Table §2.18. Regarding effectiveness, there is a tie, because there



	A1				
Dataset	P	R	F_1	LT	ET
Insight into Diversity	0.98	0.92	0.94	52.99	6.33
4 Jobs	0.90	0.78	0.83	8.11	2.85
6 Figure Jobs	1.00	0.89	0.94	67.55	7.81
Career Builder	0.92	0.91	0.91	7.74	2.70
Job of Mine	0.99	0.96	0.97	6.62	2.23
Auto Trader	0.98	0.96	0.97	23.94	7.55
Car Max	1.00	1.00	1.00	11.35	5.03
Car Zone	0.99	0.96	0.98	10.72	6.99
Classic Cars for Sale	0.98	0.96	0.97	8.92	11.41
Internet Autoguide	0.86	0.94	0.89	12.47	4.97
Amazon Cars	1.00	1.00	1.00	0.70	0.69
UEFA Players	0.91	1.00	0.95	14.13	0.79
Amazon Pop Artists	1.00	1.00	1.00	5.60	4.10
UEFA Teams	0.93	1.00	0.94	4.91	2.47
Aus Open Players	0.88	0.84	0.85	60.01	18.54
Ebay Bids	0.87	0.86	0.83	14.50	9.29
Major League Baseball	0.89	1.00	0.93	22.90	1.71
Netflix Films	1.00	0.92	0.95	24.19	9.42
RPM Find Packages	1.00	0.99	1.00	4.43	4.59
Haart	1.00	0.99	0.99	2.56	2.78
Homes	1.00	1.00	1.00	2.90	2.53
Remax	0.99	0.98	0.99	3.12	4.15
Trulia	0.98	0.95	0.96	131.06	17.08
Web MD	0.79	0.93	0.82	6.40	4.17
Ame. Medical Assoc.	0.98	0.94	0.96	6.08	2.91
Dentists	1.00	0.92	0.95	1.02	0.87
Dr. Score	0.94	0.84	0.86	12.28	2.77
Steady Health	0.98	1.00	0.99	23.47	8.58
Linked In	1.00	0.98	0.99	5.67	2.42
All Conferences	0.96	0.99	0.98	7.16	3.53
Mbendi	0.90	1.00	0.93	4.31	1.69
RD Learning	0.98	1.00	0.99	5.20	1.03
Bigbook	1.00	1.00	1.00	9.70	4.95
IAF	0.94	0.99	0.96	78.82	3.44
Okra	1.00	0.99	0.99	12.51	2.12
LA Weekly	0.99	0.97	0.98	1.65	0.72
Zagat	1.00	1.00	1.00	1.16	1.25
Albania Movies	1.00	0.97	0.98	5.18	2.76
All Movies	0.92	0.95	0.92	19.15	13.16
Disney Movies	0.98	0.98	0.98	11.02	2.80
IBDM	1.00	0.96	0.98	54.02	10.08
Soul Films	0.98	0.98	0.98	64.02	6.55
Abe Books	1.00	1.00	1.00	4.68	2.97
Awesome Books	1.00	1.00	1.00	3.96	2.89
Better World Books	0.99	0.99	0.99	3.76	5.25
Many Books	0.98	0.97	0.97	13.05	4.28
Waterstones	1.00	0.99	1.00	9.54	4.01
Player Profiles	0.96	0.94	0.94	14.20	12.39
UEFA	1.00	1.00	1.00	1.95	2.81
ATP World Tour	0.92	0.97	0.94	11.14	8.10
NFL	1.00	1.00	1.00	4.15	6.05
Soccer Base	0.87	0.88	0.87	243.88	47.28

Table 2.17: *Experimental results regarding Heuristic H6 .*



Best alternative from Heuristic H5							
				A0			
Summary	P	R	F$_1$	LT	ET	FR	K
Mean	0.97	0.96	0.96	30.02	4.23	-	0.55
Std. Dev.	0.05	0.06	0.05	59.66	5.43		
MDR	20.15	16.85	18.83	15.10	3.30		

Heuristic H6							
				A1			
Summary	P	R	F$_1$	LT	ET	FR	K
Mean	0.97	0.96	0.96	22.17	5.97	-	0.56
Std. Dev.	0.05	0.05	0.05	40.12	7.19		
MDR	19.28	19.67	19.59	12.25	4.96		

Table 2.18: *Ranking of alternatives regarding Heuristic H6 .*

are no clear differences between precision, recall, or the F_1 score between alternatives A0 and A1. However, alternative A1 is 7.85 ± 99.77 minutes faster than alternative A0 when learning rules, which is quite an important difference. Regarding the extraction time, the baseline is 1.74 ± 12.62 minutes faster. As the learning time that results from using alternative A1 is much better, it ranks at the top and we can then select it as the best one. It makes sense that alternative A1 is as effective as the baseline because it just prunes candidates that are not going to be selected to expand the rule according to Heuristic H1 , so that good candidates are still kept; however, it avoids considering a number of candidates that are not promising enough so that it reduces the learning time.

2.3.6 Variation point POSTPROCESSRULESET

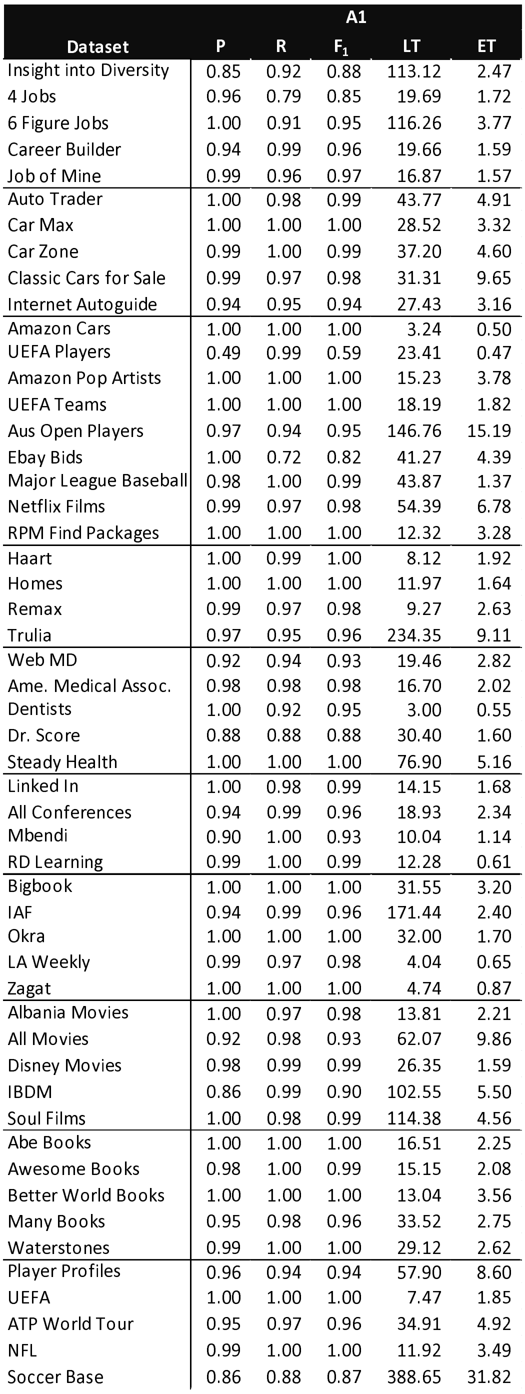
This variation point deals with simplifying a rule set. It consists of Heuristic H7 , which makes a decision regarding which conditions or which rules can be pruned. The goal is to learn rule sets that can be executed more efficiently, but remain as effective as the original ones.

Heuristic H7: post-process rule sets. Simplifying a rule set helps execute the rules more efficiently. We guessed that it would be common to find useless conditions in the rules learnt by TANGO. This happens when determinate conditions are added to a rule; such conditions typically provide little or no gain and they are added to escape local maxima; it is in the next iteration that they are expected to introduce new comparators or further feature instantiator, but there are cases in which they are neglected forever because there are other features that result in conditions that provide more gain.

We also guessed that some rules might subsume other rules, that is, the examples that they match are a subset of the examples that another rule matches; in such cases, the former rule can be discarded. This happens because every rule is learnt independently from the others. Finally, we also thought that folding constants might help make the rules a little more efficient and easier to understand. Typically, TANGO learns many pairs of conditions of the form $f(N, A), A = v$, where f denotes an attributive feature, N a variable that can be bound to a node, A a variable that is bound to the value of feature f on node N, and v is one of the values in the range of that feature; such pairs of conditions may be easily simplified as $f(N, v)$ if variable A is not used anywhere else.

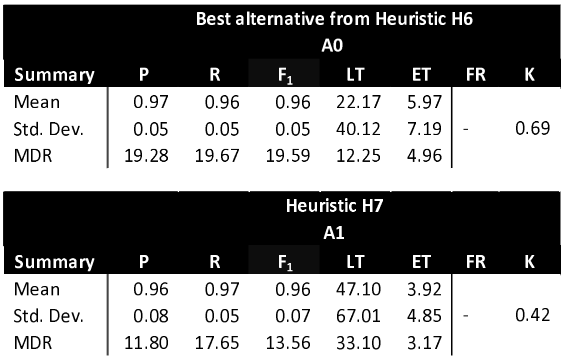
The alternatives that we have devised to implement this heuristic are the following: A0) The rule set is not simplified, that is, the rules are returned as they are learnt. A1) The rules are simplified by removing useless conditions, subsumed rules, and folding constants.

Discussion. The empirical results are presented in Table §2.19 and the ranks are shown in Table §2.20. Again, the differences in effectiveness are not very significant. In some cases, performing post processing led to better results regarding precision, recall, or the F_1 score, but it resulted in worse results in other cases; thus, we conclude that both alternatives behave similarly regarding effectiveness. We studied this issue and we found out that the problem was that removing some conditions or rules from a rule set may not have any impact when the rules are executed on the learning set, but may have an impact when they are executed on a testing set and result in different precisions or recalls. Finding useless conditions or subsumed rules also has an impact on efficiency: we have found that the learning time is 24.93 ± 107.13 minutes slower with alternative A1. Contrarily, the time to execute the rules is, as expected, better, since it is 2.05 ± 12.04 minutes faster than the baseline. However, this difference is not enough when compared to the time spent in the learning process. What makes the learning process slower in alternative A1 is the evaluation process once the rules are learnt. That is, analysing every single condition in the context of a rule to check if removing it can have a negative impact on its effectiveness. Furthermore, analysing every single rule in a rule set to find out subsumed rules also has a negative impact on effectiveness. Thus, our conclusion is that post-processing the resulting rule sets is not really worth.



	A1				
Dataset	P	R	F_1	LT	ET
Insight into Diversity	0.85	0.92	0.88	113.12	2.47
4 Jobs	0.96	0.79	0.85	19.69	1.72
6 Figure Jobs	1.00	0.91	0.95	116.26	3.77
Career Builder	0.94	0.99	0.96	19.66	1.59
Job of Mine	0.99	0.96	0.97	16.87	1.57
Auto Trader	1.00	0.98	0.99	43.77	4.91
Car Max	1.00	1.00	1.00	28.52	3.32
Car Zone	0.99	1.00	0.99	37.20	4.60
Classic Cars for Sale	0.99	0.97	0.98	31.31	9.65
Internet Autoguide	0.94	0.95	0.94	27.43	3.16
Amazon Cars	1.00	1.00	1.00	3.24	0.50
UEFA Players	0.49	0.99	0.59	23.41	0.47
Amazon Pop Artists	1.00	1.00	1.00	15.23	3.78
UEFA Teams	1.00	1.00	1.00	18.19	1.82
Aus Open Players	0.97	0.94	0.95	146.76	15.19
Ebay Bids	1.00	0.72	0.82	41.27	4.39
Major League Baseball	0.98	1.00	0.99	43.87	1.37
Netflix Films	0.99	0.97	0.98	54.39	6.78
RPM Find Packages	1.00	1.00	1.00	12.32	3.28
Haart	1.00	0.99	1.00	8.12	1.92
Homes	1.00	1.00	1.00	11.97	1.64
Remax	0.99	0.97	0.98	9.27	2.63
Trulia	0.97	0.95	0.96	234.35	9.11
Web MD	0.92	0.94	0.93	19.46	2.82
Ame. Medical Assoc.	0.98	0.98	0.98	16.70	2.02
Dentists	1.00	0.92	0.95	3.00	0.55
Dr. Score	0.88	0.88	0.88	30.40	1.60
Steady Health	1.00	1.00	1.00	76.90	5.16
Linked In	1.00	0.98	0.99	14.15	1.68
All Conferences	0.94	0.99	0.96	18.93	2.34
Mbendi	0.90	1.00	0.93	10.04	1.14
RD Learning	0.99	1.00	0.99	12.28	0.61
Bigbook	1.00	1.00	1.00	31.55	3.20
IAF	0.94	0.99	0.96	171.44	2.40
Okra	1.00	1.00	1.00	32.00	1.70
LA Weekly	0.99	0.97	0.98	4.04	0.65
Zagat	1.00	1.00	1.00	4.74	0.87
Albania Movies	1.00	0.97	0.98	13.81	2.21
All Movies	0.92	0.98	0.93	62.07	9.86
Disney Movies	0.98	0.99	0.99	26.35	1.59
IBDM	0.86	0.99	0.90	102.55	5.50
Soul Films	1.00	0.98	0.99	114.38	4.56
Abe Books	1.00	1.00	1.00	16.51	2.25
Awesome Books	0.98	1.00	0.99	15.15	2.08
Better World Books	1.00	1.00	1.00	13.04	3.56
Many Books	0.95	0.98	0.96	33.52	2.75
Waterstones	0.99	1.00	1.00	29.12	2.62
Player Profiles	0.96	0.94	0.94	57.90	8.60
UEFA	1.00	1.00	1.00	7.47	1.85
ATP World Tour	0.95	0.97	0.96	34.91	4.92
NFL	0.99	1.00	1.00	11.92	3.49
Soccer Base	0.86	0.88	0.87	388.65	31.82

Table 2.19: *Experimental results regarding Heuristic H7 .*



			Best alternative from Heuristic H6				
			A0				
Summary	P	R	F_1	LT	ET	FR	K
Mean	0.97	0.96	0.96	22.17	5.97	-	0.69
Std. Dev.	0.05	0.05	0.05	40.12	7.19		
MDR	19.28	19.67	19.59	12.25	4.96		

			Heuristic H7				
			A1				
Summary	P	R	F_1	LT	ET	FR	K
Mean	0.96	0.97	0.96	47.10	3.92	-	0.42
Std. Dev.	0.08	0.05	0.07	67.01	4.85		
MDR	11.80	17.65	13.56	33.10	3.17		

Table 2.20: *Ranking of alternatives regarding Heuristic H7 .*

2.3.7 Variation point BRANCH

This variation point deals with computing the set of conditions that can possibly be added to the current rule. It consists of several heuristics, namely: Heuristic H8 regarding whether recursive rules are allowed or not; Heuristic H9 regarding whether the conditions must be generated in a given order or not; and Heuristic H10 regarding whether the input/output modes of the features must be considered or not. The goal is to customise TANGO so that it can learn extraction rules more efficiently without degrading their effectiveness or increasing the time required to execute them.

Heuristic H8: allow recursion. A rule is recursive if its body contains a condition that is a slot instantiator. We guessed that allowing for recursive rules might help learn more general rules that are simpler in some cases.

We then considered the following alternatives: A0) No recursion is allowed. A1) Rules are allowed to be recursive.

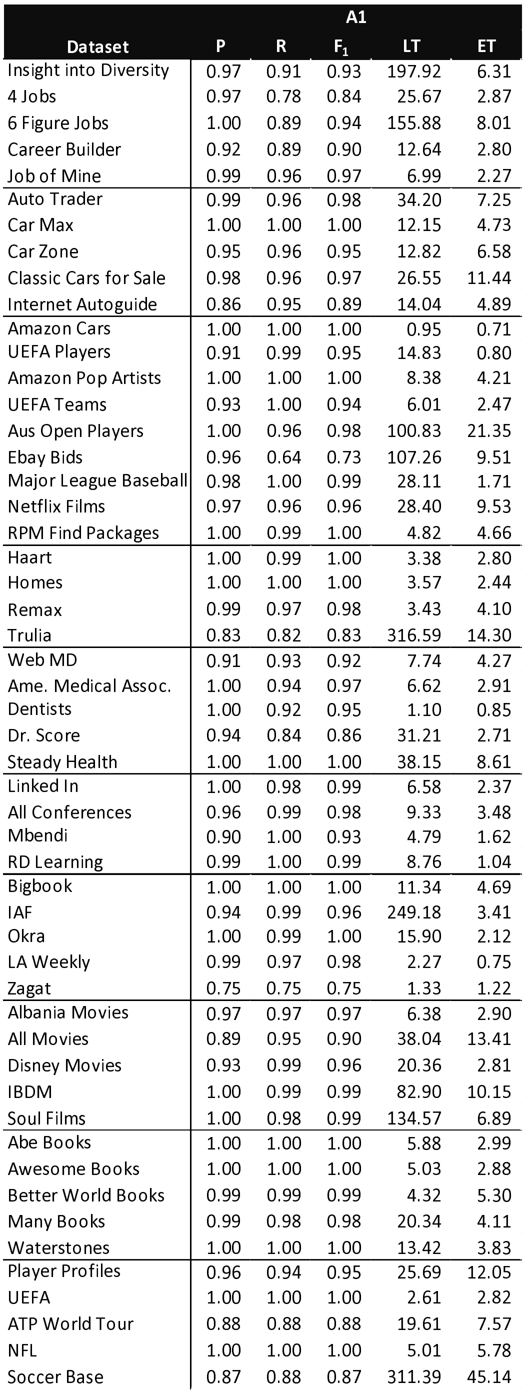
Alternative A0 is very simple, as usual. Alternative A1 is a little more involved because we have to make sure that making a rule recursive does not result in infinite recursion. Such a recursion occurs when a recursive condition includes a variable that is instantiated with the same example that is going to be extracted. The simplest solution to solve this problem is to determine if there exists a complete order amongst the variable used in the recursive condition and the variable in the head of the rule, both of which must be bound to nodes. If such an order exists, then it means that the variable in the recursive condition and the variable in the head cannot be instantiated with the same examples, which guarantees that the recursion is safe; if no such order exists, then the recursion is unsafe and must be avoided. Checking if there exists a complete order amongst some variables can be easily implemented by using Ajwani and others's algorithm [3], for instance.

Discussion. The empirical results are presented in Table §2.21 and the ranks are shown in Table §2.22. The differences in effectiveness are not very significant. What makes a big difference is the learning time, which is 20.43 ± 115.14 minutes slower in alternative A1. This had a very negative impact on the computation of the rank and made us select the baseline as the best alternative in this heuristic. The reason why alternative A1 took longer during the learning phase is because it has to compute if there is an order between any two variables that can be instantiated with nodes, which took very long and the rules did not improve because there was not a single case in which the slot instantiator was included in the body of the rule, so there was not any improvement regarding making rules more simpler and/or general.

Heuristic H9: sort conditions. The order in which the conditions that result from branching a rule are examined matters because if very good conditions are explored first, then it is more likely that the learning process finds the solution faster; furthermore, the pruning process becomes more demanding. This contributes to saving time because the number of conditions that are explored is smaller. The problem is how to find such an order without actually computing the gain of a condition.

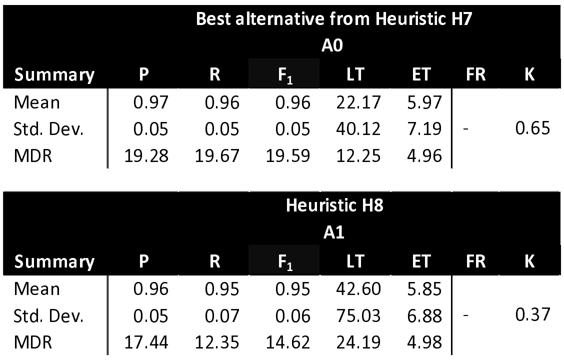
The alternatives that we have devised to implement this heuristic are the following: A0) Conditions are generated in a random order. A1) Comparators are generated first, then slot instantiators (if recursion is allowed), and then feature instantiators in random order. A2) Comparators are generated first, then slot instantiators (if recursion is allowed), and then the feature instantiators are generated using an empirical frequency-based order.

Alternative A0 is the simplest one, as usual. Alternative A1 makes it explicit that we guessed that comparators would typically result in higher gains than feature instantiators. Alternative A2 relies on an order that we have computed empirically, cf. Table §2.23. Note that we have performed hundreds of experiments and that we have used TANGO to learn thousands of rules. What we have done is to compute the frequency with which each feature in our catalogue was used in a rule; we guessed that generating the



	A1				
Dataset	P	R	F_1	LT	ET
Insight into Diversity	0.97	0.91	0.93	197.92	6.31
4 Jobs	0.97	0.78	0.84	25.67	2.87
6 Figure Jobs	1.00	0.89	0.94	155.88	8.01
Career Builder	0.92	0.89	0.90	12.64	2.80
Job of Mine	0.99	0.96	0.97	6.99	2.27
Auto Trader	0.99	0.96	0.98	34.20	7.25
Car Max	1.00	1.00	1.00	12.15	4.73
Car Zone	0.95	0.96	0.95	12.82	6.58
Classic Cars for Sale	0.98	0.96	0.97	26.55	11.44
Internet Autoguide	0.86	0.95	0.89	14.04	4.89
Amazon Cars	1.00	1.00	1.00	0.95	0.71
UEFA Players	0.91	0.99	0.95	14.83	0.80
Amazon Pop Artists	1.00	1.00	1.00	8.38	4.21
UEFA Teams	0.93	1.00	0.94	6.01	2.47
Aus Open Players	1.00	0.96	0.98	100.83	21.35
Ebay Bids	0.96	0.64	0.73	107.26	9.51
Major League Baseball	0.98	1.00	0.99	28.11	1.71
Netflix Films	0.97	0.96	0.96	28.40	9.53
RPM Find Packages	1.00	0.99	1.00	4.82	4.66
Haart	1.00	0.99	1.00	3.38	2.80
Homes	1.00	1.00	1.00	3.57	2.44
Remax	0.99	0.97	0.98	3.43	4.10
Trulia	0.83	0.82	0.83	316.59	14.30
Web MD	0.91	0.93	0.92	7.74	4.27
Ame. Medical Assoc.	1.00	0.94	0.97	6.62	2.91
Dentists	1.00	0.92	0.95	1.10	0.85
Dr. Score	0.94	0.84	0.86	31.21	2.71
Steady Health	1.00	1.00	1.00	38.15	8.61
Linked In	1.00	0.98	0.99	6.58	2.37
All Conferences	0.96	0.99	0.98	9.33	3.48
Mbendi	0.90	1.00	0.93	4.79	1.62
RD Learning	0.99	1.00	0.99	8.76	1.04
Bigbook	1.00	1.00	1.00	11.34	4.69
IAF	0.94	0.99	0.96	249.18	3.41
Okra	1.00	0.99	1.00	15.90	2.12
LA Weekly	0.99	0.97	0.98	2.27	0.75
Zagat	0.75	0.75	0.75	1.33	1.22
Albania Movies	0.97	0.97	0.97	6.38	2.90
All Movies	0.89	0.95	0.90	38.04	13.41
Disney Movies	0.93	0.99	0.96	20.36	2.81
IBDM	1.00	0.99	0.99	82.90	10.15
Soul Films	1.00	0.98	0.99	134.57	6.89
Abe Books	1.00	1.00	1.00	5.88	2.99
Awesome Books	1.00	1.00	1.00	5.03	2.88
Better World Books	0.99	0.99	0.99	4.32	5.30
Many Books	0.99	0.98	0.98	20.34	4.11
Waterstones	1.00	1.00	1.00	13.42	3.83
Player Profiles	0.96	0.94	0.95	25.69	12.05
UEFA	1.00	1.00	1.00	2.61	2.82
ATP World Tour	0.88	0.88	0.88	19.61	7.57
NFL	1.00	1.00	1.00	5.01	5.78
Soccer Base	0.87	0.88	0.87	311.39	45.14

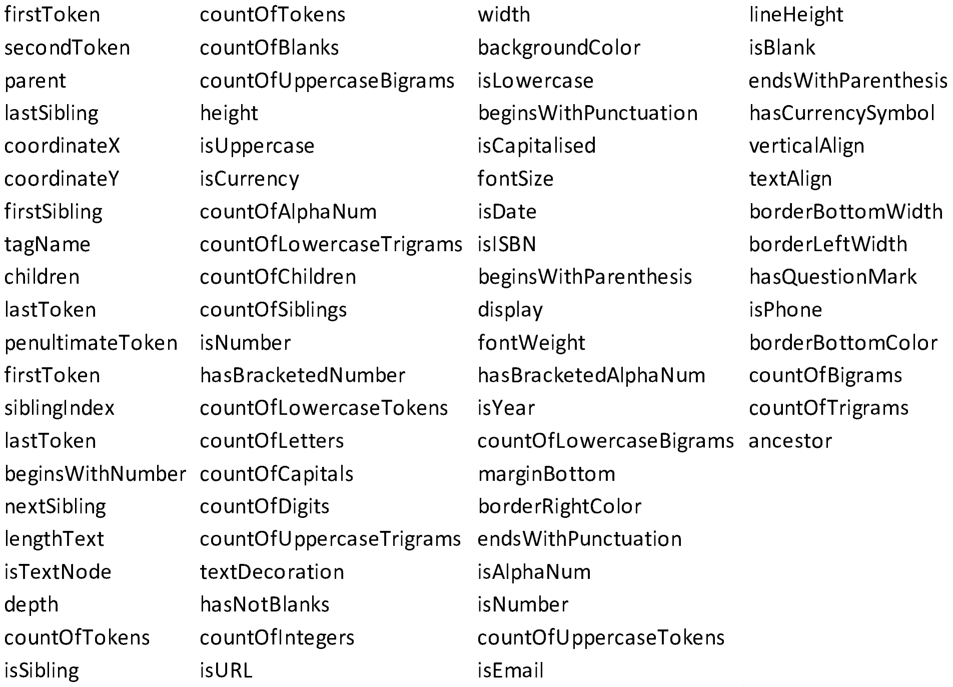
Table 2.21: *Experimental results regarding Heuristic H8 .*



Best alternative from Heuristic H7							
A0							
Summary	P	R	F_1	LT	ET	FR	K
Mean	0.97	0.96	0.96	22.17	5.97	-	0.65
Std. Dev.	0.05	0.05	0.05	40.12	7.19		
MDR	19.28	19.67	19.59	12.25	4.96		

Heuristic H8							
A1							
Summary	P	R	F_1	LT	ET	FR	K
Mean	0.96	0.95	0.95	42.60	5.85	-	0.37
Std. Dev.	0.05	0.07	0.06	75.03	6.88		
MDR	17.44	12.35	14.62	24.19	4.98		

Table 2.22: *Ranking of alternatives regarding Heuristic H8 .*



firstToken
secondToken
parent
lastSibling
coordinateX
coordinateY
firstSibling
tagName
children
lastToken
penultimateToken
firstToken
siblingIndex
lastToken
beginsWithNumber
nextSibling
lengthText
isTextNode
depth
countOfTokens
isSibling

countOfTokens
countOfBlanks
countOfUppercaseBigrams
height
isUppercase
isCurrency
countOfAlphaNum
countOfLowercaseTrigrams
countOfChildren
countOfSiblings
isNumber
hasBracketedNumber
countOfLowercaseTokens
countOfLetters
countOfCapitals
countOfDigits
countOfUppercaseTrigrams
textDecoration
hasNotBlanks
countOfIntegers
isURL

width
backgroundColor
isLowercase
beginsWithPunctuation
isCapitalised
fontSize
isDate
isISBN
beginsWithParenthesis
display
fontWeight
hasBracketedAlphaNum
isYear
countOfLowercaseBigrams
marginBottom
borderRightColor
endsWithPunctuation
isAlphaNum
isNumber
countOfUppercaseTokens
isEmail

lineHeight
isBlank
endsWithParenthesis
hasCurrencySymbol
verticalAlign
textAlign
borderBottomWidth
borderLeftWidth
hasQuestionMark
isPhone
borderBottomColor
countOfBigrams
countOfTrigrams
ancestor

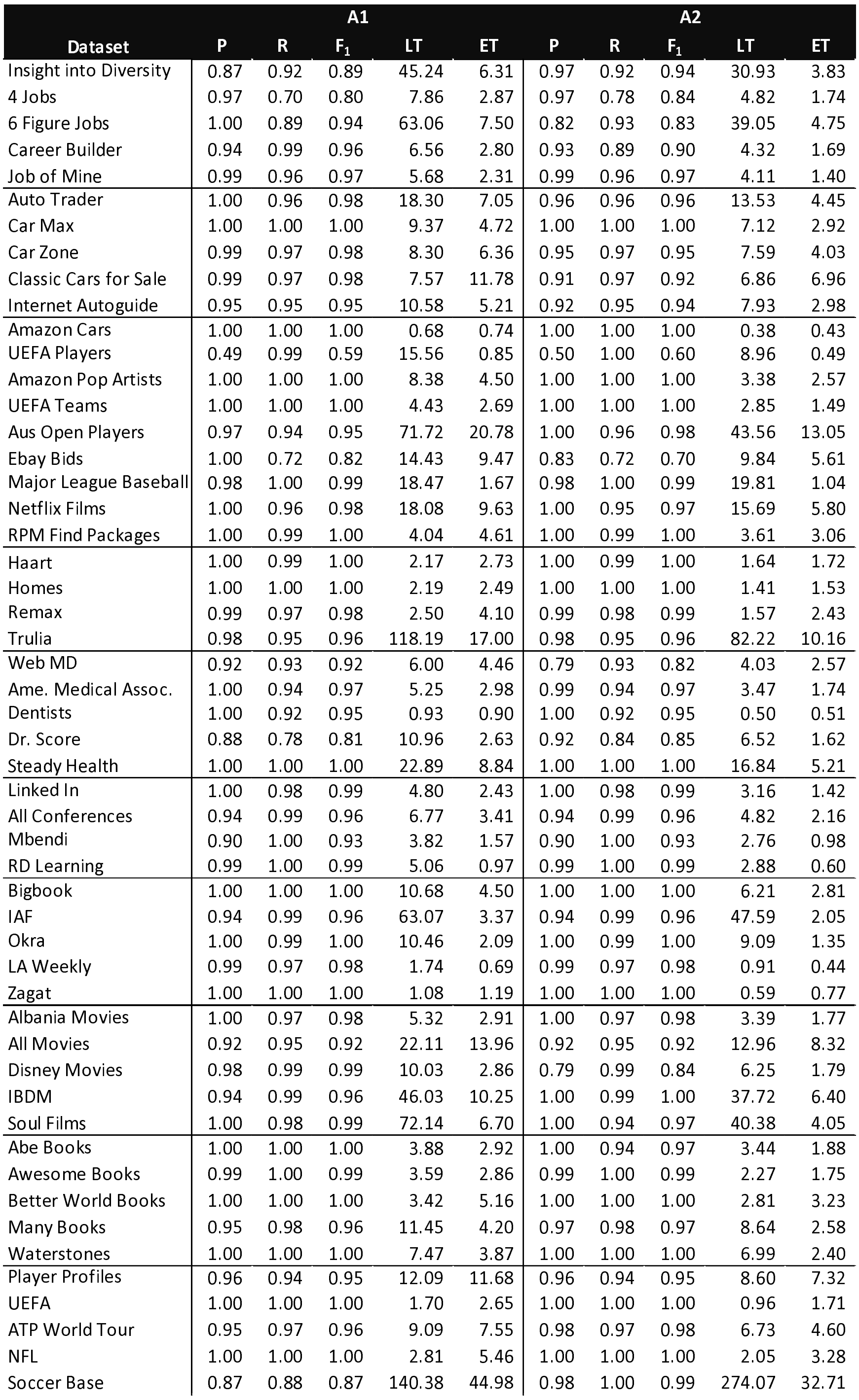
Table 2.23: *Partial catalogue of features. (Sorted by empirical frequency.)*

feature instantiators using that empirical order might help learn rules more efficiently.

Discussion. The empirical results are presented in Table §2.24 and the ranks are shown in Table §2.25. Note that alternative A2 is the one that performs the best according to our rank, which was not surprising. The baseline seems to perform a little better than alternative A2 regarding effectiveness, but the differences are negligible. Alternative A0 results in a precision that is 0.01 ± 0.13 higher, a recall that is 0.00 ± 0.10 higher, and an F_1 score that is 0.01 ± 0.12 higher than the corresponding ones in alternative A2. However, both alternatives A1 and A2 are faster than the baseline when learning rules, namely: alternative A1 is 3.54 ± 68.91 minutes faster and alternative A2 is 5.86 ± 79.74 minutes faster. However, regarding the extraction time, alternative A2 beats both A0 and A1 since it is 2.28 ± 12.02 minutes faster. The improvement in both learning and extraction times has made us select alternative A2 as the best one. It makes sense that sorting the features according to their empirical frequencies results in better timings since the features that have proven to work better at making a difference amongst the positive and the negative examples are prioritised and this helps find the best conditions faster.

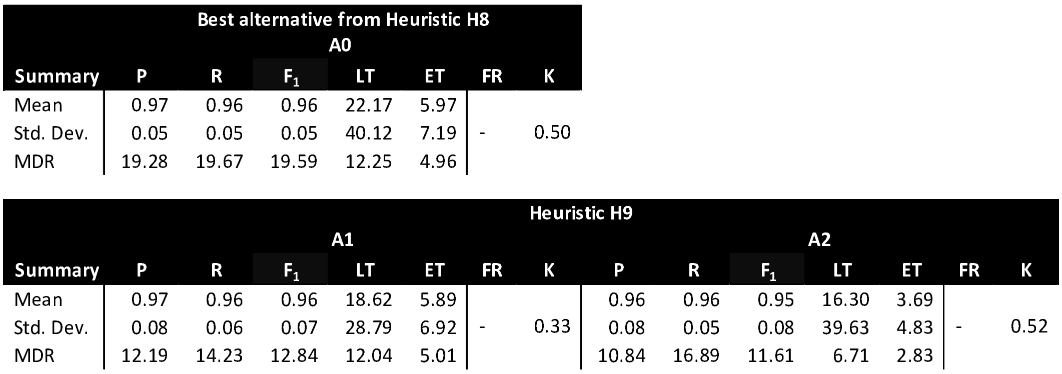
Heuristic H10: consider input/output modes. Feature instantiators are of the form $f(N, X)$; typically, when such a condition is added to a rule, variable N is expected to be bound, that is, it is expected to have been used in a previous condition or in the header of the rule; contrarily, variable X is expected to be unbound, that is, a fresh variable that has not been used before. The rationale behind this idea is that the current rule binds some nodes to variable N and for each such node the value/s of feature f is/are bound to variable X. Obviously, since we are working with first-order conditions, we might also consider a feature instantiator in which variable N is unbound and variable X is bound, which would allow to find all of the nodes for which feature f has a given value. When a parameter is expected to be a bound variable, it is said that its mode is input; when it is expected to be an unbound variable, it is said that its mode is output; when it is expected to be a bound or an unbound variable, it is said that its mode is input/output. By default all of the parameters in the features of our catalogue have input/output modes with the only restriction that when a feature instantiator is created, one of its parameters must be a bound variable.

The alternatives that we have devised to implement this heuristic are the following: A0) Do not take input/output modes into account regarding relational features. A1) Take them into account.



	A1					A2				
Dataset	P	R	F_1	LT	ET	P	R	F_1	LT	ET
Insight into Diversity	0.87	0.92	0.89	45.24	6.31	0.97	0.92	0.94	30.93	3.83
4 Jobs	0.97	0.70	0.80	7.86	2.87	0.97	0.78	0.84	4.82	1.74
6 Figure Jobs	1.00	0.89	0.94	63.06	7.50	0.82	0.93	0.83	39.05	4.75
Career Builder	0.94	0.99	0.96	6.56	2.80	0.93	0.89	0.90	4.32	1.69
Job of Mine	0.99	0.96	0.97	5.68	2.31	0.99	0.96	0.97	4.11	1.40
Auto Trader	1.00	0.96	0.98	18.30	7.05	0.96	0.96	0.96	13.53	4.45
Car Max	1.00	1.00	1.00	9.37	4.72	1.00	1.00	1.00	7.12	2.92
Car Zone	0.99	0.97	0.98	8.30	6.36	0.95	0.97	0.95	7.59	4.03
Classic Cars for Sale	0.99	0.97	0.98	7.57	11.78	0.91	0.97	0.92	6.86	6.96
Internet Autoguide	0.95	0.95	0.95	10.58	5.21	0.92	0.95	0.94	7.93	2.98
Amazon Cars	1.00	1.00	1.00	0.68	0.74	1.00	1.00	1.00	0.38	0.43
UEFA Players	0.49	0.99	0.59	15.56	0.85	0.50	1.00	0.60	8.96	0.49
Amazon Pop Artists	1.00	1.00	1.00	8.38	4.50	1.00	1.00	1.00	3.38	2.57
UEFA Teams	1.00	1.00	1.00	4.43	2.69	1.00	1.00	1.00	2.85	1.49
Aus Open Players	0.97	0.94	0.95	71.72	20.78	1.00	0.96	0.98	43.56	13.05
Ebay Bids	1.00	0.72	0.82	14.43	9.47	0.83	0.72	0.70	9.84	5.61
Major League Baseball	0.98	1.00	0.99	18.47	1.67	0.98	1.00	0.99	19.81	1.04
Netflix Films	1.00	0.96	0.98	18.08	9.63	1.00	0.95	0.97	15.69	5.80
RPM Find Packages	1.00	0.99	1.00	4.04	4.61	1.00	0.99	1.00	3.61	3.06
Haart	1.00	0.99	1.00	2.17	2.73	1.00	0.99	1.00	1.64	1.72
Homes	1.00	1.00	1.00	2.19	2.49	1.00	1.00	1.00	1.41	1.53
Remax	0.99	0.97	0.98	2.50	4.10	0.99	0.98	0.99	1.57	2.43
Trulia	0.98	0.95	0.96	118.19	17.00	0.98	0.95	0.96	82.22	10.16
Web MD	0.92	0.93	0.92	6.00	4.46	0.79	0.93	0.82	4.03	2.57
Ame. Medical Assoc.	1.00	0.94	0.97	5.25	2.98	0.99	0.94	0.97	3.47	1.74
Dentists	1.00	0.92	0.95	0.93	0.90	1.00	0.92	0.95	0.50	0.51
Dr. Score	0.88	0.78	0.81	10.96	2.63	0.92	0.84	0.85	6.52	1.62
Steady Health	1.00	1.00	1.00	22.89	8.84	1.00	1.00	1.00	16.84	5.21
Linked In	1.00	0.98	0.99	4.80	2.43	1.00	0.98	0.99	3.16	1.42
All Conferences	0.94	0.99	0.96	6.77	3.41	0.94	0.99	0.96	4.82	2.16
Mbendi	0.90	1.00	0.93	3.82	1.57	0.90	1.00	0.93	2.76	0.98
RD Learning	0.99	1.00	0.99	5.06	0.97	0.99	1.00	0.99	2.88	0.60
Bigbook	1.00	1.00	1.00	10.68	4.50	1.00	1.00	1.00	6.21	2.81
IAF	0.94	0.99	0.96	63.07	3.37	0.94	0.99	0.96	47.59	2.05
Okra	1.00	0.99	1.00	10.46	2.09	1.00	0.99	1.00	9.09	1.35
LA Weekly	0.99	0.97	0.98	1.74	0.69	0.99	0.97	0.98	0.91	0.44
Zagat	1.00	1.00	1.00	1.08	1.19	1.00	1.00	1.00	0.59	0.77
Albania Movies	1.00	0.97	0.98	5.32	2.91	1.00	0.97	0.98	3.39	1.77
All Movies	0.92	0.95	0.92	22.11	13.96	0.92	0.95	0.92	12.96	8.32
Disney Movies	0.98	0.99	0.99	10.03	2.86	0.79	0.99	0.84	6.25	1.79
IBDM	0.94	0.99	0.96	46.03	10.25	1.00	0.99	1.00	37.72	6.40
Soul Films	1.00	0.98	0.99	72.14	6.70	1.00	0.94	0.97	40.38	4.05
Abe Books	1.00	1.00	1.00	3.88	2.92	1.00	0.94	0.97	3.44	1.88
Awesome Books	0.99	1.00	0.99	3.59	2.86	0.99	1.00	0.99	2.27	1.75
Better World Books	1.00	1.00	1.00	3.42	5.16	1.00	1.00	1.00	2.81	3.23
Many Books	0.95	0.98	0.96	11.45	4.20	0.97	0.98	0.97	8.64	2.58
Waterstones	1.00	1.00	1.00	7.47	3.87	1.00	1.00	1.00	6.99	2.40
Player Profiles	0.96	0.94	0.95	12.09	11.68	0.96	0.94	0.95	8.60	7.32
UEFA	1.00	1.00	1.00	1.70	2.65	1.00	1.00	1.00	0.96	1.71
ATP World Tour	0.95	0.97	0.96	9.09	7.55	0.98	0.97	0.98	6.73	4.60
NFL	1.00	1.00	1.00	2.81	5.46	1.00	1.00	1.00	2.05	3.28
Soccer Base	0.87	0.88	0.87	140.38	44.98	0.98	1.00	0.99	274.07	32.71

Table 2.24: *Experimental results regarding Heuristic H9 .*



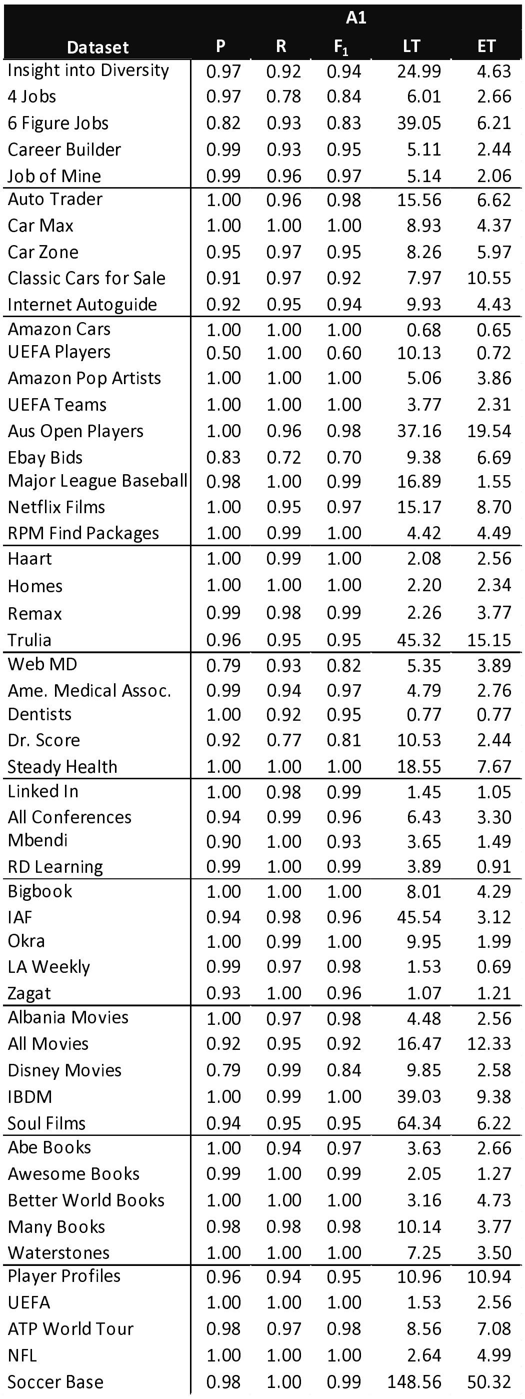
	Best alternative from Heuristic H8						
	A0						
Summary	P	R	F_1	LT	ET	FR	K
Mean	0.97	0.96	0.96	22.17	5.97		
Std. Dev.	0.05	0.05	0.05	40.12	7.19	-	0.50
MDR	19.28	19.67	19.59	12.25	4.96		

	Heuristic H9													
	A1							A2						
Summary	P	R	F_1	LT	ET	FR	K	P	R	F_1	LT	ET	FR	K
Mean	0.97	0.96	0.96	18.62	5.89			0.96	0.96	0.95	16.30	3.69		
Std. Dev.	0.08	0.06	0.07	28.79	6.92	-	0.33	0.08	0.05	0.08	39.63	4.83	-	0.52
MDR	12.19	14.23	12.84	12.04	5.01			10.84	16.89	11.61	6.71	2.83		

Table 2.25: *Ranking of alternatives regarding Heuristic H9 .*

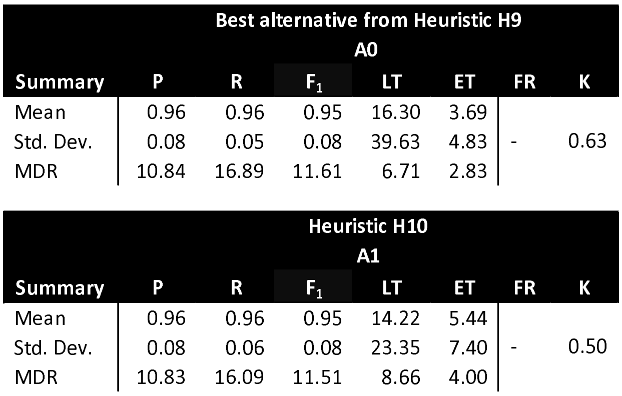
Alternative A0 is the simplest one and it helps explore as many feature instantiators as possible. Alternative A1 restricts the relational feature instantiators so that the first parameter is a bound variable and the second one is an unbound variable. Note that such a restriction may have a subtle implication regarding the catalogue of features and some relational features. For instance, recall that we included relational features $\mathtt{left}$ and $\mathtt{parent}$ in our running example. If input/output modes are not taken into account, then a feature instantiator like $\mathtt{left}(N, M)$ helps navigate from a node to its left sibling if N is bound and M is unbound, or to navigate to its right sibling if N is unbound and M is bound; similarly, $\mathtt{parent}(N, M)$ helps navigate from a node to its parent or from a node to its children. Note that this is not possible if input/out modes are taken into account; in such cases, features $\mathtt{right}$ and $\mathtt{child}$ must be provided explicitly in the catalogue of features or, otherwise, TANGO shall not be able to find the right sibling of a node or its children.

Discussion. The empirical results are presented in Table §2.26 and the ranks are shown Table §2.27. Again, the differences in effectiveness are not very significant. It learnt almost the same rules for the datasets. Regarding the efficiency, it was expected that alternative A1 reduced the learning time since there are some conditions that are not generated during the branching procedure; our experimental results confirm this idea because alternative A1 is 2.08 ± 62.98 minutes faster than alternative A0. Unfortunately, the extraction time worsened because it was 1.74 ± 2.57 minutes slower in alternative A1. Therefore, the improvement on the learning time of alternative A1 regarding alternative A0 is not enough to select it. So we keep alternative A0 as the best one.



	A1				
Dataset	P	R	F_1	LT	ET
Insight into Diversity	0.97	0.92	0.94	24.99	4.63
4 Jobs	0.97	0.78	0.84	6.01	2.66
6 Figure Jobs	0.82	0.93	0.83	39.05	6.21
Career Builder	0.99	0.93	0.95	5.11	2.44
Job of Mine	0.99	0.96	0.97	5.14	2.06
Auto Trader	1.00	0.96	0.98	15.56	6.62
Car Max	1.00	1.00	1.00	8.93	4.37
Car Zone	0.95	0.97	0.95	8.26	5.97
Classic Cars for Sale	0.91	0.97	0.92	7.97	10.55
Internet Autoguide	0.92	0.95	0.94	9.93	4.43
Amazon Cars	1.00	1.00	1.00	0.68	0.65
UEFA Players	0.50	1.00	0.60	10.13	0.72
Amazon Pop Artists	1.00	1.00	1.00	5.06	3.86
UEFA Teams	1.00	1.00	1.00	3.77	2.31
Aus Open Players	1.00	0.96	0.98	37.16	19.54
Ebay Bids	0.83	0.72	0.70	9.38	6.69
Major League Baseball	0.98	1.00	0.99	16.89	1.55
Netflix Films	1.00	0.95	0.97	15.17	8.70
RPM Find Packages	1.00	0.99	1.00	4.42	4.49
Haart	1.00	0.99	1.00	2.08	2.56
Homes	1.00	1.00	1.00	2.20	2.34
Remax	0.99	0.98	0.99	2.26	3.77
Trulia	0.96	0.95	0.95	45.32	15.15
Web MD	0.79	0.93	0.82	5.35	3.89
Ame. Medical Assoc.	0.99	0.94	0.97	4.79	2.76
Dentists	1.00	0.92	0.95	0.77	0.77
Dr. Score	0.92	0.77	0.81	10.53	2.44
Steady Health	1.00	1.00	1.00	18.55	7.67
Linked In	1.00	0.98	0.99	1.45	1.05
All Conferences	0.94	0.99	0.96	6.43	3.30
Mbendi	0.90	1.00	0.93	3.65	1.49
RD Learning	0.99	1.00	0.99	3.89	0.91
Bigbook	1.00	1.00	1.00	8.01	4.29
IAF	0.94	0.98	0.96	45.54	3.12
Okra	1.00	0.99	1.00	9.95	1.99
LA Weekly	0.99	0.97	0.98	1.53	0.69
Zagat	0.93	1.00	0.96	1.07	1.21
Albania Movies	1.00	0.97	0.98	4.48	2.56
All Movies	0.92	0.95	0.92	16.47	12.33
Disney Movies	0.79	0.99	0.84	9.85	2.58
IBDM	1.00	0.99	1.00	39.03	9.38
Soul Films	0.94	0.95	0.95	64.34	6.22
Abe Books	1.00	0.94	0.97	3.63	2.66
Awesome Books	0.99	1.00	0.99	2.05	1.27
Better World Books	1.00	1.00	1.00	3.16	4.73
Many Books	0.98	0.98	0.98	10.14	3.77
Waterstones	1.00	1.00	1.00	7.25	3.50
Player Profiles	0.96	0.94	0.95	10.96	10.94
UEFA	1.00	1.00	1.00	1.53	2.56
ATP World Tour	0.98	0.97	0.98	8.56	7.08
NFL	1.00	1.00	1.00	2.64	4.99
Soccer Base	0.98	1.00	0.99	148.56	50.32

Table 2.26: *Experimental results regarding Heuristic H10 .*



			Best alternative from Heuristic H9				
			A0				
Summary	P	R	F_1	LT	ET	FR	K
Mean	0.96	0.96	0.95	16.30	3.69	-	0.63
Std. Dev.	0.08	0.05	0.08	39.63	4.83		
MDR	10.84	16.89	11.61	6.71	2.83		

			Heuristic H10				
			A1				
Summary	P	R	F_1	LT	ET	FR	K
Mean	0.96	0.96	0.95	14.22	5.44	-	0.50
Std. Dev.	0.08	0.06	0.08	23.35	7.40		
MDR	10.83	16.09	11.51	8.66	4.00		

Table 2.27: *Ranking of alternatives regarding Heuristic H10 .*

2.3.8 Variation point isTooComplex

This variation point deals with producing general rules, that is, to avoid producing very specific rules. It consists of Heuristic H11 , which makes a decision regarding wether the rule being learnt is becoming too specific. The goal is to learn extraction rules that can be executed more efficiently.

Heuristic H11: check complexity of rules. A rule is complex when the number of examples that it matches is relatively small with respect to the number of conditions that it has. Avoiding producing complex rules has a positive impact on the extraction time; it may also improve recall because complex rules tend to be overly-specific. We guessed that pruning a rule when it becomes too complex and performing backtracking would result in better rules.

The alternatives that we have devised to implement this heuristic are the following: A0) Do not use any complexity criteria. A1) Use the Minimum Description Length principle [146].

The idea behind the Minimum Description Length principle is that a rule that requires more bits to be encoded than to encode the examples that it matches is too complex to be explored. The number of bits to encode a rule is computed as the sum of the bits required to encode every condition in its body, which is computed as the number of bits required to encode the features in the catalogue, plus the number of combinations that would result from combining their parameters, the built-in comparators, and the target slot instantiator (if recursion is allowed).

Discussion. The empirical results are presented in Table §2.28 and the ranks are shown Table §2.29. There are no differences in effectiveness since the rules learnt are exactly the same. It seems that the rules never become very complex since, in most of the cases, just one rule was enough to match the whole set of positive examples in the learning set. Consequently, there are not any differences regarding extraction time. However, computing the bits to encode each condition that is added to a rule and encoding the examples that it matches makes the learning process a bit more inefficient. This is why the learning time in alternative A1 is 6.85 ± 95.31 minutes slower. As a conclusion, we prefer to keep alternative A0 as the best one.

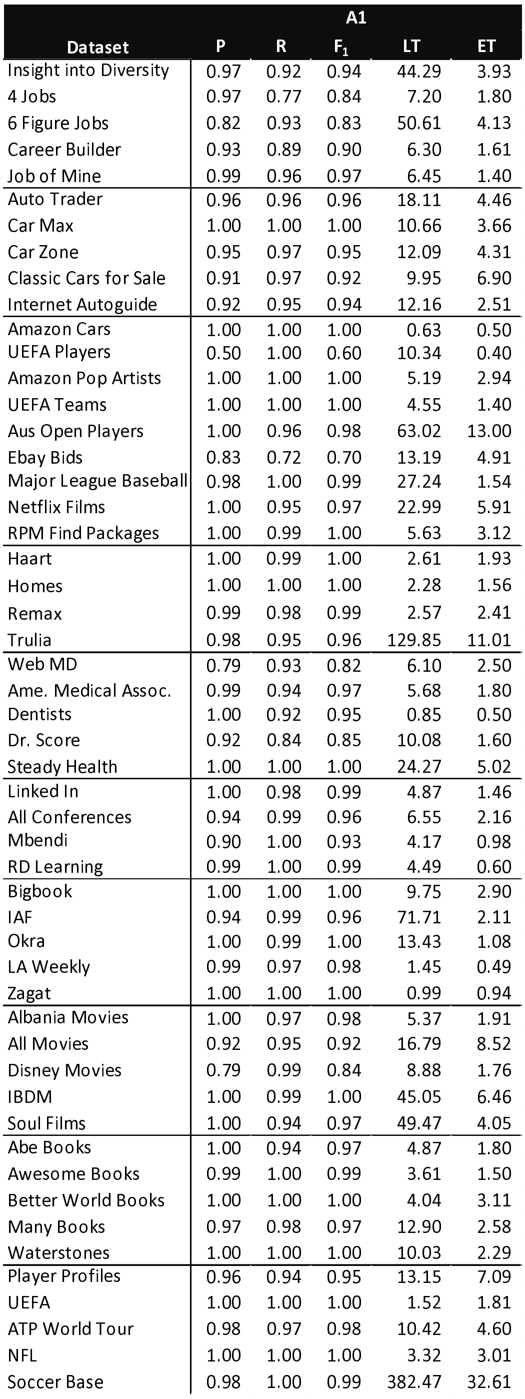
2.4 Experimental analysis

In this section, we first report on the results of our experimental analysis regarding effectiveness and then regarding efficiency. Please, consult the Appendices §A and §B for further details regarding our experimental environment, which includes a description of the hardware and the software used, the evaluation datasets, the catalogue of features, the proposals with which we have compared ours, the performance measures that we collected, and the statistical tests that we used.

2.4.1 Effectiveness analysis

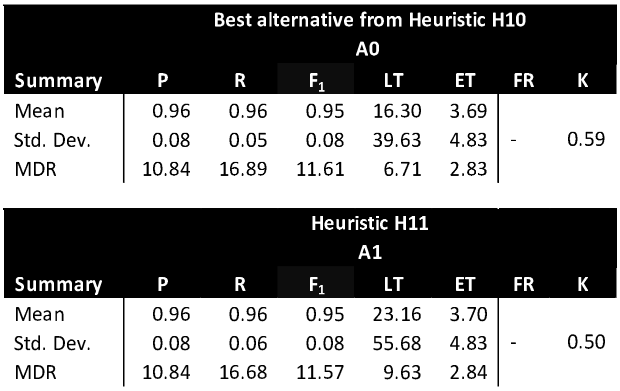
Table §2.30 reports on the raw effectiveness data that we got from our experimentation. For each proposal, we report on its effectiveness measures regarding our datasets. The first two lines also provide a summary of the results in terms of mean value and the standard deviation of each measure. Since it is difficult to spot a trend in this table, we decided to summarise the data using boxplots.

Table §2.31 summarises the results regarding precision. Empirically, TANGO seems to be the proposal that can achieve the best precision, and it is, indeed, the one that is more stable regarding this effectiveness measure since its standard deviation is the smallest, and its inter-quartile range is also the smallest; the other proposals can also achieve good results regarding precision, but their deviation with respect to the mean is larger. Note, however, that some other techniques can achieve results that are very good, too, chiefly Aleph. Iman-Davenport's test returns a p-value that is nearly zero, which is a strong indication that there are differences in rank amongst the proposals that we have compared. We then have to compare TANGO, which ranks



	A1				
Dataset	P	R	F_1	LT	ET
Insight into Diversity	0.97	0.92	0.94	44.29	3.93
4 Jobs	0.97	0.77	0.84	7.20	1.80
6 Figure Jobs	0.82	0.93	0.83	50.61	4.13
Career Builder	0.93	0.89	0.90	6.30	1.61
Job of Mine	0.99	0.96	0.97	6.45	1.40
Auto Trader	0.96	0.96	0.96	18.11	4.46
Car Max	1.00	1.00	1.00	10.66	3.66
Car Zone	0.95	0.97	0.95	12.09	4.31
Classic Cars for Sale	0.91	0.97	0.92	9.95	6.90
Internet Autoguide	0.92	0.95	0.94	12.16	2.51
Amazon Cars	1.00	1.00	1.00	0.63	0.50
UEFA Players	0.50	1.00	0.60	10.34	0.40
Amazon Pop Artists	1.00	1.00	1.00	5.19	2.94
UEFA Teams	1.00	1.00	1.00	4.55	1.40
Aus Open Players	1.00	0.96	0.98	63.02	13.00
Ebay Bids	0.83	0.72	0.70	13.19	4.91
Major League Baseball	0.98	1.00	0.99	27.24	1.54
Netflix Films	1.00	0.95	0.97	22.99	5.91
RPM Find Packages	1.00	0.99	1.00	5.63	3.12
Haart	1.00	0.99	1.00	2.61	1.93
Homes	1.00	1.00	1.00	2.28	1.56
Remax	0.99	0.98	0.99	2.57	2.41
Trulia	0.98	0.95	0.96	129.85	11.01
Web MD	0.79	0.93	0.82	6.10	2.50
Ame. Medical Assoc.	0.99	0.94	0.97	5.68	1.80
Dentists	1.00	0.92	0.95	0.85	0.50
Dr. Score	0.92	0.84	0.85	10.08	1.60
Steady Health	1.00	1.00	1.00	24.27	5.02
Linked In	1.00	0.98	0.99	4.87	1.46
All Conferences	0.94	0.99	0.96	6.55	2.16
Mbendi	0.90	1.00	0.93	4.17	0.98
RD Learning	0.99	1.00	0.99	4.49	0.60
Bigbook	1.00	1.00	1.00	9.75	2.90
IAF	0.94	0.99	0.96	71.71	2.11
Okra	1.00	0.99	1.00	13.43	1.08
LA Weekly	0.99	0.97	0.98	1.45	0.49
Zagat	1.00	1.00	1.00	0.99	0.94
Albania Movies	1.00	0.97	0.98	5.37	1.91
All Movies	0.92	0.95	0.92	16.79	8.52
Disney Movies	0.79	0.99	0.84	8.88	1.76
IBDM	1.00	0.99	1.00	45.05	6.46
Soul Films	1.00	0.94	0.97	49.47	4.05
Abe Books	1.00	0.94	0.97	4.87	1.80
Awesome Books	0.99	1.00	0.99	3.61	1.50
Better World Books	1.00	1.00	1.00	4.04	3.11
Many Books	0.97	0.98	0.97	12.90	2.58
Waterstones	1.00	1.00	1.00	10.03	2.29
Player Profiles	0.96	0.94	0.95	13.15	7.09
UEFA	1.00	1.00	1.00	1.52	1.81
ATP World Tour	0.98	0.97	0.98	10.42	4.60
NFL	1.00	1.00	1.00	3.32	3.01
Soccer Base	0.98	1.00	0.99	382.47	32.61

Table 2.28: *Experimental results regarding Heuristic H11 .*



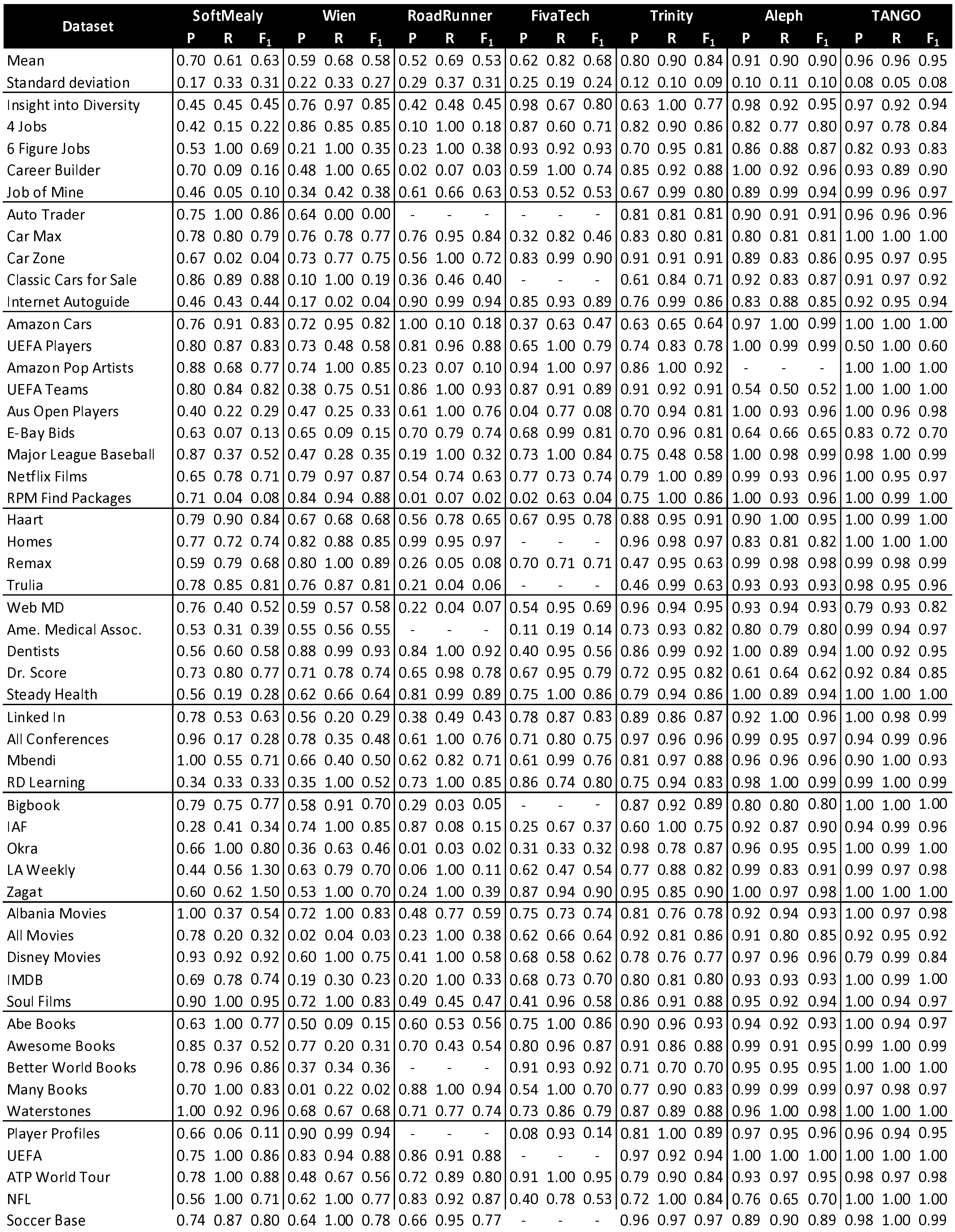
Best alternative from Heuristic H10 A0							
Summary	P	R	F_1	LT	ET	FR	K
Mean	0.96	0.96	0.95	16.30	3.69	-	0.59
Std. Dev.	0.08	0.05	0.08	39.63	4.83		
MDR	10.84	16.89	11.61	6.71	2.83		

Heuristic H11 A1							
Summary	P	R	F_1	LT	ET	FR	K
Mean	0.96	0.96	0.95	23.16	3.70	-	0.50
Std. Dev.	0.08	0.06	0.08	55.68	4.83		
MDR	10.84	16.68	11.57	9.63	2.84		

Table 2.29: *Ranking of alternatives regarding Heuristic H11 .*

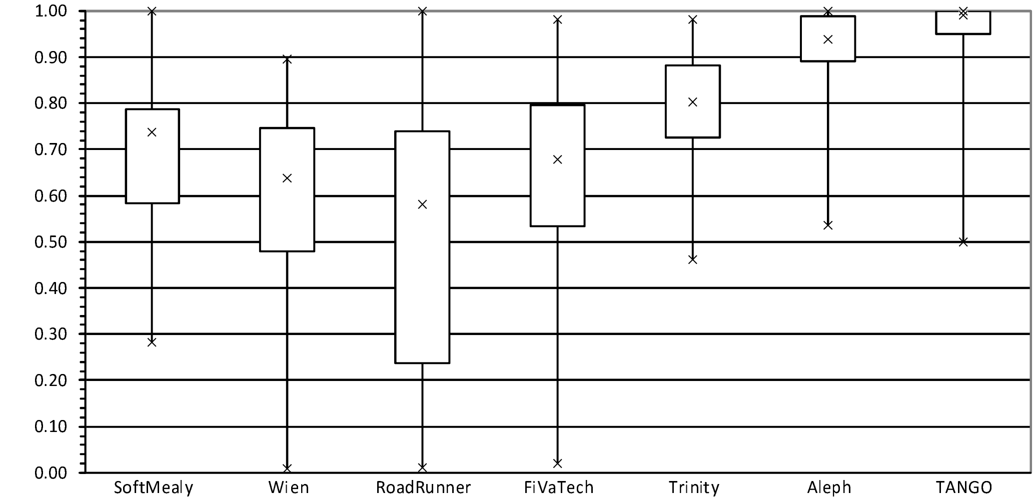
the first regarding precision, to the other techniques. Hommel's test confirms that the differences in rank amongst TANGO and Trinity, SoftMealy, FivaTech, Wien, and RoadRunner, are statistically significant because it returns adjusted p-values that are very small with regard to the significance level. There exists just one technique, Aleph, with which the statistical test did not find any significant differences since the adjusted p-value that corresponds to the comparison between TANGO and Aleph is not greater than the standard significance level. Thus, they both rank at the top. In other words, our experimental data provide enough evidence to reject the hypothesis that TANGO behaves similarly to Trinity, SoftMealy, FivaTech, Wien, and RoadRunner, regarding precision, that is, it supports the idea that TANGO can learn rules that are more precise than the others', but we cannot reject the hypothesis that TANGO behaves similarly to Aleph.

Table §2.32 summarises the results regarding recall. Empirically, TANGO seems to be the proposal that can achieve a higher recall and it is the one that seems more stable regarding this measure because its deviation is the smallest and its inter-quartile range is also the smallest. Note, however, that the other techniques can achieve results that are very good, too, chiefly Trinity and Aleph. Iman-Davenport's test returns a p-value that is very close to zero, which is a strong indication that there are differences in rank amongst the proposals that we have compared. Hommel's test confirms that the differences in rank amongst TANGO and the other techniques are statistically significant at the standard significance level. As a conclusion, the experimental data provide enough evidence to reject the hypothesis that TANGO behaves similarly to the other proposals regarding recall, that is, we can



Dataset	SoftMealy			Wien			RoadRunner			FivaTech			Trinity			Aleph			TANGO		
	P	R	F_1	P	R	F_1	P	R	F_1	P	R	F_1	P	R	F_1	P	R	F_1	P	R	F_1
Mean	0.70	0.61	0.63	0.59	0.68	0.58	0.52	0.69	0.53	0.62	0.82	0.68	0.80	0.90	0.84	0.91	0.90	0.90	0.96	0.96	0.95
Standard deviation	0.17	0.33	0.31	0.22	0.33	0.27	0.29	0.37	0.31	0.25	0.19	0.24	0.12	0.10	0.09	0.10	0.11	0.10	0.08	0.05	0.08
Insight into Diversity	0.45	0.45	0.45	0.76	0.97	0.85	0.42	0.48	0.45	0.98	0.67	0.80	0.63	1.00	0.77	0.98	0.92	0.95	0.97	0.92	0.94
4 Jobs	0.42	0.15	0.22	0.86	0.85	0.85	0.10	1.00	0.18	0.87	0.60	0.71	0.82	0.90	0.86	0.82	0.77	0.80	0.97	0.78	0.84
6 Figure Jobs	0.53	1.00	0.69	0.21	1.00	0.35	0.23	1.00	0.38	0.93	0.92	0.93	0.70	0.95	0.81	0.86	0.88	0.87	0.82	0.93	0.83
Career Builder	0.70	0.09	0.16	0.48	1.00	0.65	0.02	0.07	0.03	0.59	1.00	0.74	0.85	0.92	0.88	1.00	0.92	0.96	0.93	0.89	0.90
Job of Mine	0.46	0.05	0.10	0.34	0.42	0.38	0.61	0.66	0.63	0.53	0.52	0.53	0.67	0.99	0.80	0.89	0.99	0.94	0.99	0.96	0.97
Auto Trader	0.75	1.00	0.86	0.64	0.00	0.00	-	-	-	-	-	-	0.81	0.81	0.81	0.90	0.91	0.91	0.96	0.96	0.96
Car Max	0.78	0.80	0.79	0.76	0.78	0.77	0.76	0.95	0.84	0.32	0.82	0.46	0.83	0.80	0.81	0.80	0.81	0.81	1.00	1.00	1.00
Car Zone	0.67	0.02	0.04	0.73	0.77	0.75	0.56	1.00	0.72	0.83	0.99	0.90	0.91	0.91	0.91	0.89	0.83	0.86	0.95	0.97	0.95
Classic Cars for Sale	0.86	0.89	0.88	0.10	1.00	0.19	0.36	0.46	0.40	-	-	-	0.61	0.84	0.71	0.92	0.83	0.87	0.91	0.97	0.92
Internet Autoguide	0.46	0.43	0.44	0.17	0.02	0.04	0.90	0.99	0.94	0.85	0.93	0.89	0.76	0.99	0.86	0.83	0.88	0.85	0.92	0.95	0.94
Amazon Cars	0.76	0.91	0.83	0.72	0.95	0.82	1.00	0.10	0.18	0.37	0.63	0.47	0.63	0.65	0.64	0.97	1.00	0.99	1.00	1.00	1.00
UEFA Players	0.80	0.87	0.83	0.73	0.48	0.58	0.81	0.96	0.88	0.65	1.00	0.79	0.74	0.83	0.78	1.00	0.99	0.99	0.50	1.00	0.60
Amazon Pop Artists	0.88	0.68	0.77	0.74	1.00	0.85	0.23	0.07	0.10	0.94	1.00	0.97	0.86	1.00	0.92	-	-	-	1.00	1.00	1.00
UEFA Teams	0.80	0.84	0.82	0.38	0.75	0.51	0.86	1.00	0.93	0.87	0.91	0.89	0.91	0.92	0.91	0.54	0.50	0.52	1.00	1.00	1.00
Aus Open Players	0.40	0.22	0.29	0.47	0.25	0.33	0.61	1.00	0.76	0.04	0.77	0.08	0.70	0.94	0.81	1.00	0.93	0.96	1.00	0.96	0.98
E-Bay Bids	0.63	0.07	0.13	0.65	0.09	0.15	0.70	0.79	0.74	0.68	0.99	0.81	0.70	0.96	0.81	0.64	0.66	0.65	0.83	0.72	0.70
Major League Baseball	0.87	0.37	0.52	0.47	0.28	0.35	0.19	1.00	0.32	0.73	1.00	0.84	0.75	0.48	0.58	1.00	0.98	0.99	0.98	1.00	0.99
Netflix Films	0.65	0.78	0.71	0.79	0.97	0.87	0.54	0.74	0.63	0.77	0.73	0.74	0.79	1.00	0.89	0.99	0.93	0.96	1.00	0.95	0.97
RPM Find Packages	0.71	0.04	0.08	0.84	0.94	0.88	0.01	0.07	0.02	0.02	0.63	0.04	0.75	1.00	0.86	1.00	0.93	0.96	1.00	0.99	1.00
Haart	0.79	0.90	0.84	0.67	0.68	0.68	0.56	0.78	0.65	0.67	0.95	0.78	0.88	0.95	0.91	0.90	1.00	0.95	1.00	0.99	1.00
Homes	0.77	0.72	0.74	0.82	0.88	0.85	0.99	0.95	0.97	-	-	-	0.96	0.98	0.97	0.83	0.81	0.82	1.00	1.00	1.00
Remax	0.59	0.79	0.68	0.80	1.00	0.89	0.26	0.05	0.08	0.70	0.71	0.71	0.47	0.95	0.63	0.99	0.98	0.98	0.99	0.98	0.99
Trulia	0.78	0.85	0.81	0.76	0.87	0.81	0.21	0.04	0.06	-	-	-	0.46	0.99	0.63	0.93	0.93	0.93	0.98	0.95	0.96
Web MD	0.76	0.40	0.52	0.59	0.57	0.58	0.22	0.04	0.07	0.54	0.95	0.69	0.96	0.94	0.95	0.93	0.94	0.93	0.79	0.93	0.82
Ame. Medical Assoc.	0.53	0.31	0.39	0.55	0.56	0.55	-	-	-	0.11	0.19	0.14	0.73	0.93	0.82	0.80	0.79	0.80	0.99	0.94	0.97
Dentists	0.56	0.60	0.58	0.88	0.99	0.93	0.84	1.00	0.92	0.40	0.95	0.56	0.86	0.99	0.92	1.00	0.89	0.94	1.00	0.92	0.95
Dr. Score	0.73	0.80	0.77	0.71	0.78	0.74	0.65	0.98	0.78	0.67	0.95	0.79	0.72	0.95	0.82	0.61	0.64	0.62	0.92	0.84	0.85
Steady Health	0.56	0.19	0.28	0.62	0.66	0.64	0.81	0.99	0.89	0.75	1.00	0.86	0.79	0.94	0.86	1.00	0.89	0.94	1.00	1.00	1.00
Linked In	0.78	0.53	0.63	0.56	0.20	0.29	0.38	0.49	0.43	0.78	0.87	0.83	0.89	0.86	0.87	0.92	1.00	0.96	1.00	0.98	0.99
All Conferences	0.96	0.17	0.28	0.78	0.35	0.48	0.61	1.00	0.76	0.71	0.80	0.75	0.97	0.96	0.96	0.99	0.95	0.97	0.94	0.99	0.96
Mbendi	1.00	0.55	0.71	0.66	0.40	0.50	0.62	0.82	0.71	0.61	0.99	0.76	0.81	0.97	0.88	0.96	0.96	0.96	0.90	1.00	0.93
RD Learning	0.34	0.33	0.33	0.35	1.00	0.52	0.73	1.00	0.85	0.86	0.74	0.80	0.75	0.94	0.83	0.98	1.00	0.99	0.99	1.00	0.99
Bigbook	0.79	0.75	0.77	0.58	0.91	0.70	0.29	0.03	0.05	-	-	-	0.87	0.92	0.89	0.80	0.80	0.80	1.00	1.00	1.00
IAF	0.28	0.41	0.34	0.74	1.00	0.85	0.87	0.08	0.15	0.25	0.67	0.37	0.60	1.00	0.75	0.92	0.87	0.90	0.94	0.99	0.96
Okra	0.66	1.00	0.80	0.36	0.63	0.46	0.01	0.03	0.02	0.31	0.33	0.32	0.98	0.78	0.87	0.96	0.95	0.95	1.00	0.99	1.00
LA Weekly	0.44	0.56	1.30	0.63	0.79	0.70	0.06	1.00	0.11	0.62	0.47	0.54	0.77	0.88	0.82	0.99	0.83	0.91	0.99	0.97	0.98
Zagat	0.60	0.62	1.50	0.53	1.00	0.70	0.24	1.00	0.39	0.87	0.94	0.90	0.95	0.85	0.90	1.00	0.97	0.98	1.00	1.00	1.00
Albania Movies	1.00	0.37	0.54	0.72	1.00	0.83	0.48	0.77	0.59	0.75	0.73	0.74	0.81	0.76	0.78	0.92	0.94	0.93	1.00	0.97	0.98
All Movies	0.78	0.20	0.32	0.02	0.04	0.03	0.23	1.00	0.38	0.62	0.66	0.64	0.92	0.81	0.86	0.91	0.80	0.85	0.92	0.95	0.92
Disney Movies	0.93	0.92	0.92	0.60	1.00	0.75	0.41	1.00	0.58	0.68	0.58	0.62	0.78	0.76	0.77	0.97	0.96	0.96	0.79	0.99	0.84
IMDB	0.69	0.78	0.74	0.19	0.30	0.23	0.20	1.00	0.33	0.68	0.73	0.70	0.80	0.81	0.80	0.93	0.93	0.93	1.00	0.99	1.00
Soul Films	0.90	1.00	0.95	0.72	1.00	0.83	0.49	0.45	0.47	0.41	0.96	0.58	0.86	0.91	0.88	0.95	0.92	0.94	1.00	0.94	0.97
Abe Books	0.63	1.00	0.77	0.50	0.09	0.15	0.60	0.53	0.56	0.75	1.00	0.86	0.90	0.96	0.93	0.94	0.92	0.93	1.00	0.94	0.97
Awesome Books	0.85	0.37	0.52	0.77	0.20	0.31	0.70	0.43	0.54	0.80	0.96	0.87	0.91	0.86	0.88	0.99	0.91	0.95	0.99	1.00	0.99
Better World Books	0.78	0.96	0.86	0.37	0.34	0.36	-	-	-	0.91	0.93	0.92	0.71	0.70	0.70	0.95	0.95	0.95	1.00	1.00	1.00
Many Books	0.70	1.00	0.83	0.01	0.22	0.02	0.88	1.00	0.94	0.54	1.00	0.70	0.77	0.90	0.83	0.99	0.99	0.99	0.97	0.98	0.97
Waterstones	1.00	0.92	0.96	0.68	0.67	0.68	0.71	0.77	0.74	0.73	0.86	0.79	0.87	0.89	0.88	0.96	1.00	0.98	1.00	1.00	1.00
Player Profiles	0.66	0.06	0.11	0.90	0.99	0.94	-	-	-	0.08	0.93	0.14	0.81	1.00	0.89	0.97	0.95	0.96	0.96	0.94	0.95
UEFA	0.75	1.00	0.86	0.83	0.94	0.88	0.86	0.91	0.88	-	-	-	0.97	0.92	0.94	1.00	1.00	1.00	1.00	1.00	1.00
ATP World Tour	0.78	1.00	0.88	0.48	0.67	0.56	0.72	0.89	0.80	0.91	1.00	0.95	0.79	0.90	0.84	0.93	0.97	0.95	0.98	0.97	0.98
NFL	0.56	1.00	0.71	0.62	1.00	0.77	0.83	0.92	0.87	0.40	0.78	0.53	0.72	1.00	0.84	0.76	0.65	0.70	1.00	1.00	1.00
Soccer Base	0.74	0.87	0.80	0.64	1.00	0.78	0.66	0.95	0.77	-	-	-	0.96	0.97	0.97	0.89	0.90	0.89	0.98	1.00	0.99

Table 2.30: *Effectiveness results.*



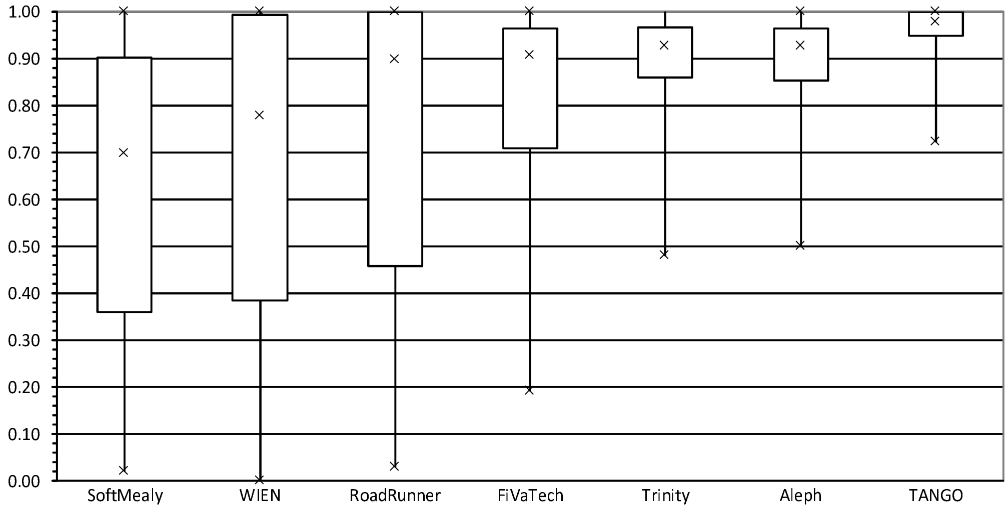
	SoftMealy	Wien	RoadRunner	FiVaTech	Trinity	Aleph	TANGO
Quartile 1	0.58	0.48	0.24	0.53	0.73	0.89	0.95
Minimum	0.28	0.01	0.01	0.02	0.46	0.54	0.50
Median	0.74	0.64	0.58	0.68	0.80	0.94	0.99
Maximum	1.00	0.90	1.00	0.98	0.98	1.00	1.00
Quartile 3	0.79	0.75	0.74	0.80	0.88	0.99	1.00

Sample ranking		Iman-Davenport's	Hommel's							Statistical ranking	
Technique	Rank	p-value	adjusted p-values							Technique	Rank
TANGO	1.60	5.76E-46		Aleph	Trinity	SoftMealy	FiVaTech	Wien	RoadRunner	TANGO	1
Aleph	2.33		TANGO	8.45E-02	9.10E-06	1.33E-11	1.02E-15	2.34E-18	4.21E-20	Aleph	1
Trinity	3.54									Trinity	2
SoftMealy	4.53									SoftMealy	2
FiVaTech	5.07									FiVaTech	2
Wien	5.38									Wien	2
RoadRunner	5.57									RoadRunner	2

Table 2.31: *Summary of results regarding precision.*

assume that TANGO ranks at the first position.

Table §2.33 summarises the results regarding the F_1 score. Empirically, TANGO seems to be the proposal that can achieve the best F_1 score, and it is, again, the most stable. Trinity and Aleph are also very stable, but their results regarding the F_1 score are a bit poorer. Iman-Davenport's test returns a p-value that is nearly zero, which strongly supports the hypothesis that there are statistically significant differences in rank. Hommel's test returns adjusted p-values that are clearly smaller than the significance level in every case, which supports the hypothesis that the differences in rank amongst TANGO and every other proposal are statistically significant, too; that is, we can safely assume that it ranks the first.



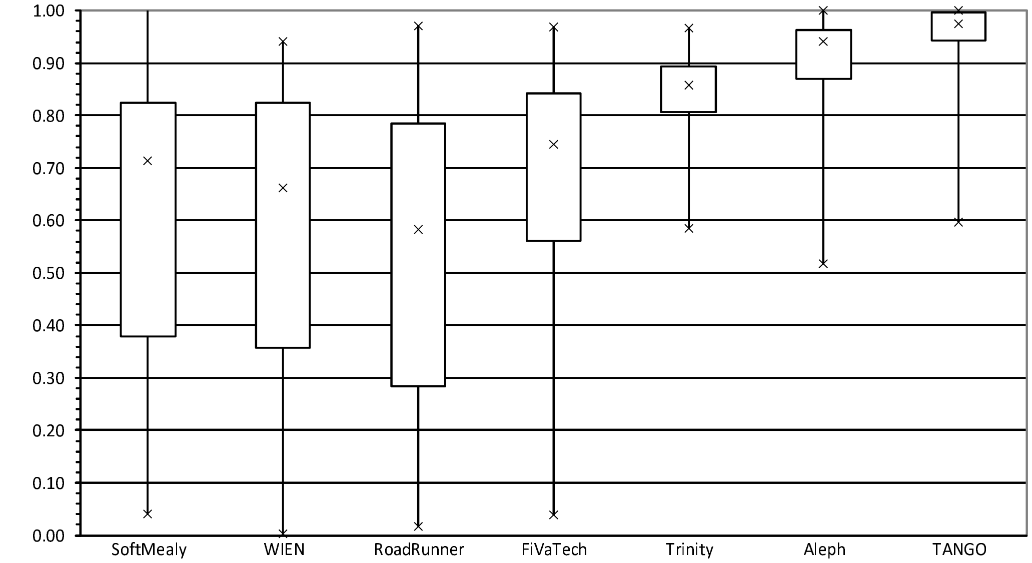
	SoftMealy	WIEN	RoadRunner	FiVaTech	Trinity	Aleph	TANGO
Quartile 1	0.36	0.38	0.46	0.71	0.86	0.85	0.95
Minimum	0.02	0.00	0.03	0.19	0.48	0.50	0.72
Median	0.70	0.78	0.90	0.91	0.93	0.93	0.98
Maximum	1.00	1.00	1.00	1.00	1.00	1.00	1.00
Quartile 3	0.90	0.99	1.00	0.96	0.97	0.96	1.00

Sample ranking		Iman-Davenport's	Hommel's							Statistical ranking	
Technique	Rank	p-value	adjusted p-values							Technique	Rank
				Trinity	Aleph	RoadRunner	FiVaTech	Wien	SoftMealy		
TANGO	2.52	6.42E-09	TANGO	3.11E-02	2.00E-03	1.94E-04	9.71E-05	1.31E-05	6.16E-09	TANGO	1
Trinity	3.43									Trinity	2
Aleph	3.91									Aleph	2
RoadRunner	4.21									RoadRunner	2
FiVaTech	4.31									FiVaTech	2
Wien	4.51									Wien	2
SoftMealy	5.11									SoftMealy	2

Table 2.32: *Summary of results regarding recall.*

Table §2.33 summarises the results regarding the F_1 score. Empirically, TANGO seems to be the proposal that can achieve the best F_1 score, and it is, again, the most stable. Trinity and Aleph are also very stable, but their results regarding the F_1 scorer are a bit poorer. Iman-Davenport's test returns a p-value that is nearly zero, which strongly supports the hypothesis that there are statistically significant differences in rank. Hommel's test returns adjusted p-values that are clearly smaller than the significance level in every case, which supports the hypothesis that the differences in rank amongst TANGO and every other proposal are statistically significant, too; that is, we can safely assume that it ranks the first.

Since TANGO works on the tree representation of the input documents, we need to parse them and correct the errors in their HTML code. Such er-



	SoftMealy	WIEN	RoadRunner	FiVaTech	Trinity	Aleph	TANGO
Quartile 1	0.38	0.36	0.28	0.56	0.81	0.87	0.94
Minimum	0.04	0.00	0.02	0.04	0.58	0.52	0.60
Median	0.71	0.66	0.58	0.74	0.86	0.94	0.97
Maximum	1.50	0.94	0.97	0.97	0.97	1.00	1.00
Quartile 3	0.83	0.82	0.78	0.84	0.89	0.96	1.00

Sample ranking		Iman-Davenport's	Hommel's							Statistical ranking	
Technique	Rank	p-value	adjusted p-values							Technique	Rank
				Aleph	Trinity	FiVaTech	SoftMealy	Wien	RoadRunner		
TANGO	1.53	7.33E-42	TANGO	1.52E-02	1.13E-05	3.00E-14	2.15E-15	1.26E-18	8.35E-19	TANGO	1
Aleph	2.56									Aleph	2
Trinity	3.45									Trinity	2
FiVaTech	4.81									SoftMealy	2
SoftMealy	4.96									FiVaTech	2
Wien	5.34									Wien	2
RoadRunner	5.36									RoadRunner	2

Table 2.33: *Summary of results regarding the* F_1 *score.*

rors are very common, cf. Table §A.1. As a conclusion, it was also necessary to carry out a statistical analysis to find out if our experiments provide enough evidence to conclude that the presence of errors in the input documents has an impact on the effectiveness of our proposal. We have used Kendall's Tau test, which returned $\tau = -0.10$ with p-value 0.59. Note that τ is very close to zero and that the p-value is clearly greater than the standard significance level, which means that the experimental data do not provide enough evidence to reject the hypothesis that the correlation is zero. In other words, our experiments do not provide any evidence that the effectiveness of our proposal may be biased by the presence of errors in the HTML code of the input documents.

Our conclusions are that TANGO outperforms the other proposals regarding effectiveness and that it is the proposal whose results are more stable. The statistical tests that we have performed have found enough evidence in our experimental data to support the hypothesis that the differences in the empirical rank amongst TANGO and the other proposals are significant at the standard significance level, except for the case of precision, in which case the experimental data do not provide enough evidence to conclude that TANGO and Aleph perform differently. Note, too, that proposals like RoadRunner, FiVaTech, and Aleph cannot deal with all of our datasets; in Table §2.30 such situations are indicated with a dash. The reason is that they took more than 1 CPU day to learn a rule or that they raised an exception; in both cases, we could not compute effectiveness measures for the corresponding datasets.

2.4.2 Efficiency analysis

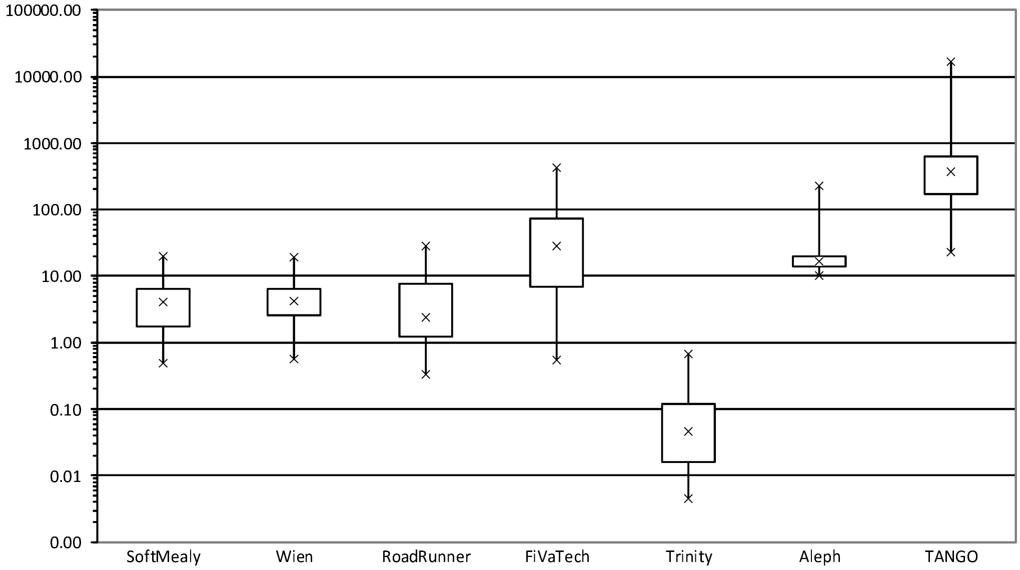
Table §2.34 reports on the raw efficiency data that we got from our experimentation. For each proposal, we report on its efficiency measures regarding our datasets. The first two lines also provide a summary of the results in terms of the mean value and the standard deviation of each measure. Since it is difficult to spot a trend in this table, we decided to summarise the data using boxplots.

Table §2.35 summarises the results regarding learning times, that is, the mean CPU time that each proposal took to learn a rule set. Experimentally, it seems that Trinity is the proposal that takes less time to learn a rule set; in most cases, it does not take more than a tenth of a second. It is followed by RoadRunner, SoftMealy, and Wien, whose learning times are very similar; then come Aleph and FivaTech. TANGO ranks at the last position, being the most inefficient. Iman-Davenport's test returns a p-value that is very close to zero, which clearly supports the hypothesis that there are differences in rank amongst these proposals. Hommel's test also returns adjusted p-values that are very small with respect to the significance level, which also reveals that the experimental data provide enough evidence to support the hypothesis that Trinity is the proposal that performs the best and that the others rank below it.

Table §2.36 summarises the results regarding extraction times, that is, the mean CPU time that it took to apply a rule set to a dataset. Wien, SoftMealy, Aleph, and TANGO seem to be the proposals that have the worst performance; RoadRunner and Trinity seem to be very similar in both mean

Dataset	SoftMealy		Wien		RoadRunner		FivaTech		Trinity		Aleph		TANGO	
	LT	ET	LT	ET	LT	ET	LT	ET	LT	ET	LT	ET	LT	ET
Mean	5.00	35.51	5.28	8.58	5.24	0.36	64.25	0.44	0.10	0.34	27.61	53.06	978.19	221.68
Standard deviation	4.33	38.35	3.88	9.78	6.20	0.48	104.17	0.58	0.14	0.47	36.73	47.34	2377.61	289.77
Insight into Diversity	4.90	11.67	3.20	4.99	2.41	1.00	9.35	0.08	0.05	1.00	13.40	50.57	1855.92	229.80
4 Jobs	4.07	36.60	6.09	6.57	1.40	1.00	8.32	0.07	0.04	0.01	17.08	37.44	289.08	104.22
6 Figure Jobs	7.65	18.51	7.44	8.32	13.83	1.00	53.21	0.24	0.02	0.01	12.20	71.17	2342.82	285.24
Career Builder	5.20	35.44	5.62	5.39	5.75	0.01	136.03	0.22	0.03	1.00	16.24	34.12	259.20	101.10
Job of Mine	3.63	15.61	2.86	3.32	1.49	0.01	30.33	0.14	0.03	0.01	15.42	30.23	246.48	83.76
Auto Trader	6.78	72.39	6.07	9.73	-	-	-	-	0.12	0.02	18.40	41.81	811.68	267.00
Car Max	9.72	22.91	3.50	5.34	13.27	0.01	19.24	0.14	0.14	0.01	15.54	32.91	426.96	175.02
Car Zone	3.43	28.88	5.76	3.32	2.72	1.00	198.79	0.41	0.02	1.00	15.23	30.35	455.22	241.74
Classic Cars for Sale	13.74	78.55	11.89	7.19	28.40	0.01	-	-	0.13	0.01	22.01	129.59	411.66	417.42
Internet Autoguide	4.33	68.24	4.01	6.26	4.42	1.00	71.50	1.00	0.05	1.00	15.73	31.38	475.80	178.68
Amazon Cars	0.68	7.02	8.41	5.66	0.80	1.00	2.55	1.00	0.01	1.00	13.39	11.00	22.62	25.92
UEFA Players	1.43	11.64	3.79	2.58	0.41	0.01	12.58	0.03	0.01	0.01	16.93	18.68	537.36	29.22
Amazon Pop Artists	3.61	16.20	7.96	10.22	1.03	1.00	107.04	0.08	0.01	0.01	-	-	202.68	153.96
UEFA Teams	0.49	3.33	0.80	2.28	0.47	0.01	0.54	0.01	0.02	1.00	11.74	15.23	170.76	89.64
Aus Open Players	0.81	15.27	5.26	16.96	3.12	1.00	80.70	0.18	0.14	1.00	21.67	147.52	2613.30	782.70
E-Bay Bids	1.14	11.11	5.59	13.65	2.11	0.01	397.98	0.34	0.25	0.06	22.60	119.96	590.46	336.84
Major League Baseball	1.33	13.67	4.39	4.28	1.75	0.01	184.47	1.00	0.02	0.01	21.28	13.79	1188.78	62.22
Netflix Films	2.26	23.88	4.85	31.55	4.73	0.01	399.01	0.53	0.08	0.01	17.60	105.26	941.46	348.18
RPM Find Packages	0.83	18.23	1.39	10.42	0.75	1.00	28.59	0.06	0.02	1.00	14.93	68.16	216.60	183.78
Haart	4.03	69.80	3.27	4.82	3.12	0.01	9.40	0.06	0.02	1.00	17.36	41.82	98.64	103.38
Homes	4.36	45.57	4.80	8.18	2.43	0.01	-	-	0.11	0.01	13.79	23.64	84.72	92.04
Remax	3.09	17.80	8.30	5.17	7.85	0.01	47.73	0.07	0.22	0.01	16.40	36.77	94.44	146.04
Trulia	11.87	196.63	12.94	25.37	19.97	1.00	-	-	0.48	1.00	18.95	119.56	4933.20	609.42
Web MD	4.54	20.85	10.47	10.17	14.49	0.01	7.64	1.00	0.01	0.01	18.24	46.42	241.62	154.26
Ame. Medical Assoc.	4.16	7.30	3.68	7.39	-	-	1.53	0.20	0.03	1.00	16.05	33.56	207.90	104.34
Dentists	1.56	5.67	1.36	1.28	0.39	0.01	4.86	0.05	0.01	1.00	12.78	9.40	29.82	30.54
Dr. Score	2.81	17.51	2.07	2.51	1.01	1.00	32.29	1.00	0.01	0.01	13.89	17.72	390.90	97.26
Steady Health	4.85	40.00	5.95	6.78	7.61	0.02	5.71	0.08	0.30	0.01	34.05	97.53	1010.58	312.30
Linked In	6.06	29.18	1.87	3.34	1.66	0.01	34.56	2.35	0.02	0.01	16.15	26.32	189.78	84.96
All Conferences	6.34	43.01	3.68	3.32	1.88	0.01	18.54	1.00	0.05	0.01	18.49	36.57	289.32	129.60
Mbendi	1.64	3.98	2.08	1.28	0.75	1.00	0.97	0.02	0.00	0.01	13.82	19.06	165.60	58.62
RD Learning	2.24	4.96	1.44	1.36	0.33	0.01	3.51	0.01	0.01	0.01	12.43	11.31	172.62	35.88
Bigbook	1.61	114.85	1.78	62.29	14.27	1.00	-	-	0.06	1.00	17.81	48.88	372.84	168.36
IAF	1.77	9.23	0.90	1.67	0.44	0.01	7.00	0.04	0.07	0.01	68.21	17.15	2855.22	123.18
Okra	0.78	150.01	1.13	24.48	10.39	1.00	425.36	0.26	0.06	0.01	228.41	28.76	545.28	81.12
LA Weekly	1.30	23.45	0.57	1.89	0.48	0.01	2.69	0.03	0.01	0.01	13.27	16.60	54.36	26.34
Zagat	1.50	14.11	5.83	13.75	3.85	1.00	73.51	0.04	0.07	1.00	13.54	19.61	35.28	45.90
Albania Movies	2.08	3.45	1.36	1.16	0.91	0.01	1.88	0.01	0.01	1.00	15.19	67.41	203.34	106.32
All Movies	11.54	38.87	3.23	4.11	1.90	0.01	6.23	1.00	0.27	0.01	125.19	37.21	777.78	499.14
Disney Movies	4.35	31.14	2.00	2.67	1.99	0.01	121.28	0.05	0.67	0.01	74.71	25.91	374.70	107.40
IMDB	19.89	63.25	19.47	11.47	9.92	0.01	32.13	2.32	0.24	1.00	111.09	68.47	2263.44	383.88
Soul Films	6.68	26.09	4.03	9.37	1.91	0.01	10.64	0.03	0.02	0.01	27.51	122.65	2422.62	243.18
Abe Books	9.95	18.71	10.74	10.27	3.29	0.01	9.78	0.09	0.02	0.01	15.06	32.12	206.46	112.62
Awesome Books	2.53	11.49	3.25	6.28	1.55	0.01	3.67	0.10	0.02	0.01	14.92	25.81	136.26	104.88
Better World Books	19.55	28.83	7.93	11.89	-	-	47.54	0.27	0.10	0.01	12.21	73.74	168.48	193.62
Many Books	7.39	21.03	2.82	5.62	1.31	0.01	82.71	0.09	0.13	0.01	10.08	49.78	518.10	154.98
Waterstones	5.00	53.46	5.58	6.03	3.47	1.00	29.37	1.38	0.04	0.01	11.45	49.49	419.64	143.76
Player Profiles	2.92	17.43	5.83	3.43	-	-	8.06	1.00	0.06	0.16	19.90	103.81	515.76	439.32
UEFA	4.00	23.08	2.69	3.97	6.89	0.02	-	-	0.03	0.01	18.20	31.93	57.30	102.72
ATP World Tour	9.31	125.86	11.81	12.76	8.87	0.03	50.01	0.58	0.38	0.02	19.60	56.60	403.68	276.00
NFL	6.34	27.66	9.69	6.64	19.88	0.02	72.49	1.00	0.08	0.01	16.55	43.50	122.88	197.04
Soccer Base	8.07	33.30	12.95	7.56	10.06	1.00	-	-	0.34	1.00	51.36	277.67	16444.44	1962.54

Table 2.34: *Efficiency results.*

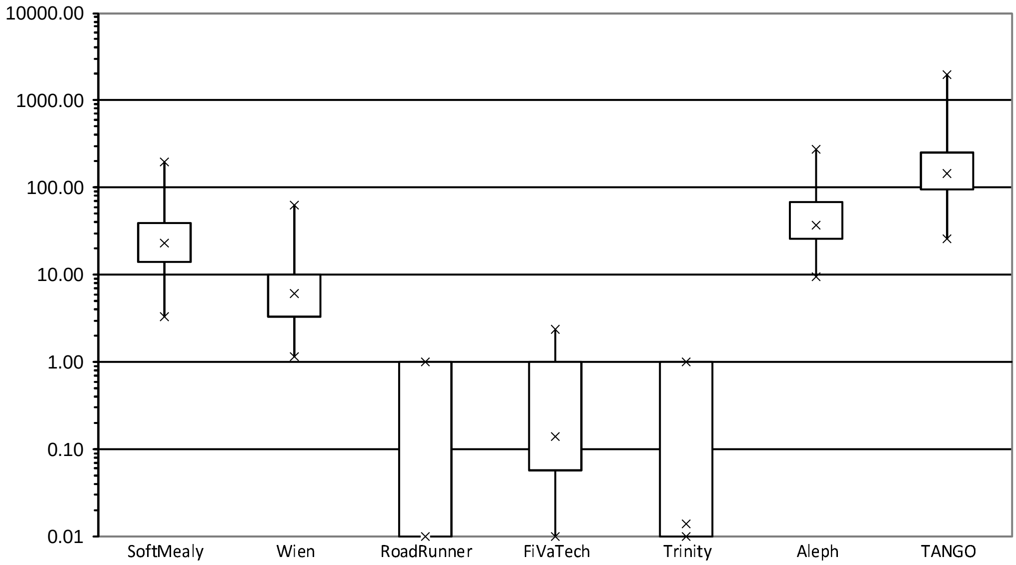


	SoftMealy	Wien	RoadRunner	FiVaTech	Trinity	Aleph	TANGO
Quartile 1	1.74	2.54	1.24	7.00	0.02	13.85	172.16
Minimum	0.49	0.57	0.33	0.54	0.00	10.08	22.62
Median	4.05	4.21	2.42	28.59	0.05	16.40	373.77
Maximum	19.89	19.47	28.40	425.36	0.67	228.41	16444.44
Quartile 3	6.43	6.43	7.67	72.49	0.12	19.75	637.29

Sample ranking		Iman-Davenport's	Hommel's							Statistical ranking	
Technique	Rank	p-value	adjusted p-values							Technique	Rank
Trinity	1.00			RoadRunner	SoftMealy	Wien	Aleph	FivaTech	TANGO	Trinity	1
RoadRunner	3.00	4.78E-111	Trinity	2.35E-06	1.19E-06	7.08E-08	2.79E-23	3.38E-24	4.37E-43	RoadRunner	2
SoftMealy	3.12									SoftMealy	2
Wien	3.37									Wien	2
Aleph	5.27									Aleph	2
FivaTech	5.37									FivaTech	2
TANGO	6.88									TANGO	2

Table 2.35: *Summary of results regarding learning times.*

extraction time and deviation since their inter-quartile ranges are identical; finally, FivaTech seems to be in the middle and its extraction time is still competitive. The timings regarding TANGO are the worst, since applying the rules learnt takes roughly 221.68 seconds in average; neither is its standard deviation small, which means that the results are not as stable as we wished. Iman-Davenport's test returns a p-value that is nearly zero, which clearly indicates that there are statistically significant differences in the empirical rank. Hommel's test returns adjusted p-values that are not smaller than the standard significance level regarding the comparisons of Trinity, which is the best-ranked proposal according to the empirical



	SoftMealy	Wien	RoadRunner	FiVaTech	Trinity	Aleph	TANGO
Quartile 1	14.00	3.34	0.01	0.06	0.01	25.86	95.96
Minimum	3.33	1.16	0.01	0.01	0.01	9.40	25.92
Median	23.00	6.15	0.01	0.14	0.01	36.77	144.90
Maximum	196.63	62.29	1.00	2.35	1.00	277.67	1962.54
Quartile 3	39.15	10.18	1.00	1.00	1.00	67.79	249.14

Sample ranking		Iman-Davenport's	Hommel's							Statistical ranking	
Technique	Rank	p-value	adjusted p-values							Technique	Rank
				RoadRunner	FiVaTech	Wien	SoftMealy	Aleph	TANGO		
Trinity	1.67	6.25E-159	Trinity	6.50E-01	1.25E-01	3.21E-08	1.82E-16	7.69E-21	5.56E-35	Trinity	1
RoadRunner	1.87									RoadRunner	1
FivaTech	2.46									FivaTech	1
Wien	4.10									Wien	2
SoftMealy	5.23									SoftMealy	2
Aleph	5.71									Aleph	2
TANGO	6.96									TANGO	2

Table 2.36: *Summary of results regarding extraction times.*

ranking, RoadRunner, and FivaTech. Therefore, we cannot reject the hypothesis that they behave statistically similarly regarding extraction times, that is, we have to assume that they all rank at the first position. The test, however, finds enough evidence to reject the hypothesis that the previous proposals and the others behave similarly regarding the extraction time.

As a conclusion, our experiments do not allow us to conclude that TANGO is as efficient as the other proposals. It was the worst performing regarding learning times and extraction times, which is an indication that we have to keep working on improving its efficiency. From a production point of view, learning times are not a big deal since learning rules is not such a frequent step. However, applying the rules learnt in a production scenario takes a time in the order of minutes, which is not as competitive from a practical point of view as the other techniques. We think that the efficiency results can be improved and this was the reason why we developed the information extractor that we describe in the next chapter.

2.5 Related work

In this section, we first provide an overview of the related proposals in the literature. Then, we discuss on the key features that differentiate our proposal from them. The discussion was organised along the following dimensions: the catalogue of features, the variation points, the rules, the learning procedure, and the evaluation procedure.

2.5.1 Overview of related proposals

TANGO is a supervised technique. Most current such techniques rely on ad-hoc Machine Learning algorithms that were specifically tailored to the problem of learning web information extraction rules. Most of them try to learn token or XPath patterns that are based on token lexemes, their lexical classes, or HTML tags and their attributes. That is, they are specifically tailored to analysing the HTML text of the DOM tree representation of the input documents in an attempt to find regularities that help learn extraction rules.

It is surprising that very few authors have attempted to map the input documents onto feature-based representations. Such representations allow to represent the input documents using HTML, DOM, rendering, or arbitrary user-defined features that may help learn additional regularities that result in better rules or rules that are easier to learn. The only few exceptions are the following ones: a) SRV [59, 62], which works on the textual view of the input documents. To learn a rule, it starts with the most general rule and then specialises it by adding conditions so that the resulting rule matches as many positive examples as possible whilst reducing the number of negative ones matched. The conditions are added one at a time and they are based on features that are selected from a predefined catalogue. The rules are represented using an ad-hoc language that requires an ad-hoc procedure to generate and analyse each kind of condition. b) Irmak and Suel's proposal [88] works on the DOM-tree view of the input documents and their rules are sets of conditions that work on XPaths. The algorithm creates several sets of extraction rules that generalise the user-annotated examples in different ways. Next, the user has to select the set of examples that best suits her or his interests, and the learning process is executed again to correct mistakes. Unfortunately, few details were disclosed regarding the learning algorithm. c) Bădică and others' proposal [19] also works on the DOM-tree view of the input documents. Their rules basically attempt to classify positive examples by means of their tags and/or the tags of their neighbours. The rule-learning procedure relies on the FOIL system [139] to learn a set of Horn clauses from a non-standard logic representation of the DOM-tree nodes and their features. d) One of the latest proposals was introduced by Fernández-Villamor and others [53]. It works on the DOM-tree view of the input documents and attempts to generalise an initial set of overly-specific rules, that is, rules that match a single positive example. Basically, it iterates the inner Cartesian product of this rule set and attempts to compress every two rules. The compression continues as long as the rule set being learnt improves on the F_1 score. The conditions can only constraint the values of a few attributive features and the only neighbour of a node that can be explored is its parent.

TANGO is closely related to SRV, Irmak and Suel's proposal, L-Wrappers, and Fernández-Villamor and others' proposal because it also works on a feature-based representation of the input documents.

2.5.2 Catalogue of features

For a machine learner to work, it has to be fed with a representation of the input documents that maps them onto a number of features from which it is possible to discern the difference between the information to be extracted and the information to be ignored. Since the Web and, consequently, the way that HTML is used is evolving continuously, it is likely that the catalogue of features has to be replaced from time to time. That is why it is of uttermost importance that it is open and that the proposal is not bound with the specific features that it provides.

SRV relies on a limited catalogue of features that hardly includes some HTML features and a few user-defined features, but no DOM or rendering features; it also includes relational features to navigate from one token to the next or the previous one, or to the first token of the next column, the previous column, the next row, the previous row, or the header when dealing with tables. SRV features can only be computed on tokens, not on sequences of tokens or nodes. Irmak and Suel's proposal builds on a catalogue that includes a subset of HTML and DOM features, plus some user-defined features; unfortunately, the authors did not provide many details on them. L-Wrappers relies on a unique attributive feature to map nodes onto their corresponding tags and four relational features: next sibling, parent, first child, and last child. As a result, the conditions in the rules basically attempt to classify nodes by means of their tags and the tags of their neighbours. Finally, Fernández-Villamor and others' proposal considers a unique HTML feature that is the tag of a node (restricted to link, image, and other), a subset of DOM features regarding bounding boxes, widths, heights, font size, font weight, font family (restricted to serif, sans-serif, and monospace), and a unique relational feature that allows to fetch the parent of a node.

TANGO relies on quite an extensive catalogue of attributive features that includes every HTML, DOM, and rendering feature defined by the W3C recommendations plus user-defined features; the catalogue of relational features includes features to fetch the parent of a node, its ancestors, its children, and siblings. The features are computed on nodes, which means that some of them work on their token sequences. The catalogue has been designed so that it can be easily replaced, since there is nothing in our proposal that is bound with the specific features provided by the catalogue. The catalogues of features that are provided by other proposals are very limited. In the case of SRV, it is open and can be replaced because there is nothing in the proposals that is specific to the features in the catalogue; in the case of Irmak and Suel's proposal, the catalogue seems to be open, too, but the authors did not provide many details; in the case of L-Wrappers or Fernández-Villamor and others' proposal the catalogue is closed since the proposals themselves can only work with the features that are provided in their catalogues.

2.5.3 Variation points

The variation points of a proposal identify the procedures for which different alternatives exist. A priori, it is not possible to make a decision regarding which the best implementation is because it depends on a variety of factors. So it is necessary to identify them, to identify some alternatives, and to have a method to make a decision regarding which the best configuration is.

Unfortunately, neither SRV, Irmak and Suel's proposal, L-Wrappers, nor Fernández-Villamor and others' proposal rely on variation points. The authors devised a number of algorithms that were configured to perform as well as possible, but it is not clear at all which of them need to be replaced in order to adapt the proposals. That is the common theme behind every proposal that we have surveyed, and we think that this is one of the reasons why they tend to fade away quickly as the Web evolves and the features on which they rely and the heuristics that they implement are not enough to make a difference between the information of interest and the information to be ignored.

In TANGO, we have carefully identified a number of variation points, namely: how to pre-process a learning set, how to post-process a rule set, how to select the candidates to extend a rule and to create savepoints, how to score a rule, how to check if a rule is too complex, how to branch a rule into a number of candidate conditions, how to bound candidate conditions, and how to check if a candidate is promising. We have also devised a method that allows to decide which configuration is the best performing as objectively as possible.

2.5.4 Rules

All of the related proposals learn Horn-like rules, but they differentiate regarding the conditions that can be used in their bodies. This implies that they have different expressiveness levels.

SRV's rules rely on the following kinds of conditions: checking the length of a token, checking that a token has a given value for a feature, checking that every token in a positive example has a given value for a feature, checking the position of a token, and checking the distance between two tokens. The conditions cannot be negated and slot instantiators are not allowed, which means that the rules cannot be recursive. Irmak and Suel's proposal rely on conditions that can be applied to either element nodes, e.g., checking that the tag is a given one or checking that an attribute has a given value, or text nodes, e.g., checking that it matches a given regular expression or checking that it is the i-th child; it is not clear if their proposal can deal with negated conditions or recursion; neither is it clear if inequalities are allowed. L-Wrappers' rules rely on two types of conditions only: checking whether a node has a given tag and fetching a neighbour; the authors researched regarding using negated conditions and came to the conclusion that they were not useful with their catalogue of features because they only helped identify nodes without a left sibling, i.e., the first child, or nodes without a right sibling, i.e., the last child; furthermore, they did not explore recursion. Fernández-Villamor and others' rules rely on two kinds of conditions: comparators to constraint the tag, the width, the height, the font size, the font weight, or the font family and parent instantiators.

The main difference is that TANGO's rules rely on slot instantiators (which allow for recursive rules), feature instantiators (which allow to instantiate any feature in the catalogue, if possible), and comparators (which help constraint the values of attributive features); furthermore, the conditions can be negated. These kinds of conditions have proven to work well in our experiments and they can also be represented very straightforwardly in Prolog, which allows to execute them using any available Prolog engine.

2.5.5 Learning procedure

Learning procedures can be top-down or bottom-up. In the former case, the search for rules starts with overly-general rules that match every example in the learning set; it then adds conditions that constraint the examples that are matched, and the process continues until a rule that matches at least a positive example and no negative example is found. In the latter case, the search starts with overly-specific rules that match a single positive example; it then generalises or drops some conditions so that the resulting rules match as many positive examples as possible, and the process continues until no further generalisation is possible. In practice, both approaches have proven to work well, even though the bottom-up approach has got some criticism regarding information extraction [59]. TANGO is a top-down proposal, so we restrict our attention to SRV and L-Wrappers, which are also top-down.

A difference with regard to SRV and L-Wrappers is that TANGO is intended to learn extraction rules for slots that are structured hierarchically. That is, TANGO can deal with data models in which the information to be extracted is represented by means of records that are composed of attributes or further nested records. In other words, TANGO first learns rules to extract first-level slots and then creates specific learning sets to learn additional rules to extract their nested records or attributes. This approach has proven to work very well in practice because it reduces the size of the learning sets significantly. Furthermore, it is a sensible approach to work with documents that have listings of records, since, otherwise, it would not be easy to identify which slots are nested into which slots.

There are many additional differences between TANGO and SRV, namely: a) SRV's learning process requires an ad-hoc procedure for each type of condition. Furthermore, it also requires an ad-hoc procedure to generate the first condition in the body of a rule. Such first condition is of the form $some(T, L, F, V)$, where T is a variable that can be bound to a token inside a positive example, L denotes a sequence of relational features that allow to navigate from that token to its neighbours, F denotes a feature, and V a value for that feature. In other words, these conditions are intended to check that a token has a given value for a feature. Unfortunately, if more features of that node have to be constrained, the token must be re-bound. This means that tokens whose features help discern well amongst positive and negative examples need to be rebound several times. This might have a negative impact on efficiency because this requires to search the whole condition space several times, which also includes exploring and evaluating the same conditions several times. An additional intricate implication of re-bounding is that tokens that belong to negative examples and tokens that belong to positive examples cannot be compared regarding their relative positions. TANGO does not require ad-hoc procedures to generate different types of conditions; it relies on a variation point called BRANCH that generates every condition that might possibly be added to a rule; furthermore, TANGO does not consider every possible condition as a candidate, but implements a heuristic to bound the conditions and generate a subset of candidates. Neither does TANGO suffer from the re-binding problem in SRV since a relational feature instantiator can bind any node to a variable, which allows to analyse as many attributive features as necessary in the forthcoming steps. b) SRV has many problems to compute the negative examples. Such examples include every subsequence of tokens in the input documents that is not explicitly annotated as a positive example; the problem is that a document with n tokens has $O(n^2)$ possible subsequences of tokens, which are typically too many to be computed explicitly. As a consequence, SRV has to introduce a hard bias regarding the size of the negative examples that are considered. In TANGO, computing the negative examples is as easy as fetching the set of nodes that are not explicitly labelled with a user-defined slot. c) SRV does not take into account any heuristic to select the best candidate conditions to be added to a rule; it just computes their gains and selects the condition that provides more gain. Contrarily, TANGO relies on a variation point since it is not clear which the best heuristic can be. In our experiments, we have proven that the heuristic that we propose is quite effective but it can be replaced very easily if the catalogue of features evolves. d) SRV stops searching for a rule when it finds a solution, independently from how complex it is. TANGO includes a variation point that allows to stop exploring a rule when it becomes too complex. Basically, this prevents TANGO from learning very specific rules that work well on the learning set but do not generalise well in a production setting. SRV only includes a simple heuristic to prevent learning too specific rules: it discards conditions that result in rules that match less than five positive examples, which can be problematic when dealing with detail documents that report on a single item since they typically provide only one positive example of each slot or a few ones in the case of multi-valued slots. e) SRV implements a forward-only learning procedure, which means that it cannot backtrack from bad decisions. This implies that it has to return the first rule that it finds, even in cases in which there are some candidates that might result in a rule that matches more positive examples. Since the search process is blind and there is not a guarantee that a rule that currently matches more positive examples might actually lead to a solution, SRV has to select the first solution that it finds. Contrarily, TANGO implements a savepoint mechanism that allows it to explore promising rules and backtrack if they are finally found not to be good enough. f) SRV did not take into account that preprocessing the learning set might have an impact on the efficiency, whereas TANGO has proven that reducing the negative examples is appropriate. g) SRV does not post-process the results, whereas TANGO has proven that post-processing them may result in simpler rules, although it increases the learning time.

There are also many differences regarding L-Wrappers, which basically consists in mapping the input documents onto a knowledge base and then using the FOIL system [139] to learn extraction rules. FOIL is a general-purpose inductive logic system and it was not tailored to the problem of information extraction; unfortunately, it did not prove to be efficient enough as it was used in L-Wrappers. The authors mentioned that their approach is infeasible in practice when a record has more than two attributes (records are flat in this proposal, i.e., they are actually tuples). Due to this problem, they had to design a complementary approach that learns to extract pairs of attributes and then merges the results into a single rule. The main problem is regarding the exponential explosion of negative examples, which was estimated in the order of $O(n^k)$ for a document with n nodes and records with k attributes. Negative examples are computed by the FOIL system using the Closed World Assumption. Unfortunately, this is inefficient because FOIL has to examine every possible instantiation of every possible feature on every possible node; in practice, the authors had to reduce the number of negative examples to roughly 0.10% for their approach to be manageable; it is not clear whether that reduction works well in a general setting because the proposal was evaluated on very few datasets. Merging can alleviate the problem, but does not solve it completely because it requires to compute a rule for each of the pairs of attributes in a record. This approach may be problematic insofar missing or permuted attributes and different formattings increase the number of pairs significantly. TANGO learns rules to extract the positive examples independently from each other, which is more efficient and resilient to missing and permuted attributes or alternating formats.

2.5.6 Evaluation process

Unfortunately, none of the most closely-related proposals were evaluated on a sufficiently large number of datasets; neither were they compared using statistically-sound methods.

SRV was evaluated on three datasets and it was empirically compared with two naive baselines by the same authors. Irmak and Suel's paper [88] focused on evaluating their proposal on fourteen datasets; four of them had been used to evaluate previous traditional information extractors with which this proposal was compared; the others were gathered from more up-to-date web sites, but they did not conduct an exhaustive experimentation or an empirical comparison with other proposals in the literature; furthermore, the way that they computed the effectiveness of their proposal was not the standard one because they used a so-called verification set that was used to request feedback from the user and correct the extraction rules. In L-Wrappers, the authors focused on evaluating their proposal on a single dataset on which it worked reasonably well, but did not conduct an exhaustive experimentation or an empirical comparison with other proposals in the literature. Finally, Fernández-Villamor and others' paper [53] reports on an experimentation with three datasets; no empirical comparison with other techniques was provided.

Contrarily, TANGO has been evaluated on 52 datasets and it was empirically compared to 6 other state-of-the-art techniques in the literature; our conclusions were supported by means of statistically-sound methods that proved that the differences amongst TANGO and the other proposals are statistically significant.

2.6 Summary

In this chapter, we have presented TANGO, which is a new proposal to learn web information extraction rules in the context of semi-structured web documents. It relies on an open catalogue of features and a number of variation points; both the catalogue of features and the variation points are intended to help evolve it when necessary. This clearly deviates from the many existing ad-hoc proposals in the literature; it is closely related to four proposals that also rely on feature-based representations of the input documents, but deviates significantly from them regarding the approach used to solve the problem. We have performed an exhaustive experimental study to configure our proposal with the best possible heuristics to implement each variation point. The result is a system that has proven to beat others in the literature regarding effectiveness, but needs to be improved regarding efficiency, which motivated us to work on the system that we describe in the next chapter.

Chapter 3

ROLLER: a propositio-relational learner

We describe ROLLER in this chapter, which is a propositio-relational approach to learn web information extraction rules. It is organised as follows: Section §3.1 presents our motivation and sketches our system; Section §3.2 describes the details of our proposal; Section §3.3 reports on how we have configured it so that it can achieve its best results; then, the results of our experimental analysis are presented in Section §3.4; Section §3.5 presents the related work and a detailed comparison with our proposal; Section §3.6 summarises our conclusions. Appendices §A and §B report, respectively, on our experimental environment and the performance measures that we have used.

3.1 Introduction

Unfortunately, inductive logic programming proposals are typically difficult to scale as the number of documents or features increases and the experimentation carried out with TANGO in the previous chapter has proved it. Furthermore, the myopia effect is another common remarkable drawback of these proposals. The problem is that standard inductive logic programming proposals do not look ahead; that is, when a feature instantiator is selected, it does not entail that the rule is going to be improved in the following steps by adding comparators that constrain the values of the corresponding feature/s. This makes these systems more likely to make wrong choices when selecting feature instantiators. In TANGO, wrong choices only contributed to learn more complex rules that relied on a few useless feature instantiators. However, as the catalogue of features is quite extensive and the features are powerful, the learner was always able to find rules with high effectiveness with no need to perform backtracking to recover from bad choices.

In the general field of Machine Learning, there exist propositio-relational proposals [98] that attempt to provide effective and efficient means to learn from relational data using propositional techniques, but they have seldom been explored regarding web information extraction. The only partial exception is the work by Sleiman and Corchuelo [161], who introduced an approach that combines automata and neural networks. Regarding myopia, propositio-relational proposals can easily ensure that every condition that is added to a rule actually contributes to improving it. That is, they are not likely to make wrong choices from which they need to recover.

In this chapter, we introduce a new propositio-relational approach called ROLLER, which is intended to learn web information extraction rules. Our contributions to the field are the following: we have devised a new propositio-relational technique that relies on a search procedure that uses a dynamic flattening technique to explore the context of the nodes that provide the information to be extracted; it needs to be configured with an open catalogue of features, which helps it adapt as the Web evolves, plus a base learner and a rule scorer, which helps it leverage the continuous advances in the general field of Machine Learning. We have conducted an extensive experimental analysis that proves that our proposal outperforms other state-of-the-art proposals regarding effectiveness; regarding efficiency, our results prove that it is comparable to the best ones. The conclusions that we have drawn from our experimental analysis have been confirmed using standard statistical hypothesis tests in the literature.

3.2 Description of our proposal

ROLLER works on a set of documents that are represented using DOM trees and an annotation. The documents provide examples of how the information to extract is encoded and the annotation assigns each of their nodes to a slot that classifies the information that it provides. (There is an implicit null slot to which the nodes that do not provide any information to be extracted are assigned by default.) The documents are assumed to provide information on a given topic and to have regularities that help learn the rule.

The main algorithm first computes a number of attributive and relational features on the input documents. Such features are not intrinsic to our proposal; on the contrary, we assume that the user provides a procedure called FEATUREBUILDER to compute them; in other words, our proposal relies on an open catalogue of features that allows it to evolve as the Web evolves. The attributive features are then used to assemble a learning set from which a rule set is learnt. Neither is the base learner used intrinsic to our proposal; on the contrary, any technique in the literature that can work with multi-class problems using both numeric and categoric features can be plugged into our proposal using a user-provided procedure to which we refer to as BASELEARNER. The initial rule is then evaluated on the previous learning set using a user-defined rule scorer to which we refer to as RULESCORER. The main algorithm in ROLLER loops as long as a rule that is a solution is not found and the current rule can be expanded to a new rule that provides some score gain. The expansion procedure explores the context of every node, that is, the neighbour nodes according to the available relational features; it constructs several contexts around each node and selects the one whose attributive features help learn a better rule.

In the following subsections, we first present some preliminaries, then introduce the main procedures in our proposal, and, finally, describe some ancillary procedures to deal with learning sets and feature vectors.

3.2.1 Preliminaries

Next, we present the mathematical notation that we use to describe our proposal and then describe and formalise the concepts on which it relies; we illustrate every concept by means of quite a complete running example.

Definition 3.1 (Mathematical notation) *We use the standard mathematical notation to represent variables, sets, logical formulae, and the like. We would*

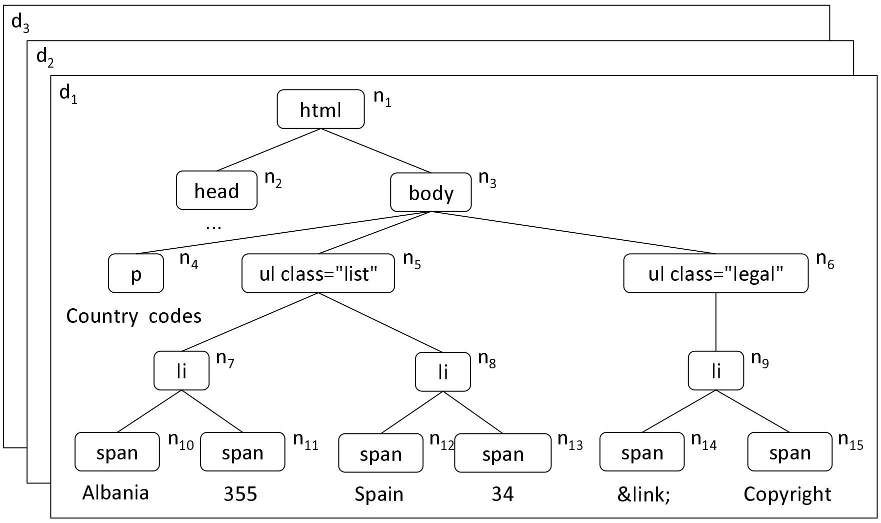


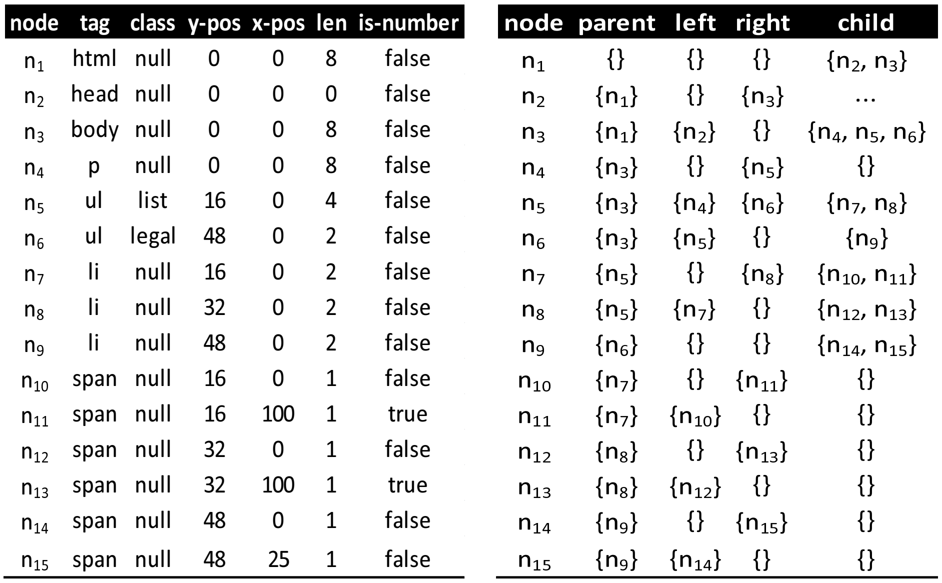
Figure 3.1: *Sample documents.*

like to highlight only a few pieces of notation for which we have not found a standard in the literature, namely: given a set X*, we denote the set of subsets of* X *as* $\mathbb{P}\,X$ *and the set of sequences of elements of* X *as* $\operatorname{seq} X$*; given elements* $x_1, x_2, \ldots, x_n$*, then* $\langle x_1, x_2, \ldots, x_n \rangle$ *denotes a sequence of them; given two sequences* s_1 *and* s_2*, we denote their concatenation as* $s_1 \oplus s_2$*; given two sets* X *and* Y*, we denote the set of maps from* X *onto* Y *as* $X \rightarrow Y$ *and we denote the tuples of which the map is composed as* $\{x \mapsto y\}$*, where* $x \in X$ *and* $y \in Y$*; given a map* M*, we denote its domain as* $\operatorname{dom} M$ *and its range as* $\operatorname{ran} M$*; maps are applied using the usual functional notation, e.g.,* $y = M(x)$*, where* $x \in X$ *and* $y \in Y$*. Given a map* M*, we denote its inverse as* M^{-1}*.*

Definition 3.2 (Documents and nodes) *Documents are character strings that adhere to the HTML syntax and can then be represented as DOM nodes [80, 176].*

Example 3.1 *Figure §3.1 illustrates a collection with documents* $\{d_1, d_2, d_3\}$*. We show a partial view of document* d_1*, which we use as a running example through the rest of this section. The set of nodes includes* $\{n_1, n_2, \ldots, n_{15}\}$*, plus the children of the head element, which are not shown, and the nodes that correspond to documents* d_2 *and* d_3*.*

Definition 3.3 (Features) *Features can be either attributive or relational. An attributive feature maps a node onto a value that represents either an HTML*



node	tag	class	y-pos	x-pos	len	is-number
n_1	html	null	0	0	8	false
n_2	head	null	0	0	0	false
n_3	body	null	0	0	8	false
n_4	p	null	0	0	8	false
n_5	ul	list	16	0	4	false
n_6	ul	legal	48	0	2	false
n_7	li	null	16	0	2	false
n_8	li	null	32	0	2	false
n_9	li	null	48	0	2	false
n_{10}	span	null	16	0	1	false
n_{11}	span	null	16	100	1	true
n_{12}	span	null	32	0	1	false
n_{13}	span	null	32	100	1	true
n_{14}	span	null	48	0	1	false
n_{15}	span	null	48	25	1	false

(a) Sample attributive features

node	parent	left	right	child
n_1	{}	{}	{}	$\{n_2, n_3\}$
n_2	$\{n_1\}$	{}	$\{n_3\}$	...
n_3	$\{n_1\}$	$\{n_2\}$	{}	$\{n_4, n_5, n_6\}$
n_4	$\{n_3\}$	{}	$\{n_5\}$	{}
n_5	$\{n_3\}$	$\{n_4\}$	$\{n_6\}$	$\{n_7, n_8\}$
n_6	$\{n_3\}$	$\{n_5\}$	{}	$\{n_9\}$
n_7	$\{n_5\}$	{}	$\{n_8\}$	$\{n_{10}, n_{11}\}$
n_8	$\{n_5\}$	$\{n_7\}$	{}	$\{n_{12}, n_{13}\}$
n_9	$\{n_6\}$	{}	{}	$\{n_{14}, n_{15}\}$
n_{10}	$\{n_7\}$	{}	$\{n_{11}\}$	{}
n_{11}	$\{n_7\}$	$\{n_{10}\}$	{}	{}
n_{12}	$\{n_8\}$	{}	$\{n_{13}\}$	{}
n_{13}	$\{n_8\}$	$\{n_{12}\}$	{}	{}
n_{14}	$\{n_9\}$	{}	$\{n_{15}\}$	{}
n_{15}	$\{n_9\}$	$\{n_{14}\}$	{}	{}

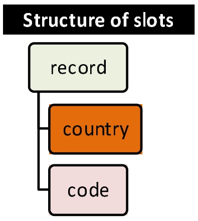
(b) Sample relational features

Table 3.1: *Sample features.*

attribute, which is specified in the HTML code of a document [80], a DOM attribute or a rendering attribute [176], which are computed by a browser, or a user-defined attribute. A relational feature maps a node onto a set of nodes with which the former is related by means of a neighbouring relationship; note that the target set of nodes may be empty, which means that the source node is not related to any other according to the corresponding relational feature.

Example 3.2 *Table §3.1 illustrates some of the features of the nodes of which the documents in Figure §3.1 are composed.* node *represents the node being examined;* tag *and* class *represent its HTML tag and its CSS class, respectively;* y-pos *and* x-pos *represent the ordinate and the abscissa of the corresponding rendering box, respectively;* len *and* is-number *represent the number of tokens in the text that is associated with the node and whether it is a number or not, respectively.*

Definition 3.4 (Annotations and slots) *An annotation is a map that associates nodes with slots. Intuitively, the slots provide a meaning to the nodes in a document. We implicitly assume that there is a special slot called* null *that indicates that a node does not provide any interesting informa-*



node	slot	node	slot	node	slot
n_1	null	n_6	null	n_{11}	code
n_2	null	n_7	Record	n_{12}	country
n_3	null	n_8	Record	n_{13}	code
n_4	null	n_9	null	n_{14}	null
n_5	null	n_{10}	country	n_{15}	null

Table 3.2: *Sample annotation.*

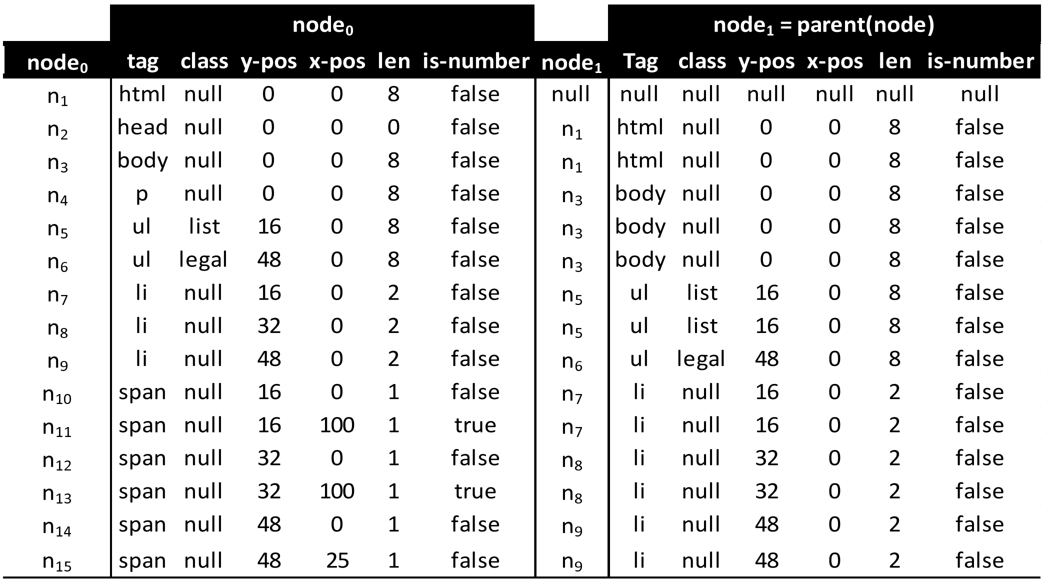
tion. The nodes that belong to the null *slot are referred to as negative examples and the others as positive examples.*

Example 3.3 *Table §3.2 presents the annotation that corresponds to document* d_1 *in Figure §3.1. The set of slots is* $\{\mathrm{Record}, \mathrm{country}, \mathrm{code}, \mathrm{null}\}$*, where* Record *labels the records to be extracted, which are composed of a country name that is denoted as* $\mathrm{country}$ *and a phone code that is denoted as* code*.*

Definition 3.5 (Contexts and bindings) *A context is a sequence of tuples of the form* $(\mathrm{t}, \mathrm{rf}, \mathrm{s})$*, where* t *denotes a target variable,* rf *denotes a relational feature, and* s *denotes a source variable. Both* rf *and* s *can be* null*, which indicates that it is an initial context tuple in which there is no source variable or relational feature involved; such a context tuple indicates that* t *can be bound to any of the nodes in the input documents. If* s *and* rf *are not* null*, then the tuple is interpreted as an expression of the form* $\mathrm{t} = \mathrm{rf}(\mathrm{s})$*, that is,* t *is bound to the result of applying relational feature* rf *to* s*. Simply put, a context is a symbolic representation of a binding; the binding itself is a map in which the variables in a context are bound to their corresponding nodes. Given a context tuple* $\mathrm{c} = (\mathrm{t}, \mathrm{rf}, \mathrm{s})$*, we introduce the following projection functions:* $\text{target}\,\mathrm{c} = \mathrm{t}$*,* $\text{relation}\,\mathrm{c} = \mathrm{rf}$*, and* $\text{source}\,\mathrm{c} = \mathrm{s}$*.*

Example 3.4 *Regarding the documents shown in Figure §3.1, context* $\langle(\mathrm{node}_0, \mathrm{null}, \mathrm{null}), (\mathrm{node}_1, \mathrm{parent}, \mathrm{node}_0)\rangle$ *sets variable* node_0 *to the nodes in the input documents and then variable* node_1 *to their parents.*

Definition 3.6 (Datasets and rules) *A dataset maps nodes onto vectors of attributive features that correspond to the nodes themselves or to some neighbours, which are introduced by means of a context. The datasets that are used to learn rules are referred to as learning sets and the datasets that are used to assess rules are referred to as test sets. Note that there is not a structural difference between them; the difference is regarding how they are used.*



	$node_0$							$node_1$ = parent(node)					
$node_0$	tag	class	y-pos	x-pos	len	is-number	$node_1$	Tag	class	y-pos	x-pos	len	is-number
n_1	html	null	0	0	8	false	null	null	null	null	null	null	null
n_2	head	null	0	0	0	false	n_1	html	null	0	0	8	false
n_3	body	null	0	0	8	false	n_1	html	null	0	0	8	false
n_4	p	null	0	0	8	false	n_3	body	null	0	0	8	false
n_5	ul	list	16	0	8	false	n_3	body	null	0	0	8	false
n_6	ul	legal	48	0	8	false	n_3	body	null	0	0	8	false
n_7	li	null	16	0	2	false	n_5	ul	list	16	0	8	false
n_8	li	null	32	0	2	false	n_5	ul	list	16	0	8	false
n_9	li	null	48	0	2	false	n_6	ul	legal	48	0	8	false
n_{10}	span	null	16	0	1	false	n_7	li	null	16	0	2	false
n_{11}	span	null	16	100	1	true	n_7	li	null	16	0	2	false
n_{12}	span	null	32	0	1	false	n_8	li	null	32	0	2	false
n_{13}	span	null	32	100	1	true	n_8	li	null	32	0	2	false
n_{14}	span	null	48	0	1	false	n_9	li	null	48	0	2	false
n_{15}	span	null	48	25	1	false	n_9	li	null	48	0	2	false

(a) Sample dataset.

$$\begin{array}{l}
\langle \\
\quad \mathsf{node}_0.\mathsf{x\text{-}pos} \geq 100 \Rightarrow \mathsf{slot} = \mathsf{code}, \\
\quad \mathsf{node}_0.\mathsf{tag} = \mathsf{span} \wedge \mathsf{node}_1.\mathsf{y\text{-}pos} \geq 16 \wedge \mathsf{node}_1.\mathsf{y\text{-}pos} \leq 32 \Rightarrow \mathsf{slot} = \mathsf{country}, \\
\quad \mathsf{node}_1.\mathsf{y\text{-}pos} \leq 0 \Rightarrow \mathsf{slot} = \mathsf{null}, \\
\quad \mathsf{node}_1.\mathsf{y\text{-}pos} \geq 48 \Rightarrow \mathsf{slot} = \mathsf{null}, \\
\quad \Rightarrow \mathsf{slot} = \mathsf{Record} \\
\rangle,
\end{array}$$

(b) Sample rule set.

Table 3.3: *Sample dataset and rule set.*

Example 3.5 *Table §3.3(a) shows a sample dataset in which the context involves the nodes in the input documents and their parents. Columns* node_0 *and* node_1 *present the corresponding bindings. Table §3.3(b) shows a sample rule that was learnt from the previous dataset; it is a sequence of the form* $\langle r_1, r_2, \ldots, r_n \rangle$ *(*$n \geq 1$*), where each component* r_i *is of the form* $c_{i,1} \wedge c_{i,2} \wedge \ldots \wedge c_{i,k_i} \Rightarrow \mathsf{slot} = s$ *(*$i = 1 \mathinner{..} n, k_i \geq 0$*); each* $c_{i,j}$ *is a simple condition of the form* $n.f\,\theta\,v$*, where* n *denotes a target variable in a context,* f *denotes an attributive feature,* θ *is a comparator, and* v *is a value. Given a node to classify, it is first transformed into its corresponding vector, and then the components of the rule are applied in sequence; the last component as-*

```
method $\mathrm{ROLLER}(D, A)$
  – Step 1: compute features.
  $(AF, RF) = \textsc{featureBuilder}(D)$
  – Step 2: learn an initial rule.
  $c = (\mathit{node}_0, \mathit{null}, \mathit{null})$
  $C = \langle c \rangle$
  $B = \{\mathit{node}_0 \mapsto \mathrm{dom}\, A\}$
  $LS = \mathit{createLearningSet}(c, AF, RF, A)$
  $r = \textsc{baseLearner}(LS, A)$
  – Step 3: find an expansion.
  $\mathit{keepSearching} = (\textsc{ruleScorer}(r, LS) \neq 1.00)$
  while $\mathit{keepSearching}$ do
    $(C, B, LS, r') = \mathit{findExpansion}(C, B, LS, r, AF, RF, A)$
    $\mathit{keepSearching} = (r' \neq r \wedge \textsc{ruleScorer}(r', LS) \neq 1.00)$
    $r = r'$
  end
return $(C, r)$
```

Figure 3.2: *The main procedure.*

signs a default slot to nodes that cannot be better classified by the previous components.

Definition 3.7 (Base learners and rule scorers) *The general literature on Machine Learning provides many learners that can be plugged into our proposal as long as they can handle multi-class problems and deal with both numeric and categoric attributive features [58]. The literature also provides a variety of rule scorers that can also be plugged into our proposal as long as they return a normalised score in range* $[0.00 \mathinner{..} 1.00]$*, where the lower bound indicates that a rule errs all the time and the upper bound indicates that it is a solution [69].*

Example 3.6 *Regarding the rule in Table §3.3(b), a rule scorer might score it at* 0.94 *when it is evaluated on the dataset in Table §3.3(a). Note that it is not generally possible to assess a score in isolation unless it is* 0.00 *or* 1.00*; that is, a score of* 0.94 *does not mean that the rule works well in* 94% *of the examples to which it is applied or something like that; it simply means that a rule that scores at, say,* 0.90 *is worse and a rule that scores at, say,* 0.98 *is better.*

```
method $\mathit{findExpansion}(C, B, LS, r, AF, RF, A)$
  – Step 1: initialise rule configuration.
  $C^* = C; B^* = B; LS^* = LS; r^* = r$
  $g^* = 0.00$
  – Step 2: explore candidate expansions.
  for each $(c, rf) \in C \times RF$ as long as $\textsc{ruleScorer}(r^*, LS^*) \neq 1.00$ do
    $c' = (\mathit{freshVar}(), rf, \text{target}\, c)$
    if $\neg\mathit{redundant}(c', C)$ then
      – Step 2.1: expand rule configuration.
      $C' = C \oplus \langle c' \rangle$
      $B' = B \cup \bigcup\{rf(n) \mid n \in B(\text{source}\, c')\}$
      $LS' = \mathit{expandLearningSet}(LS, c', B', AF, RF, A)$
      $r' = \textsc{baseLearner}(LS', A)$
      – Step 2.2: save the expanded rule configuration.
      $g' = \textsc{ruleScorer}(r', LS') - \textsc{ruleScorer}(r, LS)$
      if $g' > g^*$ then
        $C^* = C'; B^* = B'; LS^* = LS'; r^* = r'$
        $g^* = g'$
      end
    end
  end
return $(C^*, B^*, LS^*, r^*)$
```

Figure 3.3: *Procedure to find an expansion.*

3.2.2 Main procedures

Figure §3.2 presents the main procedure in ROLLER. It works on a set of documents D and an annotation A; it returns a tuple (C, r), where C is a context and r is a rule. It consists of three steps that run in sequence, namely:

- The first step consists in computing the attributive and the relational features of the nodes of which the input documents are composed. This is performed by a user-provided procedure called FEATUREBUILDER, which works on a set of documents D and returns a tuple (AF, RF), where AF denotes the set of attributive features and RF the set of relational features that it has computed. It is quite a simple procedure from a conceptual point of view: it loads the input documents, parses them into DOM trees, iterates over the resulting nodes, and computes the features that are provided in a catalogue. Note, however, that it is a little more involved from a technology point of view since it requires to interact with DOM-specific, browser-specific, and user-defined APIs to compute the HTML, the rendering, and the user-defined features, respectively. Such technology details are out of the scope of this chapter, in which our focus is on presenting the proposal, not on delving into the technology intricacies to implement it. ROLLER has to work on tuples of the following form very often: (C, B, LS, r), where AF is a set of attributive features, RF is a set of relational features, and A is an annotation; we refer to such tuples as input configurations for the sake of brevity.

- The second step consists in learning an initial rule building, exclusively, on the attributive features of the nodes in the input documents. To do so, we have to create an initial context of the form $C = \langle c \rangle$, where $c = (\mathtt{node}_0, \mathtt{null}, \mathtt{null})$. The corresponding binding B simply maps variable $\mathtt{node}_0$ onto the set of all nodes in the input documents, which can be very easily computed from the domain of the input annotation A. Then, a learning set LS is created; it maps every node bound in the initial context tuple c onto a vector that represents its attributive features. The base learner is finally invoked on learning set LS and the input annotation A in order to learn a rule r. ROLLER has to work on tuples of the following form very often: (C, B, LS, r), where C denotes a context, B its corresponding binding, LS is a learning set for context C, and r a rule that was learnt from that learning set. Thus, we refer to such tuples as rule configurations for the sake of brevity.

- The third step consists in finding an expansion, which is a term that we use to refer to a rule configuration that results from exploring some neighbours of the nodes in the context of the best rule configuration found so far. Ideally, such expansion should provide a rule that scores better than the current rule. To achieve such a goal, we combine the attributive features of the nodes in the current context with the attributive features of the nodes that are explored in the expansion. If an expansion that achieves a better score is found, this step is repeated again; otherwise, it stops and the procedure returns the best rule found and its associated context.

The procedure to find an expansion is presented in Figure §3.3. It works on the rule configuration (C, B, LS, r) that corresponds to the best rule found so far and an input configuration (AF, RF, A); it returns a rule configuration (C^*, B^*, LS^*, r^*) such that r^* improves or equals the score achieved by r. It consists of the following steps:

- The first step initialises a rule configuration of the form (C^*, B^*, LS^*, r^*) to the input rule configuration; the procedure searches for candidate expansions and stores the best one that it finds in this starred rule configuration. The criterion used to determine if an expanded configuration is better than another is based on the score achieved by the corresponding rule; we use variable g^* to save the score gain that is associated with the best expansion found so far. It is initialised to 0.00 because the first configuration coincides with the input configuration. Note that if an expansion that improves on the rule is not found, the procedure then returns the current rule r, which was stored in r^* during this initialisation step.

- The second step is a loop that explores candidate expansions. It iterates over the set of pairs (c, rf) of the Cartesian product of the context tuples in C and the relational features in RF, as long as the best expansion found is not a solution. For each such pair, a new context tuple of the form $(x, rf, \text{target}\, c)$ is created, where x denotes a new variable that is not used in context C; simply put, the new context tuple binds a new variable to the result of applying relational feature rf to the nodes that are currently bound to the target of context tuple c. This allows to explore the neighbourhood of every node in the current context C. Note, however, that only context tuples that are not redundant with regard to the current context must be explored. Such context tuples are then used to create a new rule configuration (C', B', LS', r'). The score gain of r' with respect to the input rule r is then computed; if it is greater than the score gain of the best expansion found so far, then it means that the new expansion must be saved since it has resulted in a better rule. The second step iterates until a rule that is a solution is found or the whole Cartesian product is explored; in both cases, the best rule configuration found is returned.

The check for redundancy is implemented by means of predicate $\mathtt{redundant}$, which given a context tuple (t, rf, s) and a context C holds as long as there is a context tuple (t', rf', s') in C such that $s = s'$ and $rf = rf'$ or $s' = t$ and $rf' = rf^{-1}$. The first condition is trivial since it amounts to saying that context tuples (t, rf, s) and (t', rf, s), where $t \neq t'$, are redundant because they bind the same nodes to different variables, which does not help explore new neighbours. The second condition is a little more involved; it amounts to saying that context tuples $(\mathsf{t}, \mathsf{rf}, \mathsf{s})$ and $(\mathsf{t}', \mathsf{rf}^{-1}, \mathsf{t})$, where $\mathsf{t} \neq \mathsf{t}'$, are redundant because the second one binds t' to the same nodes that are bound to s.

Example 3.7 *Assume that ROLLER is executed on the input documents and the annotation that are sketched in Figure §3.1 and Table §3.2, respectively. It first uses the user-provided* FEATUREBUILDER *procedure to compute the sets of attributive and relational features that are sketched in Table §3.1. These features and the annotation are then used to create an initial learning set that corresponds to context tuple* $(\mathsf{node}_0, \mathsf{null}, \mathsf{null})$*, which is sketched in Table §3.3(a). Note that the previous figure sketches a learning set that corresponds to two context tuples, namely, the initial context tuple* $(\mathsf{node}_0, \mathsf{null}, \mathsf{null})$ *and another context tuple that explores the parents of the nodes that are bound to* node_0*, that is,* $(\mathsf{node}_1, \mathsf{parent}, \mathsf{node}_0)$*. The initial learning set corresponds to the part of the figure that refers to the initial context tuple. If we apply a base learner to learn a rule from this learning set, then we might get the following rule:*

$$
\begin{array}{l}
\langle \\
\quad \mathsf{node}_0.\mathsf{tag} = \mathsf{span} \wedge \mathsf{node}_0.\mathsf{x\text{-}pos} \leq 0 \Rightarrow \mathsf{slot} = \mathsf{country}, \\
\quad \mathsf{node}_0.\mathsf{len} \geq 8 \Rightarrow \mathsf{slot} = \mathsf{null}, \\
\quad \mathsf{node}_0.\mathsf{y\text{-}pos} \geq 48 \Rightarrow \mathsf{slot} = \mathsf{null}, \\
\quad \mathsf{node}_0.\mathsf{tag} = \mathsf{span} \Rightarrow \mathsf{slot} = \mathsf{code}, \\
\quad \Rightarrow \mathsf{Slot} = \mathsf{Record} \\
\rangle,
\end{array}
$$

The previous rule scores at 0.90*. That means that it is quite a reasonably-good rule at classifying each node in the input documents into the appropriate slots, but it is not a solution because it still makes some mistakes. Thus, it makes sense to explore the neighbour nodes in order to find out if there is at least one of them whose attributive features can contribute to producing a better rule. Since the context currently has the initial context tuple* $(\mathsf{node}_0, \mathsf{null}, \mathsf{null})$ *only and the relational features are* parent, left, right*, and* child*, then the procedure to find an expansion has to explore the additional contexts that we show below:*

$$
\begin{array}{l}
\langle(\mathsf{node}_0, \mathsf{null}, \mathsf{null}), (\mathsf{node}_1, \mathsf{parent}, \mathsf{node}_0)\rangle, \\
\langle(\mathsf{node}_0, \mathsf{null}, \mathsf{null}), (\mathsf{node}_1, \mathsf{left}, \mathsf{node}_0)\rangle, \\
\langle(\mathsf{node}_0, \mathsf{null}, \mathsf{null}), (\mathsf{node}_1, \mathsf{right}, \mathsf{node}_0)\rangle, \textit{and} \\
\langle(\mathsf{node}_0, \mathsf{null}, \mathsf{null}), (\mathsf{node}_1, \mathsf{child}, \mathsf{node}_0)\rangle.
\end{array}
$$

Exploring the first context amounts to creating a new learning set in which the attributive features of each node are combined with the attributive features of their corresponding parents. In this case, the resulting learning set is sketched in Table §3.3(a). If the base learner is applied to this learning set, we might then get the following new rule:

$\langle$
$\quad \mathrm{node}_0.\mathrm{x\text{-}pos} \geq 100 \Rightarrow \mathrm{slot} = \mathrm{code},$
$\quad \mathrm{node}_0.\mathrm{tag} = \mathrm{span} \wedge \mathrm{node}_1.\mathrm{y\text{-}pos} \geq 16 \wedge \mathrm{node}_1.\mathrm{y\text{-}pos} \leq 32 \Rightarrow$
$\qquad \mathrm{slot} = \mathrm{country},$
$\quad \mathrm{node}_1.\mathrm{y\text{-}pos} \leq 0 \Rightarrow \mathrm{slot} = \mathrm{null},$
$\quad \mathrm{node}_1.\mathrm{y\text{-}pos} \geq 48 \Rightarrow \mathrm{slot} = \mathrm{null},$
$\quad \Rightarrow \mathrm{slot} = \mathsf{Record}$
$\rangle,$

which scores at 0.94. Exploring the remaining context tuples results in similar rules, none of which scores better. That means that we now have to explore the following contexts:

$\langle(\mathrm{node}_0, \mathrm{null}, \mathrm{null}), (\mathrm{node}_1, \mathrm{parent}, \mathrm{node}_0), (\mathrm{node}_2, \mathrm{left}, \mathrm{node}_0)\rangle,$
$\langle(\mathrm{node}_0, \mathrm{null}, \mathrm{null}), (\mathrm{node}_1, \mathrm{parent}, \mathrm{node}_0), (\mathrm{node}_2, \mathrm{right}, \mathrm{node}_0)\rangle,$
$\langle(\mathrm{node}_0, \mathrm{null}, \mathrm{null}), (\mathrm{node}_1, \mathrm{parent}, \mathrm{node}_0), (\mathrm{node}_2, \mathrm{child}, \mathrm{node}_0)\rangle,$
$\langle(\mathrm{node}_0, \mathrm{null}, \mathrm{null}), (\mathrm{node}_1, \mathrm{parent}, \mathrm{node}_0), (\mathrm{node}_2, \mathrm{parent}, \mathrm{node}_1)\rangle,$
$\langle(\mathrm{node}_0, \mathrm{null}, \mathrm{null}), (\mathrm{node}_1, \mathrm{parent}, \mathrm{node}_0), (\mathrm{node}_2, \mathrm{left}, \mathrm{node}_1)\rangle$, and
$\langle(\mathrm{node}_0, \mathrm{null}, \mathrm{null}), (\mathrm{node}_1, \mathrm{parent}, \mathrm{node}_0), (\mathrm{node}_2, \mathrm{right}, \mathrm{node}_1)\rangle.$

Note that there are two contexts that need not be explored, namely: context $\langle(\mathrm{node}_0, \mathrm{null}, \mathrm{null}), (\mathrm{node}_1, \mathrm{parent}, \mathrm{node}_0), (\mathrm{node}_2, \mathrm{parent}, \mathrm{node}_0)\rangle$ is not explored because it does not provide any additional data to the learning set and would result in the same rule; context $\langle(\mathrm{node}_0, \mathrm{null}, \mathrm{null}), (\mathrm{node}_1, \mathrm{parent}, \mathrm{node}_0), (\mathrm{node}_2, \mathrm{child}, \mathrm{node}_1)\rangle$ is not explored because context tuples $(\mathrm{node}_1, \mathrm{parent}, \mathrm{node}_0)$ and $(\mathrm{node}_2, \mathrm{child}, \mathrm{node}_1)$ are redundant because relational feature child is the inverse of relational feature parent, so exploring it would result again in the same rule. Note, however, that the new contexts are allowed to include nodes that have been explored previously; for instance, a context of the form $\langle(\mathrm{node}_0, \mathrm{null}, \mathrm{null}), (\mathrm{node}_1, \mathrm{parent}, \mathrm{node}_0), (\mathrm{node}_2, \mathrm{left}, \mathrm{node}_0)\rangle$ explores the left sibling of every node again, but in a different context since we explored them in isolation in the previous step and we now explore them in the context of their parent nodes.

In this case, the context that results in the best rule is $\langle(\mathrm{node}_0, \mathrm{null}, \mathrm{null}),$ $(\mathrm{node}_1, \mathrm{parent}, \mathrm{node}_0), (\mathrm{node}_2, \mathrm{parent}, \mathrm{node}_1)\rangle$, namely:

$\langle$

```
method $\mathsf{createLearningSet}(c, AF, RF, A)$
    $LS = \emptyset$
    $N = \mathrm{dom}\, A$
    for $n \in N$ do
        $v = \mathsf{computeVector}(n, AF, c)$
        $LS = LS \cup \{n \mapsto \{v\}\}$
    end
return $LS$
```

(a) Creating learning sets.

```
method $\mathsf{expandLearningSet}(LS, c, B, AF, RF, A)$
    $LS' = \emptyset$
    for $\{n \mapsto V\} \in LS$ do
        $V' = \mathsf{expandVectors}(n, V, c, B, AF, RF, A)$
        $LS' = LS' \cup \{n \mapsto V'\}$
    end
return $LS'$
```

(b) Expanding learning sets.

Figure 3.4: *Procedures to deal with learning sets*

$\mathsf{node}_0.\mathsf{tag} = \mathsf{span} \wedge \mathsf{node}_0.\mathsf{x\text{-}pos} \geq 100 \Rightarrow \mathsf{slot} = \mathsf{code}$,
$\mathsf{node}_0.\mathsf{tag} = \mathsf{span} \wedge \mathsf{node}_0.\mathsf{x\text{-}pos} \leq 0 \wedge \mathsf{node}_2.\mathsf{class} = \mathsf{list} \Rightarrow$
$\mathsf{slot} = \mathsf{country}$,
$\mathsf{node}_0.\mathsf{tag} = \mathsf{li} \wedge \mathsf{node}_0.\mathsf{y\text{-}pos} \leq 32 \Rightarrow \mathsf{slot} = \mathsf{Record}$,
$\Rightarrow \mathsf{slot} = \mathsf{null}$
$\rangle$.

This rule scores at 1.00*, which means that it is a solution, that is, it assigns every node in the learning set to the correct slot. So the search for a rule finishes here. Note that the resulting rule takes into account nodes* node_0 *and* node_2 *only;* node_1 *was used just to reach* node_2*, but its attributive features do not provide any classification power in this example. This is why our main procedure returns both a rule, which provides a classifier, and a context, which allows to bind the variables in the rule to the appropriate nodes.*

3.2.3 Working with learning sets

Learning sets associate nodes with the vectors that describe their attributive features within a given context. Figure §3.4 presents the two ancillary procedures that we need to work with them.

The first procedure is $\mathsf{createLearningSet}$, which works on an initial context tuple c and an input configuration (AF, RF, A); it returns a learning set in which every node in the domain of the annotation is mapped onto a singleton that provides its representation as a vector. Initially, every node is associated with a unique vector, but if the learning set is expanded using a relational feature that returns multiple values on the same node, that is, it

```
method $\mathrm{computeVector}(n, AF, c)$
  $v = \emptyset$
  for $af \in AF$ do
    $v = v \cup \{(c, af) \mapsto af(n)\}$
  end
return $v$
```

(a) Computing vectors.

```
method $\mathrm{expandVectors}(n, V, c, B, AF, RF, A)$
  $N = B(\text{target } c)$
  $rf = \text{relation } c$
  $V' = \emptyset$
  for $(m, v) \in N \times V$ such that $m \in rf(n)$ do
    $w = \mathrm{computeVector}(m, AF, c)$
    $V' = V' \cup \{v \cup w\}$
  end
return $V'$
```

(b) Expanding vectors.

Figure 3.5: *Procedures to deal with vectors*

relates a node with multiples nodes, then the initial vectors need to be combined with the vectors that correspond to several neighbours. This is the reason why learning sets associate nodes with sets of vectors.

The second procedure is $\mathrm{expandLearningSet}$, which works on a learning set LS, a context tuple c, a binding B, and an input configuration (AF, RF, A); it returns a learning set in which every node in LS is mapped onto a set of expanded vectors that represent the attributive features that are already present in learning set LS plus the attributive features that correspond to the nodes bound in B by context tuple c.

Example 3.8 *Table §3.3(a) illustrates a learning set that is created from the attributive features in Table §3.1(a). The initial learning sets consists of the vectors that correspond to context tuple* $(\mathrm{node}_0, \mathrm{null}, \mathrm{null})$*; the same figure illustrates how this learning set is expanded to take into account the features of the parents of every node.*

3.2.4 Working with vectors

Vectors represent the attributive features of a subset of nodes in the input documents in a format that is suitable to learn a rule using a propositional base learner. We need two ancillary procedures to deal with them, which are presented in Figure §3.5.

The first procedure is $\mathrm{computeVectors}$. It works on a node n, a set of attributive features AF, and a context tuple c. It computes a vector that is implemented as a map in which each attributive feature is associated with its corresponding value on node $\mathtt{n}$ regarding context $\mathtt{c}$. Note that this representation can be very straightforwardly translated into the table representation that typical machine-learning libraries use.

The second procedure is $\mathtt{expandVectors}$. It works on a node $\mathtt{n}$, a set of vectors V that is associated with $\mathtt{n}$ in a given learning set, a context tuple $\mathtt{c}$, a binding $\mathtt{B}$, and an input configuration $(\mathsf{AF}, \mathsf{RF}, \mathsf{A})$. It first computes the set of nodes N that correspond to the target of context tuple $\mathtt{c}$ and, after getting the relational feature in $\mathtt{c}$ and initialising the result V' to the empty set, it iterates over a set of pairs $(\mathtt{m}, v)$ in which $\mathtt{m}$ denotes a node in N and v is one of the vectors in V; note that only pairs in which $\mathtt{m} \in \mathtt{rf}(\mathtt{n})$ are considered, that is, pairs in which node $\mathtt{m}$ is a neighbour of node $\mathtt{n}$ regarding relational feature $\mathtt{rf}$. For every such pair, we first compute the vector w that corresponds to $\mathtt{m}$ using the set of attributive features AF and the context tuple $\mathtt{c}$; that vector is then merged with vector v, that is, vector v is expanded with the attributive features of node $\mathtt{m}$.

Example 3.9 *Let us examine node $\mathtt{n}_{10}$ in the document in Figure §3.1 and the initial context tuple $\mathtt{c} = (\mathtt{node}_0, \mathtt{null}, \mathtt{null})$. Recall that Table §3.1(a) reports on the attributive features of the nodes in our running example. The vector that is associated with this node is the following: $v = \{(\mathtt{c}, \mathtt{tag}) \mapsto \mathtt{span}, (\mathtt{c}, \mathtt{class}) \mapsto \mathtt{null}, (\mathtt{c}, \mathtt{y\text{-}pos}) \mapsto 16, (\mathtt{c}, \mathtt{x\text{-}pos}) \mapsto 0, (\mathtt{c}, \mathtt{len}) \mapsto 1, (\mathtt{c}, \mathtt{is\text{-}number}) \mapsto \mathtt{false}\}$. If this vector is expanded with context tuple $\mathtt{c}' = (\mathtt{node}_1, \mathtt{parent}, \mathtt{node}_0)$, then it becomes $v' = \{(\mathtt{c}, \mathtt{tag}) \mapsto \mathtt{span}, (\mathtt{c}, \mathtt{class}) \mapsto \mathtt{null}, (\mathtt{c}, \mathtt{y\text{-}pos}) \mapsto 16, (\mathtt{c}, \mathtt{x\text{-}pos}) \mapsto 0, (\mathtt{c}, \mathtt{len}) \mapsto 1, (\mathtt{c}, \mathtt{is\text{-}number}) \mapsto \mathtt{false}, (\mathtt{c}', \mathtt{tag}) \mapsto \mathtt{li}, (\mathtt{c}', \mathtt{class}) \mapsto \mathtt{null}, (\mathtt{c}', \mathtt{y\text{-}pos}) \mapsto 16, (\mathtt{c}', \mathtt{x\text{-}pos}) \mapsto 0, (\mathtt{c}', \mathtt{len}) \mapsto 2, (\mathtt{c}', \mathtt{is\text{-}number}) \mapsto \mathtt{false}\}$.*

3.3 Configuring our proposal

ROLLER has three variation points, namely: procedure FEATUREBUILDER, which computes a catalogue of features, BASERLEARNER, which learns a rule from a propositional learning set, and RULESCORER, which assesses how good a rule is. There are several alternatives to implement these procedures; the decision must, obviously, be made building on an experimental study that proves that the chosen combination of alternatives is very good.

In order to make a decision regarding which configuration is the most appropriate one, we setup them, run the resulting system on a collection of datasets, and computed the usual performance measures: precision (P), recall (R), and the F_1 score (F_1), as effectiveness measures, and learning time

(LT), and extraction time (ET), as efficiency measures. We then used a method to compute a rank for each of the resulting configurations and made a decision. For further details consult Appendices §A, and §B. During this customisation, we set the weight of F_1 score to 70%, the weight of LT to 10%, and the weight of ET to 20%. In other words, we think a good proposal must be able to learn rules that are very effective, that is, that achieve a high precision and recall, and consequently a high F_1 score. Note that learning a rule is a process that is executed every now and then, when a new site needs to be analysed or when a rule breaks because the corresponding site has undergone a change to its layout; since our experimental analysis confirms that ROLLER is quite effective and can learn in a matter of seconds, we did not think that the learning time could make a big difference between two alternatives. Contrarily, once a rule is learnt, it must be executed as quickly as possible in a production environment, so the extraction time is much more important than the learning time.

In the following subsections, we first report on the feature builder, then on the base learner, and the rule scorers that we examined; finally, we report on the results of the experimental analysis that we carried out to find the best combination of alternatives.

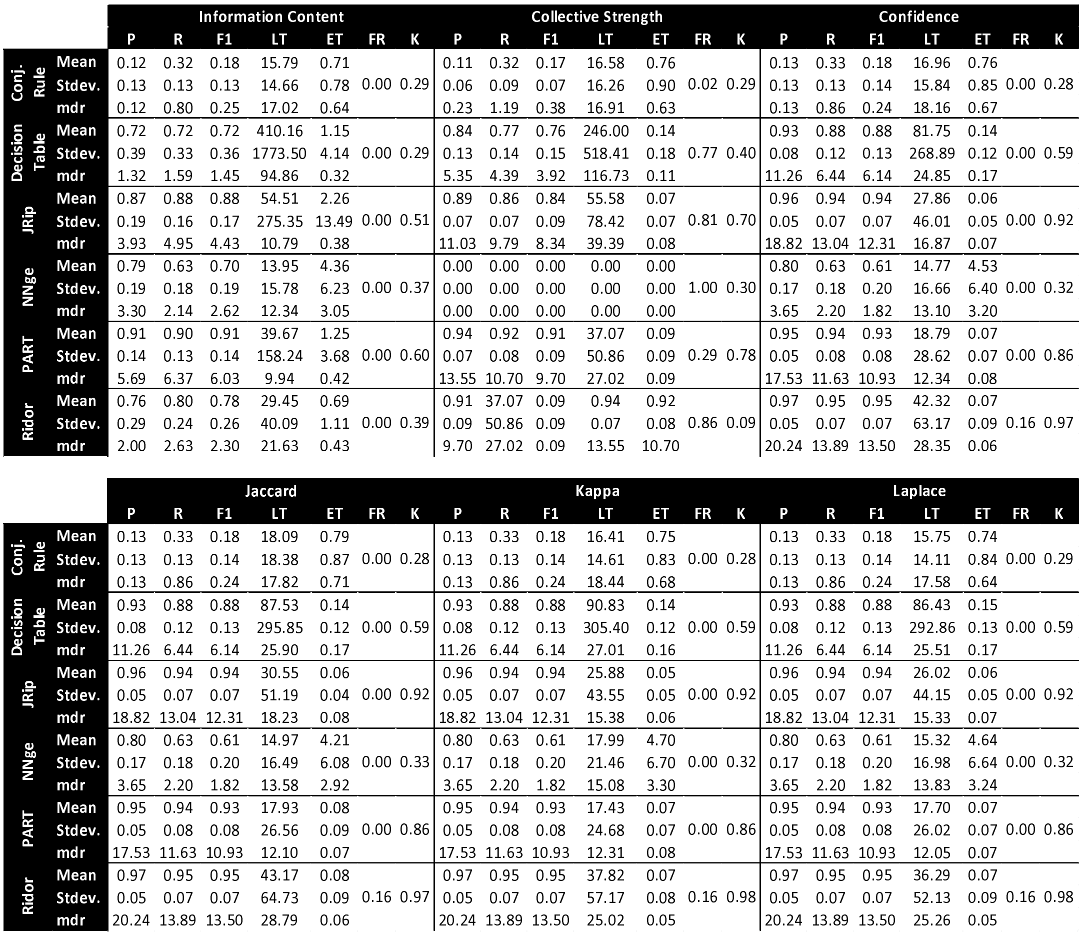
3.3.1 Our feature builder

Our feature builder computes the standard HTML features and the standard rendering features of the input documents, as they are defined in the corresponding W3C recommendations [80, 176]. Additionally, it computes some user-defined features, cf. Section §A.3.

Recall that ROLLER is not bound with a particular choice of features. This means that neither is bounded our feature builder. It computes the features defined in the open catalogue which in turn, are the features that a user thinks are the most appropriate for a given problem. The previous features are the features that we have selected for our experiments and they have proven to work very well in practice.

3.3.2 Our base learner and rule scorer

Regarding the base learner, we have explored Conjunctive Rule, Decision Table, JRip, NNge, PART, and Ridor [58]. They are available in Weka and can deal with multi-class problems and both numeric and categoric attributive features. A problem with them is that they do not work well with



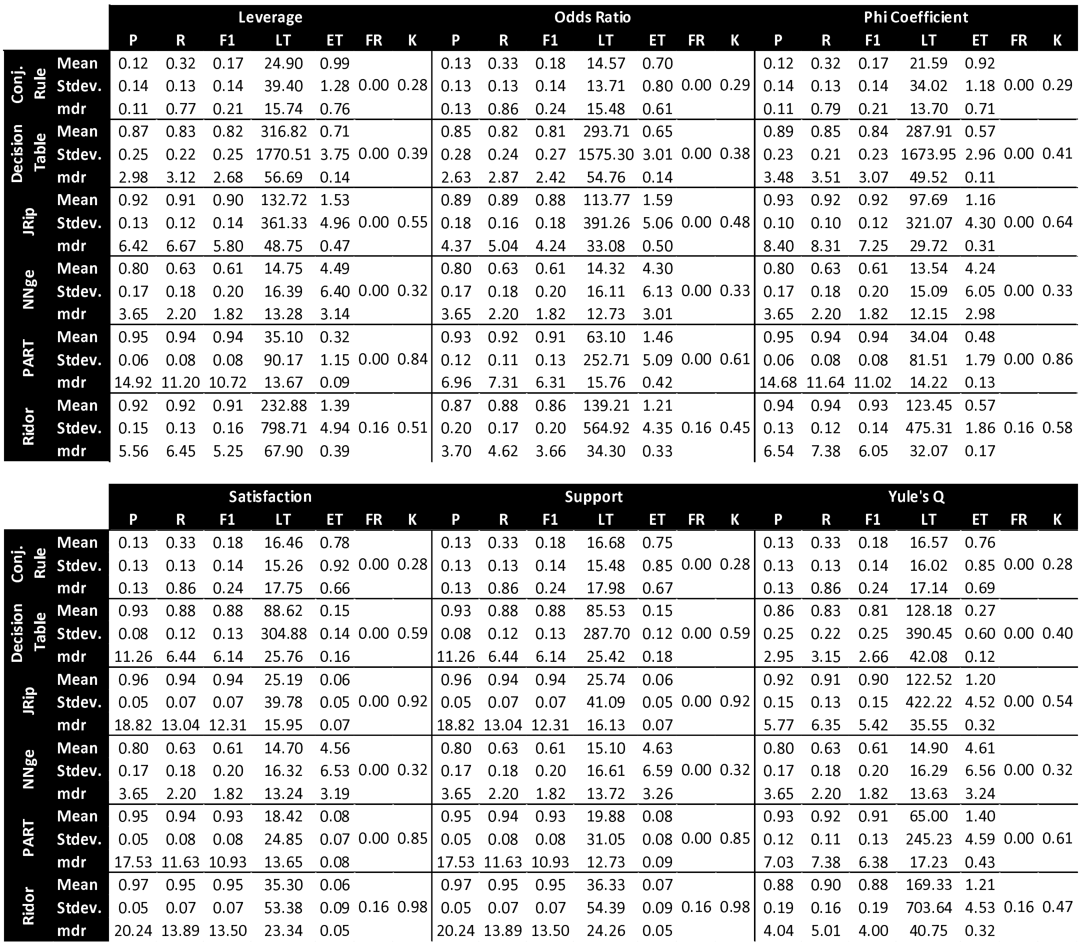
		Information Content							Collective Strength							Confidence						
		P	R	F1	LT	ET	FR	K	P	R	F1	LT	ET	FR	K	P	R	F1	LT	ET	FR	K
Conj. Rule	Mean	0.12	0.32	0.18	15.79	0.71			0.11	0.32	0.17	16.58	0.76			0.13	0.33	0.18	16.96	0.76		
	Stdev.	0.13	0.13	0.13	14.66	0.78	0.00	0.29	0.06	0.09	0.07	16.26	0.90	0.02	0.29	0.13	0.13	0.14	15.84	0.85	0.00	0.28
	mdr	0.12	0.80	0.25	17.02	0.64			0.23	1.19	0.38	16.91	0.63			0.13	0.86	0.24	18.16	0.67		
Decision Table	Mean	0.72	0.72	0.72	410.16	1.15			0.84	0.77	0.76	246.00	0.14			0.93	0.88	0.88	81.75	0.14		
	Stdev.	0.39	0.33	0.36	1773.50	4.14	0.00	0.29	0.13	0.14	0.15	518.41	0.18	0.77	0.40	0.08	0.12	0.13	268.89	0.12	0.00	0.59
	mdr	1.32	1.59	1.45	94.86	0.32			5.35	4.39	3.92	116.73	0.11			11.26	6.44	6.14	24.85	0.17		
JRip	Mean	0.87	0.88	0.88	54.51	2.26			0.89	0.86	0.84	55.58	0.07			0.96	0.94	0.94	27.86	0.06		
	Stdev.	0.19	0.16	0.17	275.35	13.49	0.00	0.51	0.07	0.07	0.09	78.42	0.07	0.81	0.70	0.05	0.07	0.07	46.01	0.05	0.00	0.92
	mdr	3.93	4.95	4.43	10.79	0.38			11.03	9.79	8.34	39.39	0.08			18.82	13.04	12.31	16.87	0.07		
NNge	Mean	0.79	0.63	0.70	13.95	4.36			0.00	0.00	0.00	0.00	0.00			0.80	0.63	0.61	14.77	4.53		
	Stdev.	0.19	0.18	0.19	15.78	6.23	0.00	0.37	0.00	0.00	0.00	0.00	0.00	1.00	0.30	0.17	0.18	0.20	16.66	6.40	0.00	0.32
	mdr	3.30	2.14	2.62	12.34	3.05			0.00	0.00	0.00	0.00	0.00			3.65	2.20	1.82	13.10	3.20		
PART	Mean	0.91	0.90	0.91	39.67	1.25			0.94	0.92	0.91	37.07	0.09			0.95	0.94	0.93	18.79	0.07		
	Stdev.	0.14	0.13	0.14	158.24	3.68	0.00	0.60	0.07	0.08	0.09	50.86	0.09	0.29	0.78	0.05	0.08	0.08	28.62	0.07	0.00	0.86
	mdr	5.69	6.37	6.03	9.94	0.42			13.55	10.70	9.70	27.02	0.09			17.53	11.63	10.93	12.34	0.08		
Ridor	Mean	0.76	0.80	0.78	29.45	0.69			0.91	37.07	0.09	0.94	0.92			0.97	0.95	0.95	42.32	0.07		
	Stdev.	0.29	0.24	0.26	40.09	1.11	0.00	0.39	0.09	50.86	0.09	0.07	0.08	0.86	0.09	0.05	0.07	0.07	63.17	0.09	0.16	0.97
	mdr	2.00	2.63	2.30	21.63	0.43			9.70	27.02	0.09	13.55	10.70			20.24	13.89	13.50	28.35	0.06		

		Jaccard							Kappa							Laplace						
		P	R	F1	LT	ET	FR	K	P	R	F1	LT	ET	FR	K	P	R	F1	LT	ET	FR	K
Conj. Rule	Mean	0.13	0.33	0.18	18.09	0.79			0.13	0.33	0.18	16.41	0.75			0.13	0.33	0.18	15.75	0.74		
	Stdev.	0.13	0.13	0.14	18.38	0.87	0.00	0.28	0.13	0.13	0.14	14.61	0.83	0.00	0.28	0.13	0.13	0.14	14.11	0.84	0.00	0.29
	mdr	0.13	0.86	0.24	17.82	0.71			0.13	0.86	0.24	18.44	0.68			0.13	0.86	0.24	17.58	0.64		
Decision Table	Mean	0.93	0.88	0.88	87.53	0.14			0.93	0.88	0.88	90.83	0.14			0.93	0.88	0.88	86.43	0.15		
	Stdev.	0.08	0.12	0.13	295.85	0.12	0.00	0.59	0.08	0.12	0.13	305.40	0.12	0.00	0.59	0.08	0.12	0.13	292.86	0.13	0.00	0.59
	mdr	11.26	6.44	6.14	25.90	0.17			11.26	6.44	6.14	27.01	0.16			11.26	6.44	6.14	25.51	0.17		
JRip	Mean	0.96	0.94	0.94	30.55	0.06			0.96	0.94	0.94	25.88	0.05			0.96	0.94	0.94	26.02	0.06		
	Stdev.	0.05	0.07	0.07	51.19	0.04	0.00	0.92	0.05	0.07	0.07	43.55	0.05	0.00	0.92	0.05	0.07	0.07	44.15	0.05	0.00	0.92
	mdr	18.82	13.04	12.31	18.23	0.08			18.82	13.04	12.31	15.38	0.06			18.82	13.04	12.31	15.33	0.07		
NNge	Mean	0.80	0.63	0.61	14.97	4.21			0.80	0.63	0.61	17.99	4.70			0.80	0.63	0.61	15.32	4.64		
	Stdev.	0.17	0.18	0.20	16.49	6.08	0.00	0.33	0.17	0.18	0.20	21.46	6.70	0.00	0.32	0.17	0.18	0.20	16.98	6.64	0.00	0.32
	mdr	3.65	2.20	1.82	13.58	2.92			3.65	2.20	1.82	15.08	3.30			3.65	2.20	1.82	13.83	3.24		
PART	Mean	0.95	0.94	0.93	17.93	0.08			0.95	0.94	0.93	17.43	0.07			0.95	0.94	0.93	17.70	0.07		
	Stdev.	0.05	0.08	0.08	26.56	0.09	0.00	0.86	0.05	0.08	0.08	24.68	0.07	0.00	0.86	0.05	0.08	0.08	26.02	0.07	0.00	0.86
	mdr	17.53	11.63	10.93	12.10	0.07			17.53	11.63	10.93	12.31	0.08			17.53	11.63	10.93	12.05	0.07		
Ridor	Mean	0.97	0.95	0.95	43.17	0.08			0.97	0.95	0.95	37.82	0.07			0.97	0.95	0.95	36.29	0.07		
	Stdev.	0.05	0.07	0.07	64.73	0.09	0.16	0.97	0.05	0.07	0.07	57.17	0.08	0.16	0.98	0.05	0.07	0.07	52.13	0.09	0.16	0.98
	mdr	20.24	13.89	13.50	28.79	0.06			20.24	13.89	13.50	25.02	0.05			20.24	13.89	13.50	25.26	0.05		

Table 3.4: *Experimental results regarding some configurations of ROLLER.*

learning sets that are unbalanced, which is the case in our context. The reason is that input documents are composed of hundreds of nodes, most of which are negative examples, cf. Table §A.1. Thus the base learner must balance the learning sets on which it works. We have explored several alternatives in the literature [12, 79], and our conclusion was that the one that best performs consists in computing the number of examples of the majority slot and then replicating as many examples of the other slots as needed to assemble a learning set that has approximately the same number of examples for each slot.

Regarding the rule scorer, Information Content is the most common in practice [139]. It has proven to guide the search process very well when deal-



		Leverage							Odds Ratio							Phi Coefficient						
		P	R	F1	LT	ET	FR	K	P	R	F1	LT	ET	FR	K	P	R	F1	LT	ET	FR	K
Conj. Rule	Mean	0.12	0.32	0.17	24.90	0.99	0.00	0.28	0.13	0.33	0.18	14.57	0.70	0.00	0.29	0.12	0.32	0.17	21.59	0.92	0.00	0.29
	Stdev.	0.14	0.13	0.14	39.40	1.28			0.13	0.13	0.14	13.71	0.80			0.14	0.13	0.14	34.02	1.18		
	mdr	0.11	0.77	0.21	15.74	0.76			0.13	0.86	0.24	15.48	0.61			0.11	0.79	0.21	13.70	0.71		
Decision Table	Mean	0.87	0.83	0.82	316.82	0.71	0.00	0.39	0.85	0.82	0.81	293.71	0.65	0.00	0.38	0.89	0.85	0.84	287.91	0.57	0.00	0.41
	Stdev.	0.25	0.22	0.25	1770.51	3.75			0.28	0.24	0.27	1575.30	3.01			0.23	0.21	0.23	1673.95	2.96		
	mdr	2.98	3.12	2.68	56.69	0.14			2.63	2.87	2.42	54.76	0.14			3.48	3.51	3.07	49.52	0.11		
JRip	Mean	0.92	0.91	0.90	132.72	1.53	0.00	0.55	0.89	0.89	0.88	113.77	1.59	0.00	0.48	0.93	0.92	0.92	97.69	1.16	0.00	0.64
	Stdev.	0.13	0.12	0.14	361.33	4.96			0.18	0.16	0.18	391.26	5.06			0.10	0.10	0.12	321.07	4.30		
	mdr	6.42	6.67	5.80	48.75	0.47			4.37	5.04	4.24	33.08	0.50			8.40	8.31	7.25	29.72	0.31		
NNge	Mean	0.80	0.63	0.61	14.75	4.49	0.00	0.32	0.80	0.63	0.61	14.32	4.30	0.00	0.33	0.80	0.63	0.61	13.54	4.24	0.00	0.33
	Stdev.	0.17	0.18	0.20	16.39	6.40			0.17	0.18	0.20	16.11	6.13			0.17	0.18	0.20	15.09	6.05		
	mdr	3.65	2.20	1.82	13.28	3.14			3.65	2.20	1.82	12.73	3.01			3.65	2.20	1.82	12.15	2.98		
PART	Mean	0.95	0.94	0.94	35.10	0.32	0.00	0.84	0.93	0.92	0.91	63.10	1.46	0.00	0.61	0.95	0.94	0.94	34.04	0.48	0.00	0.86
	Stdev.	0.06	0.08	0.08	90.17	1.15			0.12	0.11	0.13	252.71	5.09			0.06	0.08	0.08	81.51	1.79		
	mdr	14.92	11.20	10.72	13.67	0.09			6.96	7.31	6.31	15.76	0.42			14.68	11.64	11.02	14.22	0.13		
Ridor	Mean	0.92	0.92	0.91	232.88	1.39	0.16	0.51	0.87	0.88	0.86	139.21	1.21	0.16	0.45	0.94	0.94	0.93	123.45	0.57	0.16	0.58
	Stdev.	0.15	0.13	0.16	798.71	4.94			0.20	0.17	0.20	564.92	4.35			0.13	0.12	0.14	475.31	1.86		
	mdr	5.56	6.45	5.25	67.90	0.39			3.70	4.62	3.66	34.30	0.33			6.54	7.38	6.05	32.07	0.17		

		Satisfaction							Support							Yule's Q						
		P	R	F1	LT	ET	FR	K	P	R	F1	LT	ET	FR	K	P	R	F1	LT	ET	FR	K
Conj. Rule	Mean	0.13	0.33	0.18	16.46	0.78	0.00	0.28	0.13	0.33	0.18	16.68	0.75	0.00	0.28	0.13	0.33	0.18	16.57	0.76	0.00	0.28
	Stdev.	0.13	0.13	0.14	15.26	0.92			0.13	0.13	0.14	15.48	0.85			0.13	0.13	0.14	16.02	0.85		
	mdr	0.13	0.86	0.24	17.75	0.66			0.13	0.86	0.24	17.98	0.67			0.13	0.86	0.24	17.14	0.69		
Decision Table	Mean	0.93	0.88	0.88	88.62	0.15	0.00	0.59	0.93	0.88	0.88	85.53	0.15	0.00	0.59	0.86	0.83	0.81	128.18	0.27	0.00	0.40
	Stdev.	0.08	0.12	0.13	304.88	0.14			0.08	0.12	0.13	287.70	0.12			0.25	0.22	0.25	390.45	0.60		
	mdr	11.26	6.44	6.14	25.76	0.16			11.26	6.44	6.14	25.42	0.18			2.95	3.15	2.66	42.08	0.12		
JRip	Mean	0.96	0.94	0.94	25.19	0.06	0.00	0.92	0.96	0.94	0.94	25.74	0.06	0.00	0.92	0.92	0.91	0.90	122.52	1.20	0.00	0.54
	Stdev.	0.05	0.07	0.07	39.78	0.05			0.05	0.07	0.07	41.09	0.05			0.15	0.13	0.15	422.22	4.52		
	mdr	18.82	13.04	12.31	15.95	0.07			18.82	13.04	12.31	16.13	0.07			5.77	6.35	5.42	35.55	0.32		
NNge	Mean	0.80	0.63	0.61	14.70	4.56	0.00	0.32	0.80	0.63	0.61	15.10	4.63	0.00	0.32	0.80	0.63	0.61	14.90	4.61	0.00	0.32
	Stdev.	0.17	0.18	0.20	16.32	6.53			0.17	0.18	0.20	16.61	6.59			0.17	0.18	0.20	16.29	6.56		
	mdr	3.65	2.20	1.82	13.24	3.19			3.65	2.20	1.82	13.72	3.26			3.65	2.20	1.82	13.63	3.24		
PART	Mean	0.95	0.94	0.93	18.42	0.08	0.00	0.85	0.95	0.94	0.93	19.88	0.08	0.00	0.85	0.93	0.92	0.91	65.00	1.40	0.00	0.61
	Stdev.	0.05	0.08	0.08	24.85	0.07			0.05	0.08	0.08	31.05	0.08			0.12	0.11	0.13	245.23	4.59		
	mdr	17.53	11.63	10.93	13.65	0.08			17.53	11.63	10.93	12.73	0.09			7.03	7.38	6.38	17.23	0.43		
Ridor	Mean	0.97	0.95	0.95	35.30	0.06	0.16	0.98	0.97	0.95	0.95	36.33	0.07	0.16	0.98	0.88	0.90	0.88	169.33	1.21	0.16	0.47
	Stdev.	0.05	0.07	0.07	53.38	0.09			0.05	0.07	0.07	54.39	0.09			0.19	0.16	0.19	703.64	4.53		
	mdr	20.24	13.89	13.50	23.34	0.05			20.24	13.89	13.50	24.26	0.05			4.04	5.01	4.00	40.75	0.32		

Table 3.4: *Experimental results regarding some configurations of ROLLER. (Cont'd)*

ing with classical inductive logic programming problems. It relies exclusively on the number of true positives and false positives that a rule produces when it is evaluated. We wished to explore some rule scorers that also take into account the number of true negatives and false negatives. We have surveyed the literature and we have found several alternatives [69], namely: Collective Strength, Confidence, Jaccard, Kappa, Laplace, Leverage, Odds Ratio, Phi Coefficient, Satisfaction, Support, and Yule's Q.

The cartesian product of base learners and rule scorers resulted in 72 variations of ROLLER. Table §3.4 summarises the results that we obtained when we run each variation on our datasets, including the mean and standard deviations of precision (P), recall (R), the F_1 score (F_1), learning time (LT), and extraction time (ET), as well as our rank (K) and the failure ratio (FR).

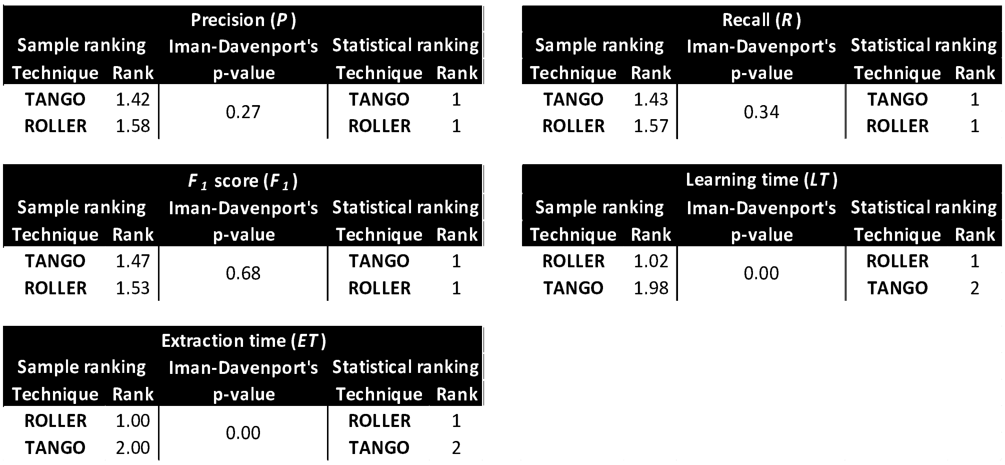
In our experimentation, we set the weight of F_1 score to 70%, the weight of LT to 10%, and the weight of ET to 20%. In other words, we think that a good proposal must be able to learn extraction rules that are very effective, that is, that achieve a high precision and recall, and, consequently, a high F_1 score. Note that learning a rule is a process that is executed every now and then, when a new site needs to be analysed or when a rule breaks because the corresponding site has undergone a change to its layout; since our experimental analysis confirmed that ROLLER is quite effective and can learn in a matter of seconds, we did not think that the learning time could make a big difference between two alternatives. Contrarily, once a rule is learnt, it must be executed as quickly as possible in a production environment, so the extraction time is much more important than the learning time.

Note that the best variations achieve $K = 0.98$ and $K = 0.97$; they all rely on Ridor as the base learner and Jaccard, Laplace, Satisfaction, or Support as the rule scorers; unfortunately, all of them have a failure ratio of 0.16, which means that they cannot deal with some datasets. The problem is that Ridor is a learner that uses a technique called Reduced Error Pruning to prune the resulting rules; unfortunately, there are a number of datasets that do not provide enough data for this technique to work, which means that it cannot be applied to relatively small documents. As a conclusion, we have to resign to use Ridor, even though it works well with sufficiently large documents.

Thus, the best variations seem to be those that achieve $K = 0.92$ with a 0.00 failure ratio. They all correspond to using JRip as the base learner and Confidence, Jaccard, Kappa, Laplace, Satisfaction, and Support as rule scorers. Since there are multiples ties, we decided to select JRip and Kappa because this is the variation that achieves the minimum extraction time.

3.4 Experimental analysis

In this section, we first report on a comparison in which we prove that ROLLER is as effective as TANGO but far more efficient. We then compare ROLLER to other state-of-the-art proposals regarding effectiveness and efficiency. Consult Appendices §A and §B regarding our experimental environment, which includes a description of the hardware and the software used, the experimental datasets, the catalogue of features, and the proposals with which we have compared ours, cf. Appendix §A. Consult Appendix §B regarding the performance measures and the statistical method used to analyse the results.



Precision (*P*)

Sample ranking		Iman-Davenport's	Statistical ranking	
Technique	Rank	p-value	Technique	Rank
TANGO	1.42	0.27	TANGO	1
ROLLER	1.58		ROLLER	1

Recall (*R*)

Sample ranking		Iman-Davenport's	Statistical ranking	
Technique	Rank	p-value	Technique	Rank
TANGO	1.43	0.34	TANGO	1
ROLLER	1.57		ROLLER	1

F_1 score (F_1)

Sample ranking		Iman-Davenport's	Statistical ranking	
Technique	Rank	p-value	Technique	Rank
TANGO	1.47	0.68	TANGO	1
ROLLER	1.53		ROLLER	1

Learning time (*LT*)

Sample ranking		Iman-Davenport's	Statistical ranking	
Technique	Rank	p-value	Technique	Rank
ROLLER	1.02	0.00	ROLLER	1
TANGO	1.98		TANGO	2

Extraction time (*ET*)

Sample ranking		Iman-Davenport's	Statistical ranking	
Technique	Rank	p-value	Technique	Rank
ROLLER	1.00	0.00	ROLLER	1
TANGO	2.00		TANGO	2

Table 3.5: *Summary of results regarding ROLLER and TANGO.*

3.4.1 Comparison with TANGO

Recall that our motivation to work on ROLLER was to produce a system that should be as effective as TANGO, which has proven to be very good at learning information extraction rules with high recall and precision, but much more efficient. Efficiency is the only problem that we can actually put down to TANGO.

Tables §2.30, §2.34, §3.6, and §3.10 report on our effectiveness and efficiency results regarding TANGO and ROLLER. TANGO achieves a precision of 0.96 ± 0.08 and Roller 0.96 ± 0.05, which means that they are very good at making a difference amongst the information to be extracted and the information to be ignored, but ROLLER is slightly more stable. Regarding recall, TANGO achieves 0.96 ± 0.05 and ROLLER seems to be a a bit worse since it achieves a value of 0.94 ± 0.07; that is, ROLLER is slightly more unstable than TANGO regarding its ability to find the information to be extracted. The figures regarding the F_1 score are very similar, too. It is regarding efficiency that the differences are very clear. Note that TANGO takes 978.19 ± 2 377.61 CPU seconds to learn rules, whereas ROLLER takes 25.06 ± 43.55 CPU seconds in average, which we think is quite a significant improvement. Regarding the extraction time the difference is also very remarkable: TANGO's rules take 221.68 ± 289.77 CPU seconds in average, whereas ROLLER's rules take only 0.05 ± 0.05 seconds in average.

Before drawing a conclusion, we conducted the Iman-Davenport's statistical test on our experimental data. Table §3.5 summarises our results. Note that the p-value that the test returns is clearly above the standard significance level $\alpha = 0.05$ in the case of precision, recall, and the F_1 score, which is a clear indication that the differences in the empirical data are not significant; that is, our experiments do not provide enough evidence to conclude that TANGO and ROLLER behave differently regarding their effectiveness. Contrarily, the p-value regarding learning time and extraction time is 0.00 in both cases, which is quite a strong indication that they behave very differently.

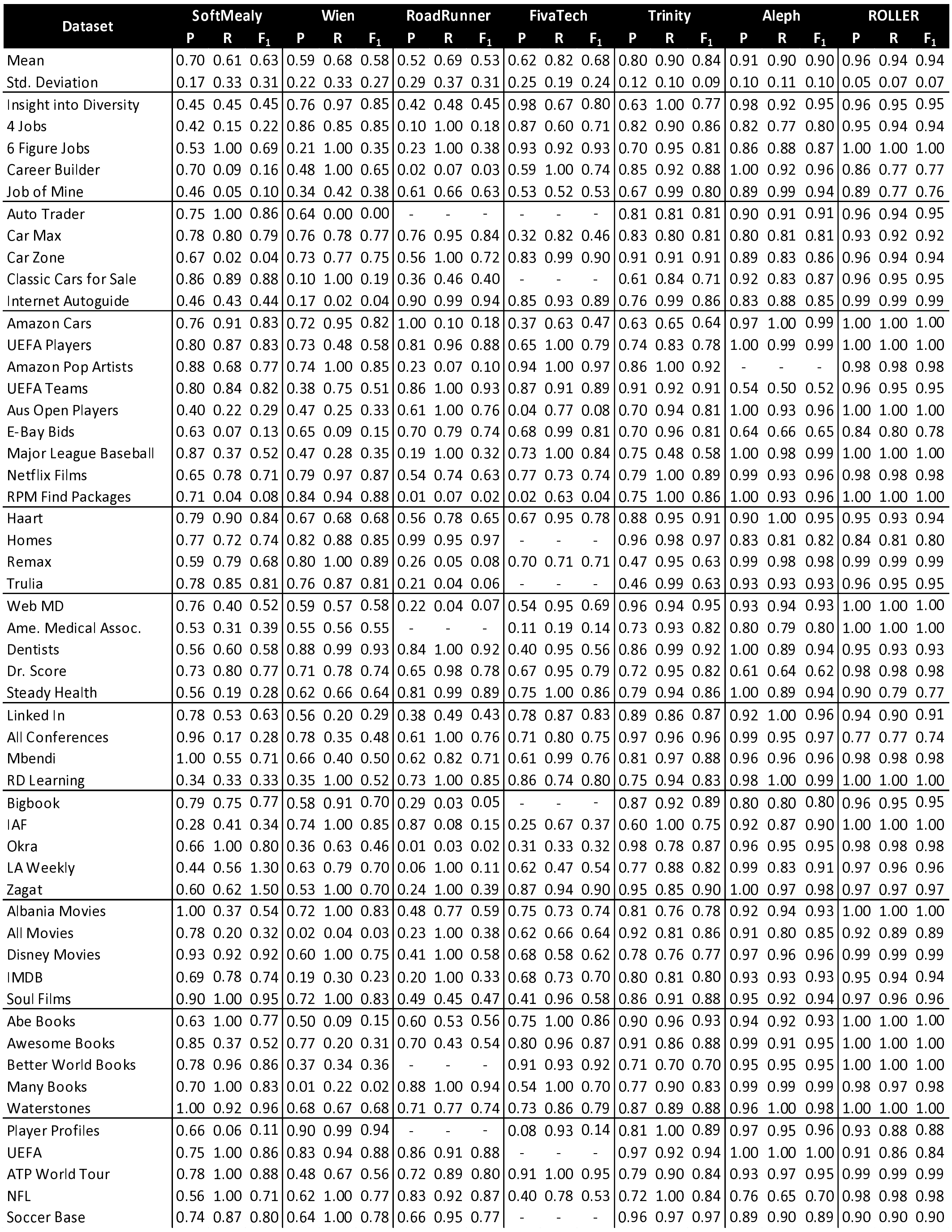
As a conclusion, we have achieved our goal since ROLLER is as effective as TANGO, but much more efficient. What remains to study is wether it can beat other state-of-the-art proposals.

3.4.2 Effectiveness analysis

Table §3.6 reports on the raw effectiveness data that we got from our experimentation. For each proposal, we report on its effectiveness measures regarding our datasets. The first two lines also provide a summary of the results in terms of the mean value and standard deviation. Since it is difficult to spot a trend in this table, we decided to summarise the data using boxplots.

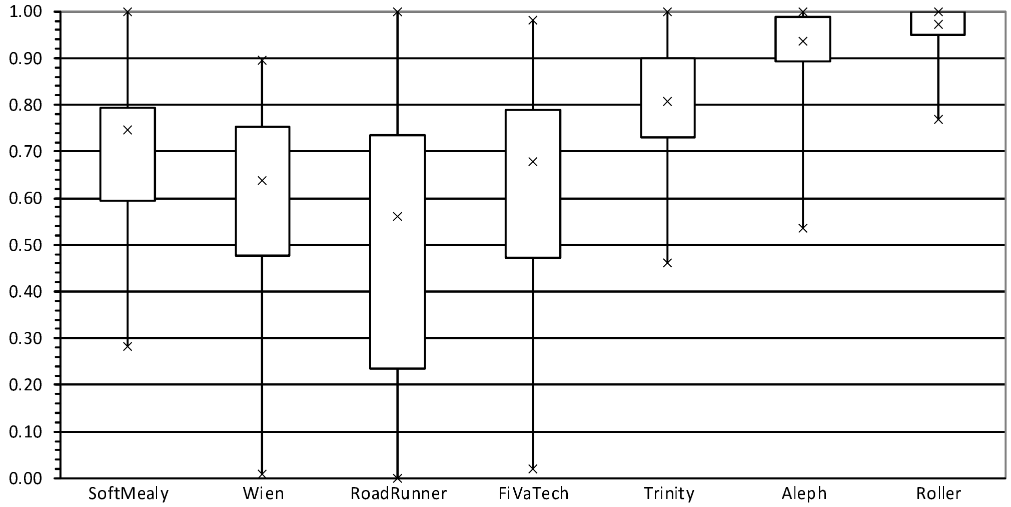
Table §3.7 summarises the results regarding precision. Empirically, ROLLER seems to be the proposal that can achieve a better precision, and it is, indeed, the one that is more stable regarding this effectiveness measure; the other proposals can achieve precisions that are as high as ROLLER's for some datasets, but their deviation with respect to the mean is larger. Iman-Davenport's test returns a p-value that is nearly zero, which is a strong indication that there are differences in rank amongst the proposals that we have compared. We then have to compare ROLLER, which ranks the first regarding precision, and the other techniques. Hommel's test confirms that the differences in rank amongst ROLLER and the other techniques are statistically significant because it returns adjusted p-values that are very small with regard to the significance level. In other words, our experimental data provide enough evidence to reject the hypothesis that ROLLER behaves similarly to the other proposals regarding precision, that is, it supports the idea that ROLLER can learn rules that are more precise than the other proposals.

Table §3.8 summarises the results regarding recall. Empirically, ROLLER seems to be the proposal that can achieve a higher recall and it is the one that seems more stable regarding this measure because its deviation is the smallest and its inter-quartile range is also the smallest. Note, however, that the



Dataset	SoftMealy			Wien			RoadRunner			FivaTech			Trinity			Aleph			ROLLER		
	P	R	F_1	P	R	F_1	P	R	F_1	P	R	F_1	P	R	F_1	P	R	F_1	P	R	F_1
Mean	0.70	0.61	0.63	0.59	0.68	0.58	0.52	0.69	0.53	0.62	0.82	0.68	0.80	0.90	0.84	0.91	0.90	0.90	0.96	0.94	0.94
Std. Deviation	0.17	0.33	0.31	0.22	0.33	0.27	0.29	0.37	0.31	0.25	0.19	0.24	0.12	0.10	0.09	0.10	0.11	0.10	0.05	0.07	0.07
Insight into Diversity	0.45	0.45	0.45	0.76	0.97	0.85	0.42	0.48	0.45	0.98	0.67	0.80	0.63	1.00	0.77	0.98	0.92	0.95	0.96	0.95	0.95
4 Jobs	0.42	0.15	0.22	0.86	0.85	0.85	0.10	1.00	0.18	0.87	0.60	0.71	0.82	0.90	0.86	0.82	0.77	0.80	0.95	0.94	0.94
6 Figure Jobs	0.53	1.00	0.69	0.21	1.00	0.35	0.23	1.00	0.38	0.93	0.92	0.93	0.70	0.95	0.81	0.86	0.88	0.87	1.00	1.00	1.00
Career Builder	0.70	0.09	0.16	0.48	1.00	0.65	0.02	0.07	0.03	0.59	1.00	0.74	0.85	0.92	0.88	1.00	0.92	0.96	0.86	0.77	0.77
Job of Mine	0.46	0.05	0.10	0.34	0.42	0.38	0.61	0.66	0.63	0.53	0.52	0.53	0.67	0.99	0.80	0.89	0.99	0.94	0.89	0.77	0.76
Auto Trader	0.75	1.00	0.86	0.64	0.00	0.00	-	-	-	-	-	-	0.81	0.81	0.81	0.90	0.91	0.91	0.96	0.94	0.95
Car Max	0.78	0.80	0.79	0.76	0.78	0.77	0.76	0.95	0.84	0.32	0.82	0.46	0.83	0.80	0.81	0.80	0.81	0.81	0.93	0.92	0.92
Car Zone	0.67	0.02	0.04	0.73	0.77	0.75	0.56	1.00	0.72	0.83	0.99	0.90	0.91	0.91	0.91	0.89	0.83	0.86	0.96	0.94	0.94
Classic Cars for Sale	0.86	0.89	0.88	0.10	1.00	0.19	0.36	0.46	0.40	-	-	-	0.61	0.84	0.71	0.92	0.83	0.87	0.96	0.95	0.95
Internet Autoguide	0.46	0.43	0.44	0.17	0.02	0.04	0.90	0.99	0.94	0.85	0.93	0.89	0.76	0.99	0.86	0.83	0.88	0.85	0.99	0.99	0.99
Amazon Cars	0.76	0.91	0.83	0.72	0.95	0.82	1.00	0.10	0.18	0.37	0.63	0.47	0.63	0.65	0.64	0.97	1.00	0.99	1.00	1.00	1.00
UEFA Players	0.80	0.87	0.83	0.73	0.48	0.58	0.81	0.96	0.88	0.65	1.00	0.79	0.74	0.83	0.78	1.00	0.99	0.99	1.00	1.00	1.00
Amazon Pop Artists	0.88	0.68	0.77	0.74	1.00	0.85	0.23	0.07	0.10	0.94	1.00	0.97	0.86	1.00	0.92	-	-	-	0.98	0.98	0.98
UEFA Teams	0.80	0.84	0.82	0.38	0.75	0.51	0.86	1.00	0.93	0.87	0.91	0.89	0.91	0.92	0.91	0.54	0.50	0.52	0.96	0.95	0.95
Aus Open Players	0.40	0.22	0.29	0.47	0.25	0.33	0.61	1.00	0.76	0.04	0.77	0.08	0.70	0.94	0.81	1.00	0.93	0.96	1.00	1.00	1.00
E-Bay Bids	0.63	0.07	0.13	0.65	0.09	0.15	0.70	0.79	0.74	0.68	0.99	0.81	0.70	0.96	0.81	0.64	0.66	0.65	0.84	0.80	0.78
Major League Baseball	0.87	0.37	0.52	0.47	0.28	0.35	0.19	1.00	0.32	0.73	1.00	0.84	0.75	0.48	0.58	1.00	0.98	0.99	1.00	1.00	1.00
Netflix Films	0.65	0.78	0.71	0.79	0.97	0.87	0.54	0.74	0.63	0.77	0.73	0.74	0.79	1.00	0.89	0.99	0.93	0.96	0.98	0.98	0.98
RPM Find Packages	0.71	0.04	0.08	0.84	0.94	0.88	0.01	0.07	0.02	0.02	0.63	0.04	0.75	1.00	0.86	1.00	0.93	0.96	1.00	1.00	1.00
Haart	0.79	0.90	0.84	0.67	0.68	0.68	0.56	0.78	0.65	0.67	0.95	0.78	0.88	0.95	0.91	0.90	1.00	0.95	0.95	0.93	0.94
Homes	0.77	0.72	0.74	0.82	0.88	0.85	0.99	0.95	0.97	-	-	-	0.96	0.98	0.97	0.83	0.81	0.82	0.84	0.81	0.80
Remax	0.59	0.79	0.68	0.80	1.00	0.89	0.26	0.05	0.08	0.70	0.71	0.71	0.47	0.95	0.63	0.99	0.98	0.98	0.99	0.99	0.99
Trulia	0.78	0.85	0.81	0.76	0.87	0.81	0.21	0.04	0.06	-	-	-	0.46	0.99	0.63	0.93	0.93	0.93	0.96	0.95	0.95
Web MD	0.76	0.40	0.52	0.59	0.57	0.58	0.22	0.04	0.07	0.54	0.95	0.69	0.96	0.94	0.95	0.93	0.94	0.93	1.00	1.00	1.00
Ame. Medical Assoc.	0.53	0.31	0.39	0.55	0.56	0.55	-	-	-	0.11	0.19	0.14	0.73	0.93	0.82	0.80	0.79	0.80	1.00	1.00	1.00
Dentists	0.56	0.60	0.58	0.88	0.99	0.93	0.84	1.00	0.92	0.40	0.95	0.56	0.86	0.99	0.92	1.00	0.89	0.94	0.95	0.93	0.93
Dr. Score	0.73	0.80	0.77	0.71	0.78	0.74	0.65	0.98	0.78	0.67	0.95	0.79	0.72	0.95	0.82	0.61	0.64	0.62	0.98	0.98	0.98
Steady Health	0.56	0.19	0.28	0.62	0.66	0.64	0.81	0.99	0.89	0.75	1.00	0.86	0.79	0.94	0.86	1.00	0.89	0.94	0.90	0.79	0.77
Linked In	0.78	0.53	0.63	0.56	0.20	0.29	0.38	0.49	0.43	0.78	0.87	0.83	0.89	0.86	0.87	0.92	1.00	0.96	0.94	0.90	0.91
All Conferences	0.96	0.17	0.28	0.78	0.35	0.48	0.61	1.00	0.76	0.71	0.80	0.75	0.97	0.96	0.96	0.99	0.95	0.97	0.77	0.77	0.74
Mbendi	1.00	0.55	0.71	0.66	0.40	0.50	0.62	0.82	0.71	0.61	0.99	0.76	0.81	0.97	0.88	0.96	0.96	0.96	0.98	0.98	0.98
RD Learning	0.34	0.33	0.33	0.35	1.00	0.52	0.73	1.00	0.85	0.86	0.74	0.80	0.75	0.94	0.83	0.98	1.00	0.99	1.00	1.00	1.00
Bigbook	0.79	0.75	0.77	0.58	0.91	0.70	0.29	0.03	0.05	-	-	-	0.87	0.92	0.89	0.80	0.80	0.80	0.96	0.95	0.95
IAF	0.28	0.41	0.34	0.74	1.00	0.85	0.87	0.08	0.15	0.25	0.67	0.37	0.60	1.00	0.75	0.92	0.87	0.90	1.00	1.00	1.00
Okra	0.66	1.00	0.80	0.36	0.63	0.46	0.01	0.03	0.02	0.31	0.33	0.32	0.98	0.78	0.87	0.96	0.95	0.95	0.98	0.98	0.98
LA Weekly	0.44	0.56	1.30	0.63	0.79	0.70	0.06	1.00	0.11	0.62	0.47	0.54	0.77	0.88	0.82	0.99	0.83	0.91	0.97	0.96	0.96
Zagat	0.60	0.62	1.50	0.53	1.00	0.70	0.24	1.00	0.39	0.87	0.94	0.90	0.95	0.85	0.90	1.00	0.97	0.98	0.97	0.97	0.97
Albania Movies	1.00	0.37	0.54	0.72	1.00	0.83	0.48	0.77	0.59	0.75	0.73	0.74	0.81	0.76	0.78	0.92	0.94	0.93	1.00	1.00	1.00
All Movies	0.78	0.20	0.32	0.02	0.04	0.03	0.23	1.00	0.38	0.62	0.66	0.64	0.92	0.81	0.86	0.91	0.80	0.85	0.92	0.89	0.89
Disney Movies	0.93	0.92	0.92	0.60	1.00	0.75	0.41	1.00	0.58	0.68	0.58	0.62	0.78	0.76	0.77	0.97	0.96	0.96	0.99	0.99	0.99
IMDB	0.69	0.78	0.74	0.19	0.30	0.23	0.20	1.00	0.33	0.68	0.73	0.70	0.80	0.81	0.80	0.93	0.93	0.93	0.95	0.94	0.94
Soul Films	0.90	1.00	0.95	0.72	1.00	0.83	0.49	0.45	0.47	0.41	0.96	0.58	0.86	0.91	0.88	0.95	0.92	0.94	0.97	0.96	0.96
Abe Books	0.63	1.00	0.77	0.50	0.09	0.15	0.60	0.53	0.56	0.75	1.00	0.86	0.90	0.96	0.93	0.94	0.92	0.93	1.00	1.00	1.00
Awesome Books	0.85	0.37	0.52	0.77	0.20	0.31	0.70	0.43	0.54	0.80	0.96	0.87	0.91	0.86	0.88	0.99	0.91	0.95	1.00	1.00	1.00
Better World Books	0.78	0.96	0.86	0.37	0.34	0.36	-	-	-	0.91	0.93	0.92	0.71	0.70	0.70	0.95	0.95	0.95	1.00	1.00	1.00
Many Books	0.70	1.00	0.83	0.01	0.22	0.02	0.88	1.00	0.94	0.54	1.00	0.70	0.77	0.90	0.83	0.99	0.99	0.99	0.98	0.97	0.98
Waterstones	1.00	0.92	0.96	0.68	0.67	0.68	0.71	0.77	0.74	0.73	0.86	0.79	0.87	0.89	0.88	0.96	1.00	0.98	1.00	1.00	1.00
Player Profiles	0.66	0.06	0.11	0.90	0.99	0.94	-	-	-	0.08	0.93	0.14	0.81	1.00	0.89	0.97	0.95	0.96	0.93	0.88	0.88
UEFA	0.75	1.00	0.86	0.83	0.94	0.88	0.86	0.91	0.88	-	-	-	0.97	0.92	0.94	1.00	1.00	1.00	0.91	0.86	0.84
ATP World Tour	0.78	1.00	0.88	0.48	0.67	0.56	0.72	0.89	0.80	0.91	1.00	0.95	0.79	0.90	0.84	0.93	0.97	0.95	0.99	0.99	0.99
NFL	0.56	1.00	0.71	0.62	1.00	0.77	0.83	0.92	0.87	0.40	0.78	0.53	0.72	1.00	0.84	0.76	0.65	0.70	0.98	0.98	0.98
Soccer Base	0.74	0.87	0.80	0.64	1.00	0.78	0.66	0.95	0.77	-	-	-	0.96	0.97	0.97	0.89	0.90	0.89	0.90	0.90	0.90

Table 3.6: *Effectiveness results.*

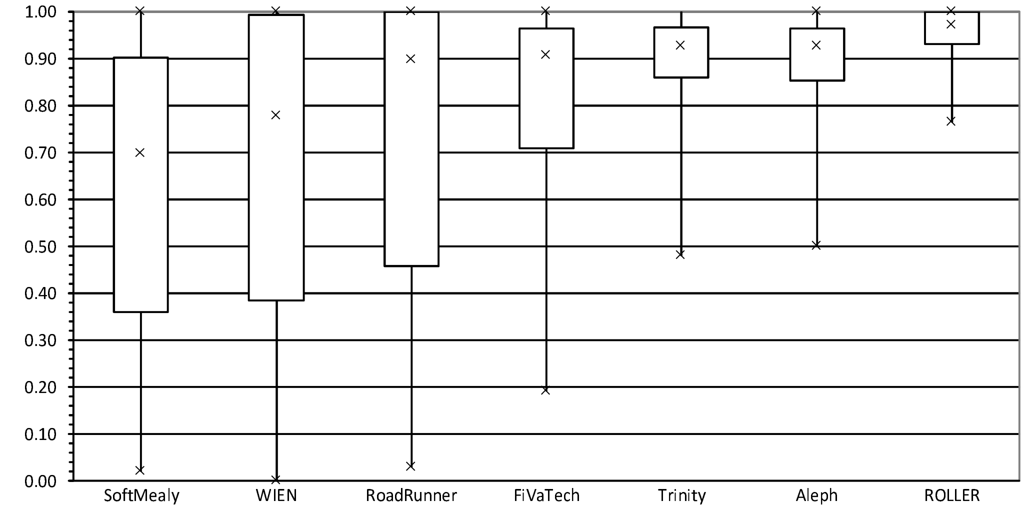


	SoftMealy	Wien	RoadRunner	FiVaTech	Trinity	Aleph	Roller
Quartile 1	0.59	0.48	0.23	0.47	0.73	0.89	0.95
Minimum	0.28	0.01	0.00	0.02	0.46	0.54	0.77
Median	0.75	0.64	0.56	0.68	0.81	0.94	0.97
Maximum	1.00	0.90	1.00	0.98	1.00	1.00	1.00
Quartile 3	0.79	0.75	0.73	0.79	0.90	0.99	1.00

Sample ranking		Iman-Davenport's	Hommel's							Statistical ranking	
Technique	Rank	p-value	adjusted p-values							Technique	Rank
Roller	1.50	8.88E-16		Aleph	Trinity	SoftMealy	FiVaTech	Wien	RoadRunner	Roller	1
Aleph	2.57		Roller	1.54E-03	1.17E-09	4.37E-09	1.71E-13	1.55E-15	9.18E-09	Aleph	2
Trinity	3.36									Trinity	2
SoftMealy	4.50									SoftMealy	2
FiVaTech	5.22									FiVaTech	2
Wien	5.26									Wien	2
RoadRunner	5.58									RoadRunner	2

Table 3.7: *Summary of results regarding precision.*

other techniques can achieve results that are very good, too, chiefly Trinity and Aleph. Iman-Davenport's test returns a p-value that is very close to zero, which is a clear indication that there are differences in rank amongst the proposals that we have compared. Hommel's test confirms that the differences in rank amongst ROLLER, which ranks the first from an empirical point of view, Aleph, RoadRunner, FiVaTech, Wien, and SoftMealy are statistically significant at the standard significance level; note, however, that the adjusted p-value that corresponds to the comparison between ROLLER and Trinity is not greater than the standard significance level, which means that the difference in empirical rank between these two proposals is not statistically significant. As a conclusion, the experimental data do not provide enough evidence to reject the hypothesis that ROLLER and Trinity be-



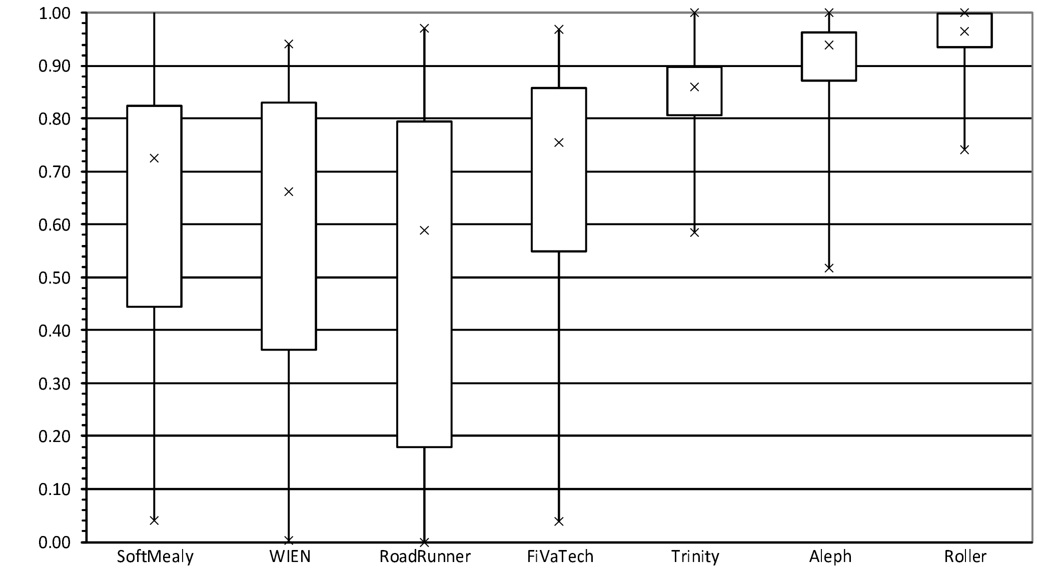
	SoftMealy	WIEN	RoadRunner	FiVaTech	Trinity	Aleph	ROLLER
Quartile 1	0.36	0.38	0.46	0.71	0.86	0.85	0.93
Minimum	0.02	0.00	0.03	0.19	0.48	0.50	0.77
Median	0.70	0.78	0.90	0.91	0.93	0.93	0.97
Maximum	1.00	1.00	1.00	1.00	1.00	1.00	1.00
Quartile 3	0.90	0.99	1.00	0.96	0.97	0.96	1.00

Sample ranking		Iman-Davenport's	Hommel's							Statistical ranking	
Technique	Rank	p-value	adjusted p-values							Technique	Rank
				Trinity	Aleph	RoadRunner	FiVaTech	Wien	SoftMealy		
ROLLER	2.68	1.22E-07	ROLLER	8.05E-02	9.11E-03	1.55E-03	8.64E-04	8.95E-05	6.41E-08	ROLLER	1
Trinity	3.42									Trinity	1
Aleph	3.88									Aleph	2
RoadRunner	4.15									RoadRunner	2
FiVaTech	4.25									FiVaTech	2
Wien	4.50									Wien	2
SoftMealy	5.11									SoftMealy	2

Table 3.8: *Summary of results regarding recall.*

have similarly regarding recall, that is, they both rank statistically at the first position; however, they provide enough evidence to reject the hypothesis that ROLLER behaves similarly to Aleph, RoadRunner, FiVaTech, Wien, and SoftMealy, that is, they rank worse than ROLLER and Trinity.

Table §3.9 summarises the results regarding the F_1 score. Empirically, ROLLER seems to be the proposal that can achieve the best F_1 score, and it is, again, the most stable. Iman-Davenport's test returns a p-value that is nearly zero, which strongly supports the hypothesis that there are statistically significant differences in rank. Hommel's test returns adjusted p-values that are clearly smaller than the significance level in every case, which supports the hypothesis that the differences in rank amongst ROLLER and every other proposal are statistically significant, too.



	SoftMealy	WIEN	RoadRunner	FiVaTech	Trinity	Aleph	Roller
Quartile 1	0.44	0.36	0.18	0.55	0.81	0.87	0.94
Minimum	0.04	0.00	0.00	0.04	0.58	0.52	0.74
Median	0.73	0.66	0.59	0.75	0.86	0.94	0.96
Maximum	1.50	0.94	0.97	0.97	1.00	1.00	1.00
Quartile 3	0.83	0.83	0.80	0.86	0.90	0.96	1.00

Sample ranking		Iman-Davenport's	Hommel's							Statistical ranking	
Technique	**Rank**	**p-value**	**adjusted p-values**							**Technique**	**Rank**
				Aleph	**Trinity**	**SoftMealy**	**FiVaTech**	**Wien**	**RoadRunner**		
Roller	1.70	4.44E-16	**Roller**	3.08E-03	2.57E-06	3.00E-10	2.62E-11	3.34E-14	4.91E-11	**Roller**	1
Aleph	2.72									**Aleph**	2
Trinity	3.23									**Trinity**	2
SoftMealy	4.82									**SoftMealy**	2
FiVaTech	4.89									**FiVaTech**	2
Wien	5.22									**Wien**	2
RoadRunner	5.41									**RoadRunner**	2

Table 3.9: *Summary of results regarding the* F_1 *score.*

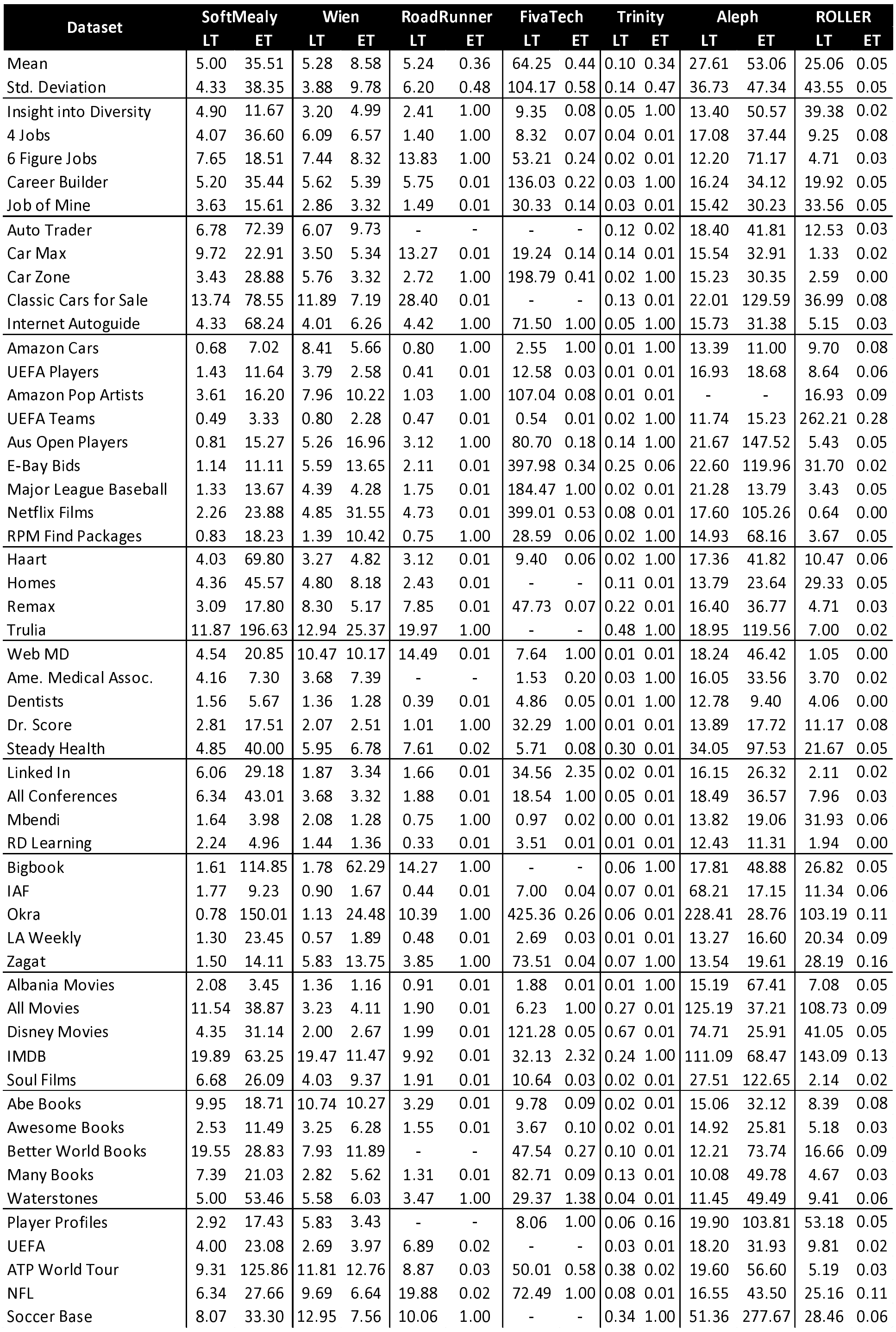
Since ROLLER works on the tree representation of the input documents, we need to parse them and correct the errors in their HTML code. Such errors are very common, cf. Table §A.1. As a conclusion, it was also necessary to carry out a statistical analysis to find out if our experiments provide enough evidence to conclude that the presence of errors in the input documents has an impact on the effectiveness of our proposal. We have used Kendall's Tau test, which returned $\tau = -0.09$ with p-value 0.37. Note that τ is very close to zero and that the p-value is clearly greater than the standard significance level, which means that the experimental data do not provide enough evidence to reject the hypothesis that the correlation is zero. In other words, our experiments do not provide any evidence that the effectiveness of our proposal may be biased by the presence of errors in the HTML code of the input documents.

Our conclusions are that ROLLER outperforms the other proposals regarding effectiveness and that it is the proposal whose results are more stable. The statistical tests that we have performed have found enough evidence in our experimental data to support the hypothesis that the differences in the empirical rank amongst ROLLER and the other proposals are significant at the standard significance level, except for the case of recall, in which case the experimental data do not provide enough evidence to conclude that ROLLER and Trinity perform differently. Note, too, that proposals like RoadRunner and FiVaTech cannot deal with all of our datasets; in Table §3.6 such situations are indicated with a dash. The reason is that they took more than 1 CPU day to learn a rule or that they raised an exception; in both cases, we could not compute effectiveness measures for the corresponding datasets.

3.4.3 Efficiency analysis

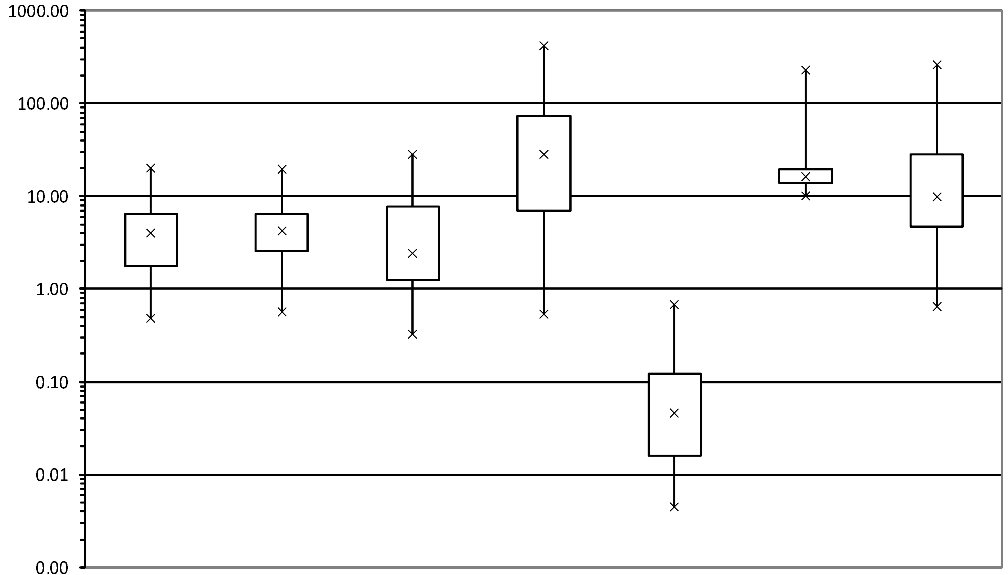
Table §3.10 reports on the raw efficiency data that we got from our experimentation. For each proposal, we report on its efficiency measures regarding our datasets. The first two lines also provide a summary of the results in terms of mean value and the standard deviation. Since it is difficult to spot a trend in this table, we decided to summarise the data using boxplots.

Table §3.11 summarises the results regarding learning times, that is, the mean CPU time that each proposal took to learn a rule set. Experimentally, it seems that Trinity is the proposal that takes less time to learn a rule set; in most cases, it does not take more than a tenth of a second. It is followed by RoadRunner, SoftMealy, and Wien, whose learning times are very similar; ROLLER seems to rank at the fifth position, before Aleph and FiVaTech, which are the most inefficient. Iman-Davenport's test returns a p-value that is very close to zero, which clearly supports the hypothesis that there are differences in rank amongst these proposals. Hommel's test also returns adjusted p-values that are very small with respect to the significance level, which also reveals that the experimental data provide enough evidence to support the hypothesis that Trinity is the proposal that performs the best and that the others rank below it. Note that we do not think that this is a serious shortcoming since our learning times still lie within the range of a few seconds in most cases and we assume that learning rules is not a task that must be executed continuously in a production



Dataset	SoftMealy		Wien		RoadRunner		FivaTech		Trinity		Aleph		ROLLER	
	LT	ET	LT	ET	LT	ET	LT	ET	LT	ET	LT	ET	LT	ET
Mean	5.00	35.51	5.28	8.58	5.24	0.36	64.25	0.44	0.10	0.34	27.61	53.06	25.06	0.05
Std. Deviation	4.33	38.35	3.88	9.78	6.20	0.48	104.17	0.58	0.14	0.47	36.73	47.34	43.55	0.05
Insight into Diversity	4.90	11.67	3.20	4.99	2.41	1.00	9.35	0.08	0.05	1.00	13.40	50.57	39.38	0.02
4 Jobs	4.07	36.60	6.09	6.57	1.40	1.00	8.32	0.07	0.04	0.01	17.08	37.44	9.25	0.08
6 Figure Jobs	7.65	18.51	7.44	8.32	13.83	1.00	53.21	0.24	0.02	0.01	12.20	71.17	4.71	0.03
Career Builder	5.20	35.44	5.62	5.39	5.75	0.01	136.03	0.22	0.03	1.00	16.24	34.12	19.92	0.05
Job of Mine	3.63	15.61	2.86	3.32	1.49	0.01	30.33	0.14	0.03	0.01	15.42	30.23	33.56	0.05
Auto Trader	6.78	72.39	6.07	9.73	-	-	-	-	0.12	0.02	18.40	41.81	12.53	0.03
Car Max	9.72	22.91	3.50	5.34	13.27	0.01	19.24	0.14	0.14	0.01	15.54	32.91	1.33	0.02
Car Zone	3.43	28.88	5.76	3.32	2.72	1.00	198.79	0.41	0.02	1.00	15.23	30.35	2.59	0.00
Classic Cars for Sale	13.74	78.55	11.89	7.19	28.40	0.01	-	-	0.13	0.01	22.01	129.59	36.99	0.08
Internet Autoguide	4.33	68.24	4.01	6.26	4.42	1.00	71.50	1.00	0.05	1.00	15.73	31.38	5.15	0.03
Amazon Cars	0.68	7.02	8.41	5.66	0.80	1.00	2.55	1.00	0.01	1.00	13.39	11.00	9.70	0.08
UEFA Players	1.43	11.64	3.79	2.58	0.41	0.01	12.58	0.03	0.01	0.01	16.93	18.68	8.64	0.06
Amazon Pop Artists	3.61	16.20	7.96	10.22	1.03	1.00	107.04	0.08	0.01	0.01	-	-	16.93	0.09
UEFA Teams	0.49	3.33	0.80	2.28	0.47	0.01	0.54	0.01	0.02	1.00	11.74	15.23	262.21	0.28
Aus Open Players	0.81	15.27	5.26	16.96	3.12	1.00	80.70	0.18	0.14	1.00	21.67	147.52	5.43	0.05
E-Bay Bids	1.14	11.11	5.59	13.65	2.11	0.01	397.98	0.34	0.25	0.06	22.60	119.96	31.70	0.02
Major League Baseball	1.33	13.67	4.39	4.28	1.75	0.01	184.47	1.00	0.02	0.01	21.28	13.79	3.43	0.05
Netflix Films	2.26	23.88	4.85	31.55	4.73	0.01	399.01	0.53	0.08	0.01	17.60	105.26	0.64	0.00
RPM Find Packages	0.83	18.23	1.39	10.42	0.75	1.00	28.59	0.06	0.02	1.00	14.93	68.16	3.67	0.05
Haart	4.03	69.80	3.27	4.82	3.12	0.01	9.40	0.06	0.02	1.00	17.36	41.82	10.47	0.06
Homes	4.36	45.57	4.80	8.18	2.43	0.01	-	-	0.11	0.01	13.79	23.64	29.33	0.05
Remax	3.09	17.80	8.30	5.17	7.85	0.01	47.73	0.07	0.22	0.01	16.40	36.77	4.71	0.03
Trulia	11.87	196.63	12.94	25.37	19.97	1.00	-	-	0.48	1.00	18.95	119.56	7.00	0.02
Web MD	4.54	20.85	10.47	10.17	14.49	0.01	7.64	1.00	0.01	0.01	18.24	46.42	1.05	0.00
Ame. Medical Assoc.	4.16	7.30	3.68	7.39	-	-	1.53	0.20	0.03	1.00	16.05	33.56	3.70	0.02
Dentists	1.56	5.67	1.36	1.28	0.39	0.01	4.86	0.05	0.01	1.00	12.78	9.40	4.06	0.00
Dr. Score	2.81	17.51	2.07	2.51	1.01	1.00	32.29	1.00	0.01	0.01	13.89	17.72	11.17	0.08
Steady Health	4.85	40.00	5.95	6.78	7.61	0.02	5.71	0.08	0.30	0.01	34.05	97.53	21.67	0.05
Linked In	6.06	29.18	1.87	3.34	1.66	0.01	34.56	2.35	0.02	0.01	16.15	26.32	2.11	0.02
All Conferences	6.34	43.01	3.68	3.32	1.88	0.01	18.54	1.00	0.05	0.01	18.49	36.57	7.96	0.03
Mbendi	1.64	3.98	2.08	1.28	0.75	1.00	0.97	0.02	0.00	0.01	13.82	19.06	31.93	0.06
RD Learning	2.24	4.96	1.44	1.36	0.33	0.01	3.51	0.01	0.01	0.01	12.43	11.31	1.94	0.00
Bigbook	1.61	114.85	1.78	62.29	14.27	1.00	-	-	0.06	1.00	17.81	48.88	26.82	0.05
IAF	1.77	9.23	0.90	1.67	0.44	0.01	7.00	0.04	0.07	0.01	68.21	17.15	11.34	0.06
Okra	0.78	150.01	1.13	24.48	10.39	1.00	425.36	0.26	0.06	0.01	228.41	28.76	103.19	0.11
LA Weekly	1.30	23.45	0.57	1.89	0.48	0.01	2.69	0.03	0.01	0.01	13.27	16.60	20.34	0.09
Zagat	1.50	14.11	5.83	13.75	3.85	1.00	73.51	0.04	0.07	1.00	13.54	19.61	28.19	0.16
Albania Movies	2.08	3.45	1.36	1.16	0.91	0.01	1.88	0.01	0.01	1.00	15.19	67.41	7.08	0.05
All Movies	11.54	38.87	3.23	4.11	1.90	0.01	6.23	1.00	0.27	0.01	125.19	37.21	108.73	0.09
Disney Movies	4.35	31.14	2.00	2.67	1.99	0.01	121.28	0.05	0.67	0.01	74.71	25.91	41.05	0.05
IMDB	19.89	63.25	19.47	11.47	9.92	0.01	32.13	2.32	0.24	1.00	111.09	68.47	143.09	0.13
Soul Films	6.68	26.09	4.03	9.37	1.91	0.01	10.64	0.03	0.02	0.01	27.51	122.65	2.14	0.02
Abe Books	9.95	18.71	10.74	10.27	3.29	0.01	9.78	0.09	0.02	0.01	15.06	32.12	8.39	0.08
Awesome Books	2.53	11.49	3.25	6.28	1.55	0.01	3.67	0.10	0.02	0.01	14.92	25.81	5.18	0.03
Better World Books	19.55	28.83	7.93	11.89	-	-	47.54	0.27	0.10	0.01	12.21	73.74	16.66	0.09
Many Books	7.39	21.03	2.82	5.62	1.31	0.01	82.71	0.09	0.13	0.01	10.08	49.78	4.67	0.03
Waterstones	5.00	53.46	5.58	6.03	3.47	1.00	29.37	1.38	0.04	0.01	11.45	49.49	9.41	0.06
Player Profiles	2.92	17.43	5.83	3.43	-	-	8.06	1.00	0.06	0.16	19.90	103.81	53.18	0.05
UEFA	4.00	23.08	2.69	3.97	6.89	0.02	-	-	0.03	0.01	18.20	31.93	9.81	0.02
ATP World Tour	9.31	125.86	11.81	12.76	8.87	0.03	50.01	0.58	0.38	0.02	19.60	56.60	5.19	0.03
NFL	6.34	27.66	9.69	6.64	19.88	0.02	72.49	1.00	0.08	0.01	16.55	43.50	25.16	0.11
Soccer Base	8.07	33.30	12.95	7.56	10.06	1.00	-	-	0.34	1.00	51.36	277.67	28.46	0.06

Table 3.10: *Efficiency results.*



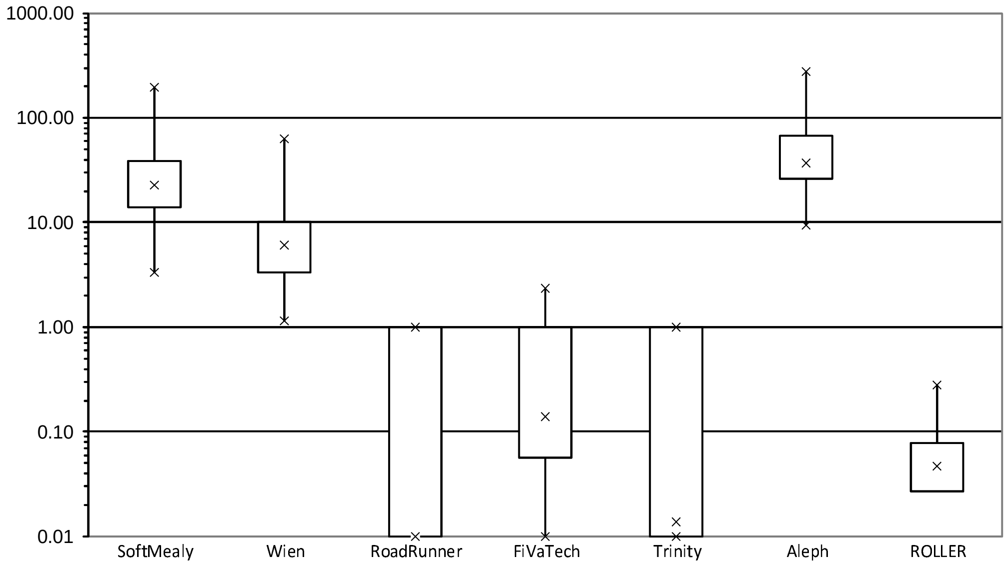
	SoftMealy	Wien	RoadRunner	FiVaTech	Trinity	Aleph	ROLLER
Quartile 1	1.74	2.54	1.24	7.00	0.02	13.85	4.71
Minimum	0.49	0.57	0.33	0.54	0.00	10.08	0.64
Median	4.05	4.21	2.42	28.59	0.05	16.40	9.76
Maximum	19.89	19.47	28.40	425.36	0.67	228.41	262.21
Quartile 3	6.43	6.43	7.67	72.49	0.12	19.75	28.26

Sample ranking		Iman-Davenport's	Hommel's							Statistical ranking	
Technique	Rank	p-value	adjusted p-values							Technique	Rank
Trinity	2.10	1.52E-113		ROLLER	RoadRunner	FivaTech	Wien	SoftMealy	Aleph	Trinity	1
ROLLER	2.27		Trinity	6.83E-01	6.83E-01	1.69E-02	5.72E-12	1.32E-21	1.21E-26	ROLLER	1
RoadRunner	2.37									RoadRunner	1
FivaTech	3.27									FivaTech	2
Wien	5.10									Wien	2
SoftMealy	6.21									SoftMealy	2
Aleph	6.69									Aleph	2

Table 3.11: *Summary of results regarding learning times.*

scenario. Note, too, that improving the efficiency whilst keeping the effectiveness was an important part of our motivation to work on a proposal that improves on TANGO, and we think that we have definitely done it.

Table §3.12 summarises the results regarding extraction times, that is, the mean CPU time that it took to apply a rule set to a dataset. Aleph, SoftMealy, Wien, and FivaTech seem to be the proposals that have the worst performance; RoadRunner and Trinity seem to be very similar in both mean extraction time and deviation since their inter-quartile ranges are identical. The timings regarding ROLLER are the best, since most rules do not



	SoftMealy	Wien	RoadRunner	FiVaTech	Trinity	Aleph	ROLLER
Quartile 1	14.00	3.34	0.01	0.06	0.01	25.86	0.03
Minimum	3.33	1.16	0.01	0.01	0.01	9.40	0.00
Median	23.00	6.15	0.01	0.14	0.01	36.77	0.05
Maximum	196.63	62.29	1.00	2.35	1.00	277.67	0.28
Quartile 3	39.15	10.18	1.00	1.00	1.00	67.79	0.08

Sample ranking		Iman-Davenport's	Hommel's							Statistical ranking	
Technique	Rank	p-value	adjusted p-values							Technique	Rank
				ROLLER	RoadRunner	FiVaTech	Wien	SoftMealy	Aleph		
Trinity	2.10	1.52E-113	Trinity	6.83E-01	6.83E-01	1.69E-02	5.72E-12	1.32E-21	1.21E-26	Trinity	1
ROLLER	2.27									ROLLER	1
RoadRunner	2.37									RoadRunner	1
FivaTech	3.27									FivaTech	2
Wien	5.10									Wien	2
SoftMealy	6.21									SoftMealy	2
Aleph	6.69									Aleph	2

Table 3.12: *Summary of results regarding extraction times.*

take more than a tenth of a second to extract information, and its interquartile range is also very small; its mean time is also very good and its results are more stable than the rest because its deviation is the smallest. Iman-Davenport's test returns a p-value that is nearly zero, which clearly indicates that there are statistically significant differences in the empirical ranks. Hommel's test returns adjusted p-values that are not smaller than the standard significance level regarding the comparisons of Trinity, which is the best-ranked proposal according to the empirical ranking, ROLLER, and RoadRunner. This means that the experimental data do not provide enough evidence to conclude that there is a statistically significant difference amongst Trinity, ROLLER, and RoadRunner regarding extraction times, that is, they all rank at the first position. The test, however, finds enough evidence to reject the hypothesis that the previous proposals and the others behave similarly regarding the extraction time. These results are very important, because they confirm that the rules that ROLLER learns are very competitive regarding efficiency.

As a conclusion, our experiments support the idea that ROLLER is very efficient. It is not the best performing regarding learning times, but it still lies within the range of seconds, which we do not think is a serious shortcoming from a practical point of view. However, the rules that it learns are as efficient as the rules that other state-of-the-art proposals can learn, which makes them competitive from a practical point of view. The reason why ROLLER takes a little more time to learn a rule than other proposals is that it has to create several learning sets and then apply the base learner several times; its efficiency clearly depends on how effective the base learner is. Anyway, we think that the efficiency results are quite reasonable and that its superiority regarding effectiveness clearly compensates for its slightly worse performance.

3.5 Related work

In this section, we delve into propositio-relational machine-learning proposals, which are closely related to ours, but have not been explored so far in our context; then, we compare them with ours from the following perspectives: adaptability, balance between effectiveness and efficiency, and approach to the problem.

3.5.1 Overview of related proposals

Inductive logic programming is a natural approach to deal with relational data. Unfortunately, it is inefficient when the datasets scale in the number of data or features because the search space is typically huge [16, 19, 59, 62, 90, 132]. We have also explored applying inductive logic programming in the context of web information extraction in Chapter §2, and our experiments highlighted the need to optimise its inefficient learning process. This has motivated some authors to work on adapting efficient propositional techniques so that they can work on relational data. The proposals in the literature can be broadly classified as follows [74, 98]: upgrading, flattening (aka. proposionalisation), and multiple view.[†1]

[†1] Note that we use adjective propositio-relational in accordance with the many proposals in which it is used to mean that a propositional technique is adapted to deal with relational

Upgrading proposals rely on a conventional propositional learner that is upgraded to deal with relational features. Some proposals upgrade a propositional learner with the ability to learn first-order rules, namely: TILDE [15] upgrades C4.5, SCART [99] upgrades CART, RIBL [50] and RIBL2 [84] upgrade k-NN, Cumby and Roth [41] and Gärtner and others [67] upgraded some kernel methods, PRM [70, 89] upgrades Bayesian networks, SLP [131] upgrades stochastic grammars, and 1BC and 1BC2 [56] upgrade Bayesian classifiers. Unfortunately, these proposals did not prove to be efficient enough [74], which motivated other authors to work on so-called relational-database proposals that transform the original problems into SQL representations that can be handled more efficiently with commodity database management systems. There are two approaches in the literature: selection graph model, which includes MRDTL [10], which builds on TILDE but represents the data in SQL, and MRDTL2 [10], which is an optimised version of MRDTL that can also handle missing attributes using a proposal based on Naive Bayes classifiers; other proposals are based on a technique called tuple ID propagation, which basically attempts to join related vectors virtually; for instance, CrossMine [181] and GraphNB [184] follow this approach by extending FOIL [139] and a Bayesian classification algorithm, respectively.

Flattening proposals convert relational data into table-based representations to which standard propositional techniques can be applied. There are two approaches in the literature: creating universal vectors that join all of the data in the learning sets, which was pioneered by LINUS [46], DINUS [114], and SINUS [101], or creating vectors that summarise and/or aggregate the data in the neighbourhood of every vector, e.g., RollUp [96] and RELAGGS [102].

Guo and Viktor [74] devised the only multiple-view proposal of which we are aware. It relies on a meta-learning approach that can learn from multiple views of the data, that is, multiple subsets of data that result from projecting them using different feature subsets, and then integrates the results using a novel technique that does not require the complex preprocessing required by flattening proposals.

data. There are a few proposals in the literature that are also called propositio-relational, but have little to do with the previous idea. For instance, nFOIL and tFOIL [110], kFOIL [111], or SAYU [43] address the problem of learning a classifier that can help decide amongst a number of rules the one that might provide the correct class to an unseen example; the classifier works on a propositionalised version of the original learning set in which each example is transformed into a binary vector in which each component indicates whether a rule holds or not on that example. In these proposals, adjective propositio-relational is used to mean that they merge a propositional and a relational learner.

Although propositio-relational proposals seem very adequate to deal with the problem of learning information extraction rules, it remains an almost unexplored research path. The only exception is the work by Sleiman and Corchuelo [161], who devised a proposal that hybridises finite automata and neural networks; the states of the automata represent the information to be extracted and the transitions the next-token relational feature; the transitions are controlled by means of neural networks that recognise token patterns building on simple features like they their HTML tags or their lexical classes.

3.5.2 Adaptability

Typically, researchers who are interested in web information extraction have designed ad-hoc proposals that are specifically tailored to this problem, which has led to a variety of alternatives. Although many of them were proven to be very effective and efficient, the problem is that they cannot leverage the many advances in the field of Machine Learning; neither can the general machine-learning field easily benefit from them. Furthermore, many of them have faded away quickly as their inherent assumptions about the structure of documents have become obsolete as the Web has evolved. Unfortunately, they could not be easily adapted to deal with such evolution because this would have required to re-work them, that is, to have devised completely new proposals. Some of the ad-hoc proposals that we have surveyed work on the textual representation of the input documents and their goal is to characterise the left and the right context of the information to extract; others work on their DOM tree representations and their goal is to characterise the path from the root node to the nodes that provide the information to be extracted.

Contrarily to the previous proposals, ROLLER can leverage many machine-learning techniques in the literature and benefit from the advances in this field. Furthermore, it is based on an open catalogue of features that can be easily extended and adapted as the Web evolves, without changing the proposal itself. Neither does ROLLER attempt to characterise the left or the right context of the information to extract or the path from the root to the nodes that provide the information to extract; but it tries to characterise a context in the DOM tree. Note that this may involve tokens in disparate positions, not necessarily on the left or the right, as well as tokens that are not on the same path to the root node, e.g., siblings or children of siblings.

3.5.3 Balance between effectiveness and efficiency

A few authors have explored using techniques that got inspiration from inductive logic programming since the tokens or the DOM nodes of semi-structured documents can be naturally represented as relational data. Their proposals are expected to be easier to adapt as the Web evolves since they need not be adapted, but their catalogues of features. In general, they can achieve high effectiveness at the cost of efficiency. They explore an unbounded context, which does not restrict them to the left or the right context or nodes within a given path, as was the case for the ad-hoc proposals. Unfortunately, they use the same heuristic to guide the search through both attributive and relational features, which typically results in a problem called myopia. The problem is that these proposals do not look ahead, which means that adding relational features to further characterise the information in the context does not always entail an improvement if characterising that surrounding information does not help discern between the information to be extracted and the information not to be extracted. In other words, when a relational feature is selected, the attributive features of the target node are not taken into account and there are cases in which a decision to explore a neighbour node may lead to a local minimum. Except for L-Wrappers and TANGO, none of the proposals that we have surveyed can backtrack to explore other choices. Note, too, that TANGO and L-wrappers are the only proposals that advocate transforming the problem of web information extraction into a first-order knowledge base and then learning extraction rules using an inductive-logic learner. This proved not to be efficient enough, even with relatively simple documents. This problem was first pointed out by Freitag, who suggested that learning from a first-order representation would simply be too inefficient.

ROLLER also works on a relational representation of the input documents that builds on an open catalogue of features that can easily evolve as the Web evolves, without making a change to the proposal itself. Furthermore, it relies on a propositional base learner that can be integrated in our proposal without a change; that is, it can benefit from the advances in the general field of Machine Learning. Our experiments prove that ROLLER is very effective and efficient. This is because it relies on a propositional learner to analyse the attributive features of the nodes to extract and then explores their context using relational features in an attempt to find neighbour nodes whose attributive features can contribute to learning a better rule. Furthermore, two different search heuristics are involved: one that is provided by the base learner, which is ad-hoc and was designed to guide the search through attributive features as effectively and efficiently as possible, and another one that was designed to guide the search through the relational features and helps explore the context as effectively and efficiently as possible. ROLLER also reduces myopia because it deals with all of the attributive features at the same time, not one after the other as was the case for the existing proposals; furthermore, the decision on which relational feature has to be explored next does not depend only on that feature itself, but on the attributive features of the target nodes. Obviously, this is not a solution to myopia, but our experiments prove that it reduces the odds of making wrong decisions; we explored using backtracking, but our experiments proved that the mechanism was not actually necessary, so we decided not to include it in the final version of ROLLER.

3.5.4 Learning procedure

Since information extraction problems can be naturally represented using relational data, one might think that it would be easy to leverage a proposal from the field of propositio-relational learning. Unfortunately, few such proposals exist in the literature since there are a number of intrinsic problems: according to Guo and Viktor [74], upgrading proposals are not generally scalable-enough, chiefly those that rely on inductive logic programming approaches, and cannot generally achieve high effectiveness when they deal with numeric data, which is very common in our context, e.g., depth of a node, number of children, font size, ratio of letters or figures, text length, co-ordinates, and the like. Relational-database proposals are more efficient because they rely on a database management system, but they do not seem easy to adapt to the problem of information extraction because they rely on a fully-fledged relational schema, that is, they were designed to deal with actual relational databases that build on a rich data schema that includes information about every attribute, primary keys, foreign keys, and so on; in other words, they are schema-driven proposals. In our context, there is not such a schema, which requires the proposals to be instance-driven, that is: they must explore the context of every instance individually, without an explicit schema. The existing multi-view proposal in the literature improves on efficiency, but it does not seem appropriate in our context because it is based on aggregating neighbour vectors. Numeric features are aggregated using the standard SQL aggregation functions (sum, average, minimum, maximum, standard deviation, and count), but categoric features are aggregated using counts only. That means that the classification power that such features can provide is lost, and such features are very common in our context, e.g., font family, colour, horizontal alignment, floating specification, and the like; furthermore, it does not take into account that attributes in disparate nodes can contribute to obtaining a good rule. Flattening proposals require much computation to flatten the datasets to be analysed and the resulting vectors may have an arbitrarily large number of components, which hinders the applicability of many learners in practice; some of the proposals require data to be duplicated, which increases statistical skewness, whereas others require data to be aggregated, which implies that data distributions are neglected; furthermore, they need to put a limit to the amount of context that can be explored because the context of the data is not explored on-demand, but pre-computed.

ROLLER naturally fits within the category of flattening proposals, but it differs significantly from the existing ones: instead of pre-processing the vectors in an attempt to make the context of every node explicit, it first tries to learn a rule that is a solution building solely on the attributive features of the nodes to be extracted; if no such a rule can be learnt, then it explores the context by means of the available relational features, which involves flattening the vectors that correspond to the nodes being analysed and the vectors that correspond to their neighbours. This results in a dynamic flattening proposal that has proven to work very well in practice according to our experiments. Note that, contrarily to existing proposals, no aggregation of data is required; it works on the attributive features themselves, which implies that no classification power is lost in the flattening process.

3.6 Summary

In this chapter, we present ROLLER, which is a new proposal to learn web information extraction rules. It relies on an open catalogue of features, which helps adapt it as the Web evolves; furthermore, it does not commit to a specific base learner or rule scorer, but can leverage many proposals in the literature and thus benefit from the continuous advances in the general field of Machine Learning. This clearly deviates from the many existing ad-hoc proposals in the literature and from the few existing proposals that are based on inductive-logic programming techniques. Technically, the learner that underlies our proposal relies on a search procedure that uses a new dynamic flattening technique to explore an unbound context of the nodes that provide the information to be extracted; our survey of the literature proves that is a novel approach to the problem. The experimental results proved that ROLLER beats others in the literature regarding effectiveness, and is very competitive regarding efficiency since learning times lie within the range of seconds, which we think is a significant improvement regarding learning times in TANGO and it is very fast from a practical point of view.

Chapter 4

VENICE: a method to rank information extractors

Our proposal to compare and rank information extractors in a homogeneous, fair, and stringent way is VENICE, which we describe in this chapter. The rest of the chapter is organised as follows: Section §4.1 motivates our work and sketches our proposal; Section §4.2 describes the details of our method, Section §4.3 presents a case study that illustrates how it works in practice, Section §4.4 reports on the related work, and Section §4.5 summarises our conclusions. Appendix §B provides additional details on the performance measures that we discuss in this chapter.

4.1 Introduction

The authors of new information extraction proposals must obviously compare them to others so that they can prove that they have introduced conceptual innovations that advance the state of the art. But this is not enough: it is also necessary to rank them regarding their performance; in other words, it is necessary to evaluate them regarding some effectiveness and efficiency measures and then compare the results so as to compute a single ranking in which the best-performing proposals are at the top. Unfortunately, there is not such an objective ranking method in the literature.

In our opinion, a good ranking method must have the following key features: it must be automated, so that researchers can bias the conclusions as little as possible, open, so that it can easily accommodate new performance measures, and agnostic, so that it can be applied to as many different kinds of proposals as possible. Furthermore, it must also address the following key questions: how to set up the experimental environment, how to create appropriate evaluation splits, how to compute the experimental data, how to cook them (regarding how to purge them, compute derived measures, and/or normalise them), how to compute the rankings, and how to report on the results. Unfortunately, neither informal methods nor formal methods have addressed these key features or questions but rather provide a foundation and some guidelines. The informal methods were not intended to be reused, but to help the authors of a proposal support the idea that it outperforms others; as a conclusion, they are not automated, open, or agnostic, but ad-hoc; furthermore, they do not usually disclose many important details regarding the experimental environment; it is not commonly clear how the evaluation splits are created; the experimental data are partial, biased, and it is not clear how the matchings required to compute effectiveness measures are counted; the experimental data are not cooked; and the resulting rankings are not generally statistically sound. As a conclusion, the stringency level varies from paper to paper, which makes the results available in the literature difficult to reuse when a new proposal needs to be compared to them. The formal methods are generally supported by software tools that aid in computing the experimental data, but they are not actually automated; neither are they open, since they commit to a particular set of performance measures and everything in the method revolves around them; they all originated in a community that was interested in supervised free-text proposals, so they have not paid attention to other kinds of proposals; they report on several alternatives to create evaluation splits, but do not assess the pros and cons or commit to a specific method; they compute experimental data that are partial, biased, and it is not commonly clear how the matchings required to compute effectiveness measures are counted; they do not provide a method to cook the experimental data; and the resulting rankings must be handcrafted, although they pay attention to ensuring that the results are statistically sound.

In this chapter, we report on a method to evaluate, compare, and then rank web information extraction proposals. It overcomes the problems that we have found in the literature in that it reduces the bias that a researcher can introduce in the results because it is automated; it can easily accommodate new performance measures as they are devised and proven to be adequate in our context because it is open; it does not commit to a particular kind of extractor, but has been designed to rank as many proposals as possible because it is agnostic; it provides a clear guideline regarding how the experimental environment must be set up, with a special emphasis on selecting the most appropriate set of performance measures so that the conclusions are global and unbiased; it provides a method to compute as many evaluation splits as possible out of the datasets available; it provides a method to compute the experimental data that takes how matchings are computed into account and does not neglect unsupervised or heuristic-based proposals; it provides a new statistically-sound method to purge the experimental data, it also takes derived measures into account, and provides a normalisation method that is the key for our proposal to be open; and it provides a statistically-sound method to compute per-measure rankings and then combine them all taking into account both a researcher's preferences and the deviations of the performance measures.

4.2 Description of our method

Our method consists of the steps that are summarised in Figure §4.1. The first step consists in setting up the experimental environment. The second step consists in computing a number of evaluation splits, that is, pairs of learning and testing sets. The third step consists in running the selected proposals on the previous evaluation splits to gather raw experimental data, that is, the values of the selected performance measures as they are computed on the available evaluation splits. The forth step consists in cooking the raw experimental data as follows: first, they are purged, then derived measures are computed, and, finally, the purged data and the derived data are normalised. The fifth step consists in computing a local ranking per performance measure and then a global ranking. The last step consists in producing a report that summarises the study.

```
Step 1: set up the experimental environment
    – Describe the hardware and the software
    – $proposals$ = select some proposals
    – $datasets$ = select some datasets
    – $measures$ = select some performance measures
    – Set parameters:
        $\alpha$ = select a statistical significance level
        $\gamma$ = select the number of repetitions to compute evaluation splits
        $\omega$ = select the relative weights of performance measures
        $\eta$ = select a purging measure from $measures$
Step 2: create evaluation splits
    – $splits = computeEvaluationSplits(datasets)$
Step 3: compute raw experimental data
    – $rawData = runExperiments(proposals, measures, splits)$
Step 4: cook the experimental data
    – $purgedData = purgeData(rawData)$
    – $derivedData = computeDerivedData(purgedData)$
    – $normalisedData = normaliseData(purgedData \cup derivedData)$
Step 5: compute rankings
    – $localRankings = computeLocalRankings(normalisedData)$
    – $globalRanking = computeGlobalRanking(normalisedData)$
Step 6: produce a report
```

Figure 4.1: *Steps of our method.*

In the following subsections, we provide additional details on each step.

4.2.1 Step 1: set up the experimental environment

The experimental environment consists of the hardware and the software used to evaluate a number of proposals plus some data that must be provided by a researcher, namely: the proposals to be ranked, the datasets on which they must be evaluated, the performance measures to compare them, and the values of the parameters of our method. Typically, the researcher is an author who has devised a new web information extraction proposal and wishes to rank it with respect to others in the literature, or a practitioner who needs to extract information from a web site and has to make an informed decision regarding which of the proposals in the literature is the most appropriate. We provide additional details in the following subsections.

Describe the hardware and the software. Our suggestion is that the researcher should describe the hardware and the software that she or he used to perform her or his experiments. We do not think that it is necessary to describe them thoroughly, because it is very unlikely that other researchers can reproduce exactly the same environment, but it commonly helps have an overall idea of the experimental conditions. Clearly, running an experimentation on a mid-class computer is not the same as running it on a super-computing facility, and this should be made explicit so that the efficiency results can be assessed properly.

Select some proposals. We suggest that the researcher should select the most closely-related proposals (where closeness is measured in terms of conceptual similarity) and some state-of-the-art ones (where state-of-the-art is measured in terms of how recent, sound, and/or well-performing they are). The most closely related proposals must be selected because, otherwise, we cannot prove that the conceptual innovations in a new proposal are worth from a practical point of view; but not only must a new proposal beat the most closely-related ones, but also others that are conceptually different but have proven to achieve good performance.

Select some datasets. Having standard dataset repositories is very important since they allow to compare different proposals on a corpora that can be carefully selected so that the documents are representative enough of both the regularities and irregularities with which web information extractors have to deal. In other words, such repositories should provide controlled, well-documented datasets that put an emphasis on the many difficult cases with which an information extractor has to deal. This makes it impossible that a proposal is evaluated only on datasets on which it works very well and, thus, reduces the chances to bias the results.

The list of public repositories available includes RISE [133], TBDW [179], and TIPSTER [170]; furthermore, Freitag [62] compiled the Seminar Announcement collection, Califf and Mooney [21] compiled the Job Posting collection, and Reuters made available the Reuters-2157 collection on company acquisitions [143]; other authors have assembled their own public repositories [4, 6, 38, 100, 156, 168] and some conferences have also published some repositories, chiefly the MUC conferences.

Note that not every repository provides adequate datasets for every proposal. For instance, there are repositories that focus on free-text documents and others that focus on semi-structured documents; there are repositories that focus on single-record documents and others that focus on multi-record documents; furthermore, there are many repositories in which each dataset was gathered from a single site and others in which some datasets were gathered from different sites and are then appropriate for open information extractors only. A researcher must select the repositories and the datasets that are adequate for the proposals that she or he wishes to rank.

Regarding the number of datasets, there is not a standard in the literature. Our suggestion is that there should be at least 20–30 datasets available and that each one should provide at least 20–30 documents; these figures are generally considered large enough to draw statistically solid conclusions [153]. VENICE requires the datasets to provide the same number of documents, so that it can create appropriate evaluation splits. In cases in which the datasets available do not provide the same number of documents, our suggestion is to discard some documents randomly or to split them into several smaller datasets. The reason why we do not think that the datasets should provide more than 20–30 documents is that they all have to be annotated for evaluation purposes, which is a time-consuming and error-prone task; neither think we that a proposal that requires a large number of documents to achieve good results is useful from a practical point of view.

Select some performance measures. Performance measures can be classified into effectiveness and efficiency measures. The former focus on assessing how good the results of a proposal are, that is, its ability to learn rules that make a clear difference between the information to be extracted and the information to be ignored. The latter focus on the computing resources that it requires to do so.

We suggest that the researcher should select a number of effectiveness measures that must fulfil the following requirements: a) they must be global, that is, they must provide an overview of how a proposal behaves regarding every kind of error it can make; b) they must not be biased in the presence of unbalanced datasets, which are natural in web information extraction; c) and they must be extensible from a per-slot level to a per-extractor level in an unbiased manner. Regarding the efficiency measures, our only requirement is that they must be stable, that is, they should not vary significantly when a proposal is applied multiple times to the same evaluation split. Furthermore, the set of selected performance measures must fulfil the following requirements: a) they must be orthogonal, that is, they must focus on assessing different complementary aspects of performance in an attempt to provide as a wide view of a proposal as possible when combining them; b) and there must not be many measures, since, otherwise, their individual contribution to the global ranking blurs easily.

Our proposal is to use three types of effectiveness measures, namely: a) error-related measures, which must assess the errors that a proposal makes, that is, the slot instances that are not correctly extracted or the pieces of information that are not correctly ignored; b) generalisation ability, which must assess the ability of a proposal to work well with as few documents as possible; c) and failure-related measures, which must assess the mistakes that are not due to a proposal itself, but its available implementation. Regarding efficiency, we suggest that both time- and memory-related measures should be used.

We have surveyed the literature, and we have found that there are a variety of performance measures that we have carefully analysed. Our conclusion is that the following ones fulfil our requirements and are then very appropriate in our context: the area under the ROC curve ($\mathsf{AUC\text{-}ROC}$) as the error-related measure, the performance knee (PK) as the generalisation measure, the failure ratio (FR) as the failure-related measure, plus learning time (LT) and extraction time (ET) as time-related measures, and learning memory (LM) and extraction memory (EM) as memory-related measures. Note that our proposal clearly deviates from other proposals in the literature, where precision- and recall-related measures are the standard, but we have proved that ours are more adequate in our context. In Sections §B.1 and §B.2, we justify our decision and provide enough details regarding each of the previous measures.

Set parameters. Regarding the parameters of our method, the researcher is requested to set α, which is the significance level at which statistical tests are performed, γ, which is the number of repetitions performed to create evaluation splits, ω, which is a vector with the relative weights of the performance measures according to the researcher's preferences, and η, which is a performance measure that we use to purge our experimental data.

In the literature, α is typically set to 0.05, which provides 95% statistical confidence. We suggest setting γ to a value in range 10–20 which generally leads to a sufficiently large number of sets of evaluation splits. The relative weight of the performance measures are completely up to the researcher. Regarding η, our suggestion is to use the area under the ROC curve, since our survey of the literature reveals that this is the most appropriate measure in our context, cf. Section §B.1.

```
method $computeEvaluationSplits(datasets)$
  $result = \emptyset$
  for each dataset $ds$ in $datasets$ do
    $h = \lfloor \frac{|ds|}{2} \rfloor$
    repeat $\gamma$ times
      $r_1$ = select $h$ documents from $ds$
      $r_2$ = select $h$ documents from $ds \setminus r_1$
      $ls_0 = \emptyset$
      $ts_{h+1} = r_2$
      $d_2 = null$
      for $s = 1$ until $h$ do
        $d_1$ = select one document from $r_1 \setminus ls_{s-1}$
        $ls_s = ls_{s-1} \cup \{d_1\}$
        $ts_{h+1-s} = ts_{h+2-s} \setminus \{d_2\}$
        $d_2$ = select one document from $ts_{h+1-s}$
        $result = result \cup \{(ls_s, r_2), (\emptyset, ts_{h+1-s})\}$
      end
    end
  end
return $result$
```

Figure 4.2: *Method to compute evaluation splits.*

4.2.2 Step 2: create evaluation splits

Before using the selected datasets, we need to split them into evaluation splits, that is, pairs of learning and testing sets. Note that learning sets do not actually make sense for proposals that are based on heuristics, so we have to create evaluation splits that are specifically tailored to these proposals.

Our proposal is to use the sub-sampling method that is presented in Figure §4.2. It takes a collection of datasets as input and returns a set of evaluation splits, i.e., a set of tuples of the form (ls, ts) in which ls denotes a learning set and ts denotes a testing set. For every dataset, the method first sets h to half its size and then repeats the following steps γ times: it first creates two reservoirs of documents called r_1, which stores a random half of the documents in the corresponding dataset, and r_2, which stores the other half. Then, it iterates h times and updates a learning set that is initialised to an empty set and a testing set that is initialised to the documents in the second

```
method $\mathit{runExperiments}(\mathit{proposals}, \mathit{measures}, \mathit{splits})$
  $\mathit{result} = \emptyset$
  for each proposal $p$ in $\mathit{proposals}$ do
    for each evaluation split $s$ in $\mathit{splits}$ do
      if $s$ is appropriate for $p$ then
        $m$ = select non-derived measures from $\mathit{measures}$
        $d$ = execute $p$ on $s$ and compute measures $m$
        let $\mathit{result}(p, s) = d$
      end
    end
  end
return $\mathit{result}$
```

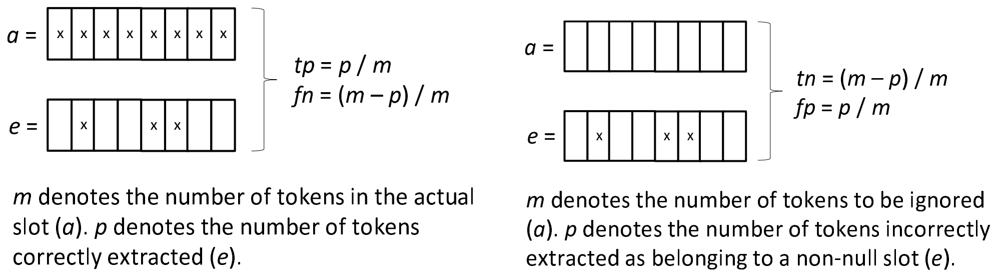
Figure 4.3: *Method to compute raw experimental data.*

reservoir. In each iteration, a random document from the first reservoir is selected and used to grow the previous learning set, as long as it has not been selected previously; furthermore, a document is removed from the previous testing set. The result that is returned by the method is updated with two tuples in each iteration: the first one is of the form (ls_s, r_2), and it is intended to be used as an evaluation set for rule-based proposals; note that the learning set grows in each iteration, but the testing set remains the same so that it is easy to evaluate how growing the learning set has an impact on the effectiveness of a proposal. The second one is of the form $(\emptyset, ts_{h+1-s})$; note that these evaluation splits are appropriate for heuristic-based proposals because they do not have a learning phase and consequently do not require a learning set.

4.2.3 Step 3: compute raw experimental data

Computing the raw experimental data consists in running every proposal on the appropriate evaluation splits, depending on whether they are rule-based or heuristic-based, and then collecting the non-derived performance measures that the researcher has selected.

Figure §4.3 shows the method that we propose to use. It works on a collection of proposals, a collection of measures, and a collection of evaluation splits; it computes a map called result in which pairs (p, s) of proposals and evaluation splits are associated with maps d that associate every performance measure with the value that was computed regarding proposal p on evaluation split s. In other words, the experimental data can be


a =
x x x x x x x x
e =
x x x
tp = p / m
fn = (m – p) / m
m denotes the number of tokens in the actual slot (a). p denotes the number of tokens correctly extracted (e).
a =
e =
x x x
tn = (m – p) / m
fp = p / m
m denotes the number of tokens to be ignored (a). p denotes the number of tokens incorrectly extracted as belonging to a non-null slot (e).

(a) Positive matching. (b) Negative matching.

Figure 4.4: *Cases when computing matchings.*

interpreted as a map from pairs of proposals and evaluation splits onto vectors with the corresponding values of the measures. Note that not every evaluation split is appropriate to run every proposal: rule-based proposals can work on evaluation splits that provide both a learning and a testing set, whereas heuristic-based proposals must be run on evaluation splits that provide a testing set only.

Our experience proves that there are cases in which it is not possible to compute the performance measures regarding a given proposal on a given evaluation split because there is a bug in the implementation or it simply takes too long or consumes too much memory and cannot be executed. Note that such failures must be recorded as missing values; such values shall later be used to compute failure-related derived measures.

The method to gather the experimental data is straightforward, but computing confusion matrices or dealing with unsupervised and heuristic-based proposals is, however, a little more involved. In the following subsections, we provide additional details.

Computing confusion matrices. The documents in the input datasets must be annotated; that is, a person must have labelled every piece of information to be extracted with a user-defined slot; the information that is not to be extracted is assumed to be implicitly labelled as belonging to a predefined null slot. Given a testing set, it is not difficult to compute the exact matchings, that is, the pieces of information that are correctly or incorrectly extracted as belonging to a given slot. The problem is how to compute inexact matchings. Such matchings are common, for instance, with proposals that work on DOM trees since DOM nodes are very likely not to be perfectly aligned with the information to be extracted.

To introduce our proposal, we assume that $S = \{s_1, s_2, \ldots, s_n\}$ denotes the set of slots on which we are working. Given a testing set, we need to compute n confusion matrices of the form $C_i = (tp_i, tn_i, fp_i, fn_i)$, where i ranges in the set of slots S, tp_i denotes the number of true positives for slot i, tn_i denotes the number of true negatives for slot i, fp_i denotes the number of false positives for slot i, and fn_i denotes the number of false negatives for slot i. To compute these matrices, it is necessary to analyse each slot in isolation and represent the documents as sequences of slot instances; note that given a slot s, its instances lead to positive matchings and the instances of the other slots, including the null slot, lead to negative matchings. Figure §4.4 illustrates both cases and how we propose to deal with them, namely:

Case 1: positive matching. In this case, the document provides an actual instance of slot s that has m tokens, p of which are extracted as belonging to that slot, whereas the remaining $m - p$ tokens are extracted as belonging to another slot or not extracted at all. In such a case, our proposal is to count the ratio of tokens that are correctly extracted as the number of true positives in this matching, that is, $tp = p/m$; similarly, the ratio of tokens that are not correctly extracted as belonging to slot s must be computed as false negatives, that is, $fn = (m - p)/m$. Note that if $m = p$, then it means that every token in the actual instance of the slot has been extracted correctly, in which case there is an exact matching that contributes with one true positive and zero false negatives, as expected. Note that if p is greater than m, that extra information extracted is computed as false positives when the corresponding negative matching is evaluated in Case 2.

Case 2: negative matching. In this case, the document does not provide an instance of slot s, that is, the slot at a given position is a slot different from s, possibly the null slot. Again, we can assume that slot has m actual tokens, that p such tokens are extracted as belonging to slot s, and that $m - p$ tokens are not extracted as belonging to slot s. In this case, our proposal is to count $(m - p)/m$ true negatives, since this is the ratio of tokens that have not been extracted as belonging to slot s, and to count p/m false positives, since this is the ratio of tokens that have been incorrectly extracted as belonging to slot s. Note that if p is greater than m, that extra information extracted is computed as true positives when the corresponding positive matching is evaluated in Case 1.

```
method $mapSlots(actual, extracted)$
  $result = \emptyset$
  for each slot $e$ in $extracted$ do
    $m = -\infty$
    for each slot $a$ in $actual$ do
      $n$ = compute $\eta$ on $a$ and $e$
      if $n > m$ then
        $m = n$
        $s$ = slot of $a$
      end
    end
    let $result(e) = s$
  end
return $result$
```

Figure 4.5: *Method to map extracted slots onto actual slots.*

Dealing with unsupervised and heuristic-based proposals. In the case of supervised proposals, it is not difficult to compute matchings because they learn extraction rules that are specifically tailored to extracting information as belonging to one of the slots that the user has defined in the input datasets. In the case of unsupervised or heuristic-based proposals, the problem is complicated by the fact that they ignore the annotations in the input datasets since they were devised to learn to extract as much information as possible from them, which is assigned to computer-generated slots. It is the user who must analyse the resulting computer-generated slots and map them onto user-defined slots, that is, she or he must assign a meaning to the computer-generated slots.

Since we are interested in an automated method, we have to perform the previous mapping automatically. Recall that we require the researcher who uses our method to decide on a so-called purging measure to which we refer to as η. Such measure is an effectiveness measure that is expected to provide a good overview of how good a proposal is and then helps purge the experimental data. We can use it to deal with unsupervised and heuristic-based proposals as shown in the method in Figure §4.5. This method gets a collection of actual slots, that is, the pieces of information and their corresponding labels as the user has provided them in a testing set, and a collection of extracted slots, that is, the pieces of information that an information extractor has returned when it was run on that testing set. The method then iterates

```
method $purgeData(data)$
  $proposals$ = get proposals in $data$
  $result = data$
  for each proposal $p$ in $proposals$ do
    $pk = computePerformanceKnee(data, p, \eta)$
    if $p$ is rule-based then
      $dataToRemove = \{(p, s, d) \mid \exists l, t \bullet s = (l, t) \land (p, s, d) \in data \land |l| \neq pk\}$
    else
      $dataToRemove = \{(p, s, d) \mid \exists t \bullet s = (\emptyset, t) \land (p, s, d) \in data \land |t| \neq pk\}$
    end
    $result = result \setminus dataToRemove$
    $t$ = compute per-extractor $\eta$ for $p$ using $result$
    if $t \leq$ minimum acceptable value of $\eta$ then
      $dataToRemove = \{(p, s, d) \mid (p, s, d) \in result\}$
      $result = result \setminus dataToRemove$
    end
  end
return $result$
```

Figure 4.6: *Method to purge experimental data.*

over every pair of actual and extracted slot and computes the purging measure η on them. It returns a map called $result$ in which each extracted slot is associated with the label of the actual slot with which the purging measure achieves its maximum value. In our experience, this mapping is as effective as a handcrafted-mapping, but it is completely automated.

4.2.4 Step 4: cook the experimental data

Cooking the experimental data consists in removing some of them so that the remaining ones can be used to compute the resulting rankings. First, the data must be purged, then derived measures must be computed, and, finally, the experimental data must be normalised. We provide additional details in the following subsections.

Purging data. In the previous step we have computed many experimental data. To perform as a fair comparison as possible, we have to compare the experimental data that corresponds to the best-performing evaluation splits which shall have a specific size, that is, we shall finally collect only the γ evaluation splits of the best-performing size, which can be different for each proposal. Thus, we have to remove the other splits as well as those that correspond to proposals that perform very bad. Recall that we require the researcher to set a purging measure η, which refers to the measure that our method uses to decide which data must be removed.

Figure §4.6 shows our method to purge the experimental data. It takes some experimental data as input and returns a subset of them. It iterates over the set of proposals in the experimental data and proceeds in two steps, namely: first, it removes the data that does not correspond to the best-performing evaluation splits, and then removes all of the data regarding a proposal if it performs very bad according to the purging measure.

The complex part of the first step is to compute the set of evaluation splits on which a proposal performs the best. We use the purging measure to compute a performance knee, that is an inflection point above which a proposal does not improve as the size of the evaluation splits increases. Computing a performance knee is not straightforward; we provide additional details on the method that we have devised in Section §B.1. Once the performance knee is computed, all of the data that does not correspond to evaluation splits whose size is equal to the performance knee can be purged. Note that computing the size of an evaluation split depends on the proposal: if it is a rule-based proposal, then the size of the evaluation split is computed as the size of the corresponding learning set; if it is a heuristic-based proposal, then it is computed as the size of the corresponding testing set. Note, too, that our method only keeps γ evaluation splits for a proposal that is not purged, all of which are the size of the best-performing evaluation splits found for that proposal.

The second step removes all of the data that correspond to proposals that are very bad. These are the proposals that do not achieve a value for the purging measure above a minimum acceptable value when it is computed on a per-extractor level. Recall that our suggestion is to use the area under the ROC curve as the purging measure; it is well-known that in cases in which a proposal does not achieve at least a 0.50 value for this measure, it performs worse than a random guess [78], which we consider bad enough to discard it.

Computing derived data. The data that we have got from the experimentation are computed on a per-evaluation-split basis. That is, a proposal is run on an evaluation split and the corresponding performance measures are computed. There are some measures that cannot be computed that way, but must be derived from the experimental data once they are purged.

Figure §4.7 presents the method that we propose to compute the derived measures. It works on some purged experimental data and returns a map in

```
method $computeDerivedData(data)$
  $proposals$ = get proposals in $data$
  $measures$ = get measures in $data$
  $result = \emptyset$
  for each proposal $p$ in $proposals$ do
    $m$ = select derived measures from $measures$
    $d$ = compute measures $m$ from $p$ and $data$
    let $result(p, null) = d$
  end
return $result$
```

Figure 4.7: *Method to compute derived data.*

which every pair of the form $(p, null)$ is associated with another map d; p is a proposal and $null$ denotes that the evaluation was not performed on a particular evaluation split, but derived from the existing ones; d is a map in which every derived measure is associated with its corresponding value.

Previously, we mentioned that our proposal is to compute the performance knee (PK) and the failure ratio (FR) as derived measures. They both can be easily computed on the purged data, namely: computing the performance knee is straightforward since we actually computed it in the previous step and discarded the evaluation splits with different sizes, so we only have to see what the size of the remaining evaluation splits is; computing the failure ratio amounts to counting the number of missing values in the input data and calculating the ratio to the total number of values.

Normalising the experimental data. Unfortunately, the performance measures do not range within the same intervals and their goodness are different, namely: the area under the ROC curve (AUC-ROC) ranges between 0.00 and 1.00 and the greater the better (recall that proposals whose AUC-ROC is equal or less than 0.50 are discarded when the experimental data are purged); the performance knee (PK) ranges between 1 and an arbitrarily large number and the smaller the better; the failure ratio (FR) ranges between 0.00 and 1.00 and the smaller the better; the learning time (LT) and the extraction time (ET) range between 0.00 CPU seconds and an arbitrarily large number and the smaller the better; finally, the learning memory (LM) and the extraction memory (EM) range between 0.00 GiB and an arbitrarily large number and the smaller the better.

```
method $\mathrm{normaliseData}(\mathrm{data})$
  $\mathrm{proposals}$ = get proposals in $\mathrm{data}$
  $\mathrm{measures}$ = get measures in $\mathrm{data}$
  $\mathrm{result}$ = $\emptyset$
  for each proposal $p$ in $\mathrm{proposals}$ do
    for each measure $m$ in $\mathrm{measures}$ do
      $W$ = $\{w \mid \exists s, d \bullet (p, s, d) \in \mathrm{data} \wedge w = d(m)\}$
      $(a, b)$ = $(\min W, \max W)$
      if $m$ must be maximised then
        $W'$ = $\{w' \mid \exists s, d \bullet (p, s, d) \in \mathrm{data} \wedge w' = (d(m) - a) \operatorname{div} (b - a)\}$
      else
        $W'$ = $\{w' \mid \exists s, d \bullet (p, s, d) \in \mathrm{data} \wedge w' = 1.00 - (d(m) - a) \operatorname{div} (b - a)\}$
      end
      let $\mathrm{result}(p, m)$ = $W'$
    end
  end
return $\mathrm{result}$
```

Figure 4.8: *Method to normalise experimental data.*

Figure §4.8 presents the method that we propose to normalise the experimental data within range 0.00 .. 1.00, so that the lower bound corresponds to bad values and the upper bound corresponds to good values. That transformation can be performed easily since it amounts to translating the range of each performance measure and then computing its complement if that measure needs to be minimised. Note that this method works on the purged experimental data, which is a map in which each pair of proposal p and evaluation split s is associated with a map d that associates every performance measure with the value that was computed regarding proposal p on evaluation split s. The method to normalise the data transforms them into a new map in which each pair of proposal p and measure m is associated with the set W' of normalised values of that measure regarding that proposal. (In the pseudo-code, x div y equals x/y if $y \neq 0.00$; otherwise, it equals 1.00.)

4.2.5 Step 5: compute rankings

The next-to-last step of our method consists in computing the final results, which consists of a number of local rankings and a global ranking. In the following subsections, we provide additional details on the methods that we propose.

```
method $computeLocalRankings(data)$
    $proposals$ = get proposals in $data$
    $measures$ = get measures in $data$
    $result = \emptyset$
    for each measure $m$ in $measures$ do
        $H_0 = \emptyset$
        $H_1 = \emptyset$
        for each pair of proposals $p_1, p_2$ in $proposals$ such that $p_1 \prec p_2$ do
            $p\text{-}value = Wilcoxon\text{-}Rank\text{-}Sum\text{-}Test(data(p_1, m), data(p_2, m))$
            $k = |proposals|$
            $n = (k^2 - k)/2$
            if $p\text{-}value \geq \alpha/n$ then
                $H_0 = H_0 \cup \{(m, p_1, p_2)\}$
            else
                $H_1 = H_1 \cup \{(m, p_1, p_2)\}$
            end
        end
        $H$ = transform $H_0$ and $H_1$ into a total pre-order
        $result = result \cup H$
    end
return $result$
```

Figure 4.9: *Method to compute local rankings.*

Computing local rankings. Figure §4.9 presents our method to compute the local rankings. It works on the normalised experimental data and returns a map in which each measure is associated with an ordered collection of proposals. It compares every pair of proposals regarding every measure using Wilcoxon's Rank-Sum test [153]. Note that only pairs of different proposals are compared and that the order in which they are compared is irrelevant; in the pseudo-code, we assume that $\prec$ denotes an arbitrary ordering of the proposals, e.g., the lexicographic ordering. The test returns a p-value that, according to Bonferroni's correction, has to be compared to the statistical significance level α set by the researcher divided by the number of comparisons to be performed, which is $(k^2 - k)/2$, where k denotes the number of proposals to be compared. In the case of derived measures, the experimental data provide only a value; in such cases Wilcoxon's Rank-Sum trivially returns 0.00 if the measures to be compared have different values, and 1.00 if the values are the same. Both sets H_0 and H_1 store triplets of the form (m, p_1, p_2); the triplets in H_0 denote the pairs of proposals for which the ranking data do not provide enough evidence to conclude that they behave differently regarding the performance measure; the triplets in H_1 denote the remaining ones.

Unfortunately, the previous procedure does not necessarily result in a total pre-order. Generally speaking, such situations occur when there is a minimal sequence of proposals $\langle p_1, p_2, \ldots, p_n \rangle$ such that Wilcoxon's Rank-Sum test does not find enough evidence to conclude that p_i behaves differently from p_{i+1} for every $i = 1 \mathinner{..} n-1$, but it finds enough evidence to conclude that p_1 behaves differently from p_n. Our proposal to transform such chains into total pre-orders is to break them assuming that p_j does not behave like p_{j+1}, where p_j and p_{j+1} ($1 \leq j < n$) denote the pair of proposals for which Wilcoxon's Rank-Sum test returns the smallest p-value above the significance level α; in other words, we suggest selecting the couple of proposals for which the experimental data provide more evidence that they behave differently. There is obviously a chance to make a mistake, but it is the only way to transform the results of Wilcoxon's Rank-Sum test into a total pre-order. Note that the decision might be taken arbitrarily at any other point in the chain and the results would be the same: there is only a chance to make a mistake at the point where the chain is arbitrarily broken.

Computing a global ranking. When we devised TANGO and ROLLER, we had to assess many different configurations in order to find the most effective and efficient. We obviously were interested in making as an objective decision as possible, which led to the heuristic that we describe in Section §B.3.1. The idea is to map each configuration or proposal onto a single scalar value that assesses how good it is from the perspective of a number of weighted performance measures.

Figure §4.10 presents the method that we propose to compute the global ranking. It iterates twice over the set of pairs of proposals and measures. In the first iteration, it computes a map called $\mathtt{mdr}$ that maps every pair of proposal p and measure m onto its corresponding mean-to-deviation ratio. In the second iteration, it computes the resulting ranks, which are referred to as K, and stores them in map $\mathtt{result}$.

4.2.6 Step 6: produce a report

The last step of our method consists in producing a report in which the results of the previous steps are summarised and commented by the researcher. Below, we present a suggestion regarding how to organise it.

```
method $\mathrm{computeGlobalRanking}(data)$
  $proposals$ = get proposals in $data$
  $measures$ = get measures in $data$
  for each proposal $p$ in $proposals$ do
    for each measure $m$ in $measures$ do
      $W = \bigcup\{V \mid (p, m, V) \in data\}$
      $(\mu, \sigma) = (\mathrm{mean} W, \mathrm{stdev} W)$
      if $\sigma \neq 0.00$ then
        let $mdr(p, m) = \mu^2/\sigma$
      else
        let $mdr(p, m) = \mu$
      end
    end
  end
  for each proposal $p$ in $proposals$ do
    let $result(p) = 0$
    for each measure $m$ in $measures$ do
      $a = \max_{q \in proposals} mdr(q, m)$
      $K = mdr(p, m)/a$
      let $result(p) = result(p) + \omega_m K$
    end
  end
return $result$
```

Figure 4.10: *Method to compute a global ranking.*

Abstract. As usual, the abstract must provide a short overview of the report and highlight the original findings.

Experimental environment. The goal of this section is to provide an overview of the experimental environment. Our suggestion is to organise it as follows:

Hardware and software. Regarding the hardware, we suggest that the researcher should report on the processors, the motherboard, the memory, the persistent storage, and whether it was virtual or bare metal. Regarding the software, we suggest that the researcher should report on the operating system, the virtual machines and the libraries used, if applicable. She or he should also report on the changes that were conducted to customise the default configurations, if any.

Proposals. For each proposal, the report should list its name, key references to the literature, a classification (that is, whether it is rule-based or heuristic-based, supervised or unsupervised in the case of rule-based proposals, free-text or semi-structured, and open or closed), the implementation used in the experiments, and some comments that may help the reader understand key facts regarding it.

Performance measures. We suggest that the researcher should organise the measures in categories since our experience proves that this usually helps understand their relative importance better. For each measure, the report should list its name, whether it is derived or not, its definition, the interval in which it ranges, its goodness (that is, whether the goal is to minimise or to maximise it), and its relative weight (ω). The report should also make it clear what the selected purging measure is (η) and provide a justification regarding the relative weights that measure their relative importance. (Please, recall that we have made some suggestions regarding the most appropriate measures, but our method is open to accommodate new measures as they are proven to be adequate in our context.)

Datasets. For each dataset, the report should list its name and version, the web site from which it was downloaded, the number of documents that it provides, and how large they are in average. We suggest that the datasets should be grouped in categories according to their topic and that the researcher should list the slots that were extracted in each category.

Statistics. The researcher must report on the significance level that she or he selected (α) and the number of repetitions set in the method to compute the evaluation splits (γ).

Experimental data. This section must report on the experimental data that was computed from the experiments. Our suggestion is to organise it as follows:

Non-derived measures. Note that the amount of data regarding these measures is typically huge. Including them in the report makes little sense and would be of little interest, since they are too many data for a person to understand them. It is, however, interesting to try to learn from these data how a proposal behaves in practice regarding the non-derived performance measures. Our suggestion is to provide charts and tables regarding the mean values of the measures, which may provide a rough intuition regarding how a proposal behaves, and then use the least squares regression method to compute the approximation that maximises the determination coefficient R^2 [124]. Recall that it is not usual at all that the papers in which new information extractors are introduced report on their theoretical complexity, so we think that this is a good approximation, and it is very important to practitioners. The researcher should comment on how the conceptual innovations in each proposal are reflected on the results. She or he should, however, avoid comparing the results to each other, since they just provide a rough approximation to how each proposal behaves. Note that the points in the charts and tables are averaged from many data, and such values do not take the distribution of values into account; as a conclusion, comparing them might lead to wrong conclusions that cannot be supported from a statistical point of view.

Derived measures. We suggest that the report should present them in a table or a chart and comment on their values from a conceptual point of view. Our proposal is to compute the performance knee and the failure ratio as derived measures. The researcher should reflect on the experimental results and try to discern the conceptual reason why a proposal has a lower performance knee than the others. Furthermore, she or he should also reflect on the reasons why the failure ratio of a proposal is not zero; it is very important to discern if the failures were due to an intrinsic feature of a proposal or a bug in its implementation.

Purged proposals. If a proposal was removed because it did not achieve a value for the purging measure above the minimum allowable threshold, then the researcher should comment on the conceptual reasons why that happened.

Rankings. This section must report on the rankings computed by our method. Our suggestion is to organise it as follows:

Local rankings. We suggest that the report should present them in a table in which the empirical rankings should be listed, and then the p-values computed by Wilcoxon's Rank-Sum test on every pair of proposals regarding every performance measure; the table should also report on the statistical ranking computed using the method that we have proposed.

Global ranking. We suggest that the report should present the global ranking in a table and a chart. The researcher should comment on the results and provide a conceptual explanation.

Name	References	Classification	Implementation
P0	-	Rule-based (supervised), semi-structured, closed	Java 7
P1	-	Heuristic-based, semi-structured, closed	Java 7
P2	-	Rule-based (unsupervised), semi-structured, closed	Java 7
P3	-	Rule-based (supervised), semi-structured, closed	Java 7
P4	-	Rule-based (supervised), semi-structured, closed	Java 7

Table 4.1: *Proposals analysed in our case study.*

Conclusions. The report should include a section in which the researcher summarises her or his conclusions from conducting the experimental study, evaluating, and comparing the proposals that she or he selected.

Bibliography. The report should include references to the literature where further information on the proposals, the datasets, or other key issues can be found.

4.3 A case study

In this section, we present one of the many case studies that we have conducted to polish our proposal. Below, we present the corresponding report.

Abstract. In this case study, we have evaluated, compared, and ranked five proposals to which we refer to as P0, P1, P2, P3, and P4. We keep them anonymous because it is not our intention to contribute with a ranking of some existing proposals, but to illustrate how our method works in practice so that it can serve as a guideline for other researchers. Our study clearly reveals that analysing the experimental data intuitively can very easily lead to wrong conclusions that are not supported from a statistical point of view.

Experimental environment. Next, we report on the experimental environment that we used in our study.

Hardware and software. The experiments were run on a virtual computer that was equipped with four Intel Xeon E7-4807 cores that ran at 1.87 GHz, had 4 GiB of RAM, and 16 GiB of persistent storage. The

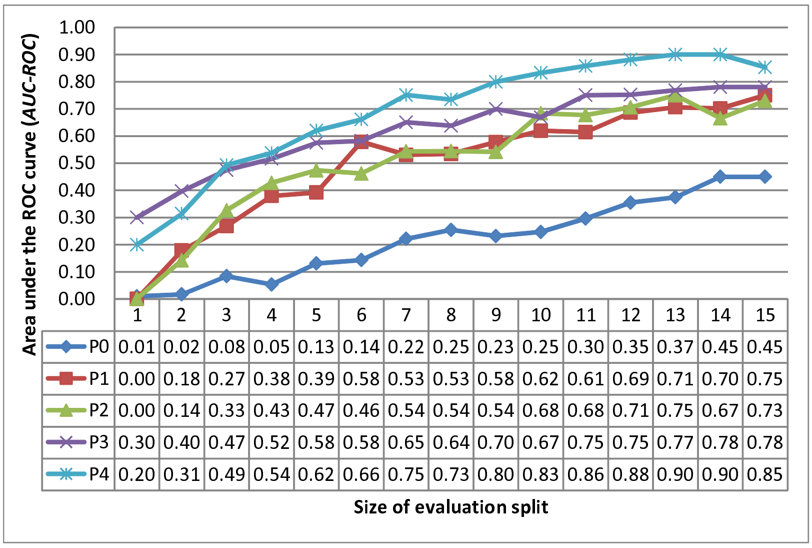
		Effectiveness measures (Weight = 70%)		
Name	**Goodness**	**Interval**	**Derived**	**Weight**
Area under the ROC curve (*AUC-ROC*)	Maximise	0.00..1.00	No	50%
Performance knee (*PK*)	Minimise	1..∞	Yes	25%
Failure ratio (*FR*)	Minimise	0.00..1.00	Yes	25%
		Learning efficiency measures (Weight = 10%)		
Name	**Goodness**	**Interval**	**Derived**	**Weight**
Learning time (*LT*)	Minimise	0.00..∞	No	50%
Learning memory (*LM*)	Minimise	0.00..∞	No	50%
		Extraction efficiency measures (Weight = 20%)		
Name	**Goodness**	**Interval**	**Derived**	**Weight**
Extraction time (E*T*)	Minimise	0.00..∞	No	50%
Extraction memory (E*M*)	Minimise	0.00..∞	No	50%

AUC-ROC is the purging measure.

Table 4.2: *Performance measures used in our case study.*

motherboard was a Supermicro X8QB6. All of the proposals were run using the Oracle Java Development Kit 1.7.9_02. The operating system was Microsoft Windows 7 Pro 64-bit. The regular expression engine required by some proposals was provided by GNU RegEx 1.1.4. No changes to the default configuration of the hardware or the software were made.

Proposals. Table §4.1 summarises the proposals that we have studied. They all work on semi-structured documents and are closed, but differ significantly regarding the techniques on which they rely, namely: P0 refers to a very simple baseline that uses rules of the form L-R, where L and R are 5-token disjunctive patterns that match the left and the right of the information that has been annotated in the learning sets; P1 is a heuristic based proposal that compares a number of documents to find the differences amongst them, which are returned as the extracted information; P2 is a rule-based, unsupervised proposal that also finds differences amongst a number of documents and generalises them into a regular expression with variables that capture the differences; P3 is a hybrid proposal that first learns the structure of the information using an automata and then learns transition conditions using a standard machine-learning technique; and P4 is a proposal that learns DOM-based extraction rules using a propositional inductive logic technique.



	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
P0	0.01	0.02	0.08	0.05	0.13	0.14	0.22	0.25	0.23	0.25	0.30	0.35	0.37	0.45	0.45
P1	0.00	0.18	0.27	0.38	0.39	0.58	0.53	0.53	0.58	0.62	0.61	0.69	0.71	0.70	0.75
P2	0.00	0.14	0.33	0.43	0.47	0.46	0.54	0.54	0.54	0.68	0.68	0.71	0.75	0.67	0.73
P3	0.30	0.40	0.47	0.52	0.58	0.58	0.65	0.64	0.70	0.67	0.75	0.75	0.77	0.78	0.78
P4	0.20	0.31	0.49	0.54	0.62	0.66	0.75	0.73	0.80	0.83	0.86	0.88	0.90	0.90	0.85

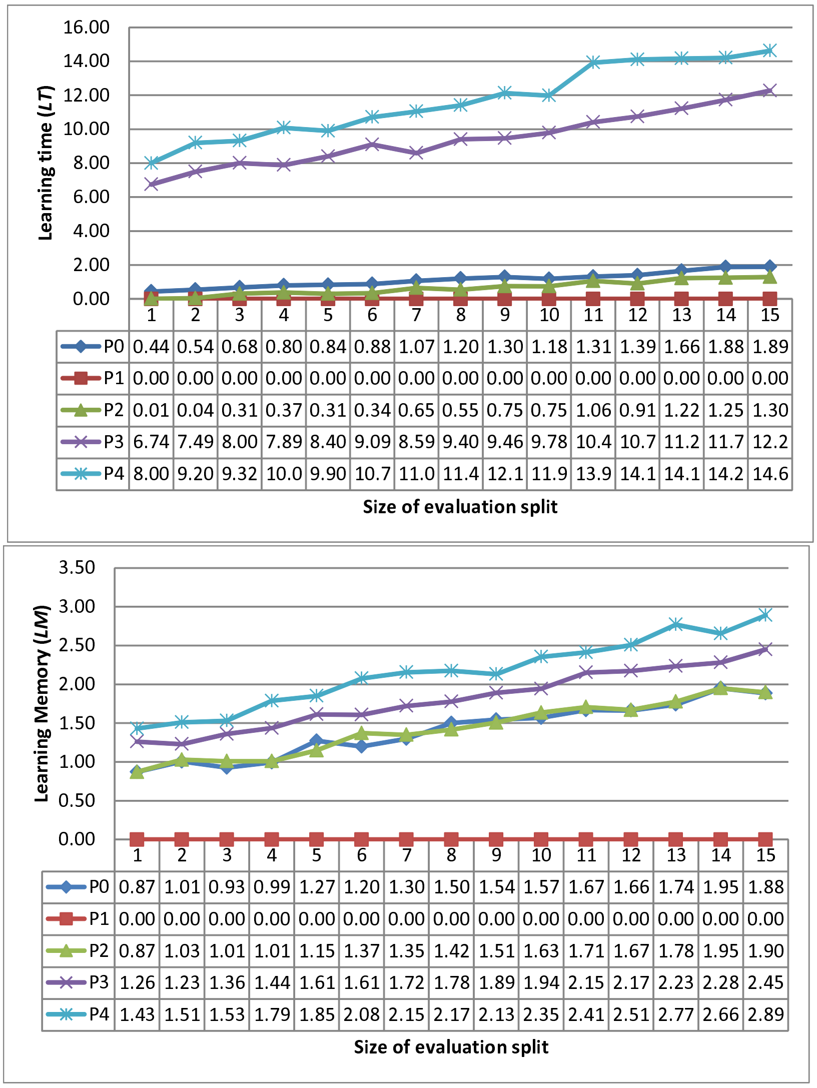
Table 4.3: *Non-derived effectiveness measures computed in our case study.*

Performance measures. Table §4.2 summarises the performance measures that we have used. We have adhered to the suggestions that we have made in the previous section; additional details are provided in Sections §B.1 and §B.2. Note that we have grouped the measures into effectiveness measures, whose relative weight is 70%, learning efficiency measures, whose relative weight is 10%, and extraction efficiency measures, whose relative weight is 20%. These figures reflect our opinion that it is very important that a proposal must produce rule sets that are very good at extracting the information of interest as quickly as possible; the efficiency regarding learning is not as important because our experience proves that, nowadays, the time or the memory required to learn a rule set does not actually make a difference from a practical point of view, although they are important and should not be neglected.

Datasets. Table §A.1 summarises the datasets that we have used. We selected 38 datasets from Sleiman and Corchuelo's repository [162], from each of which we selected 30 documents at random.

Statistics. We set the significance level to $\alpha = 0.05$ and the number of repetitions in the method to compute the evaluation splits to $\gamma = 10$.

Experimental data. Next, we report on the measures that we collected from running our experiments.



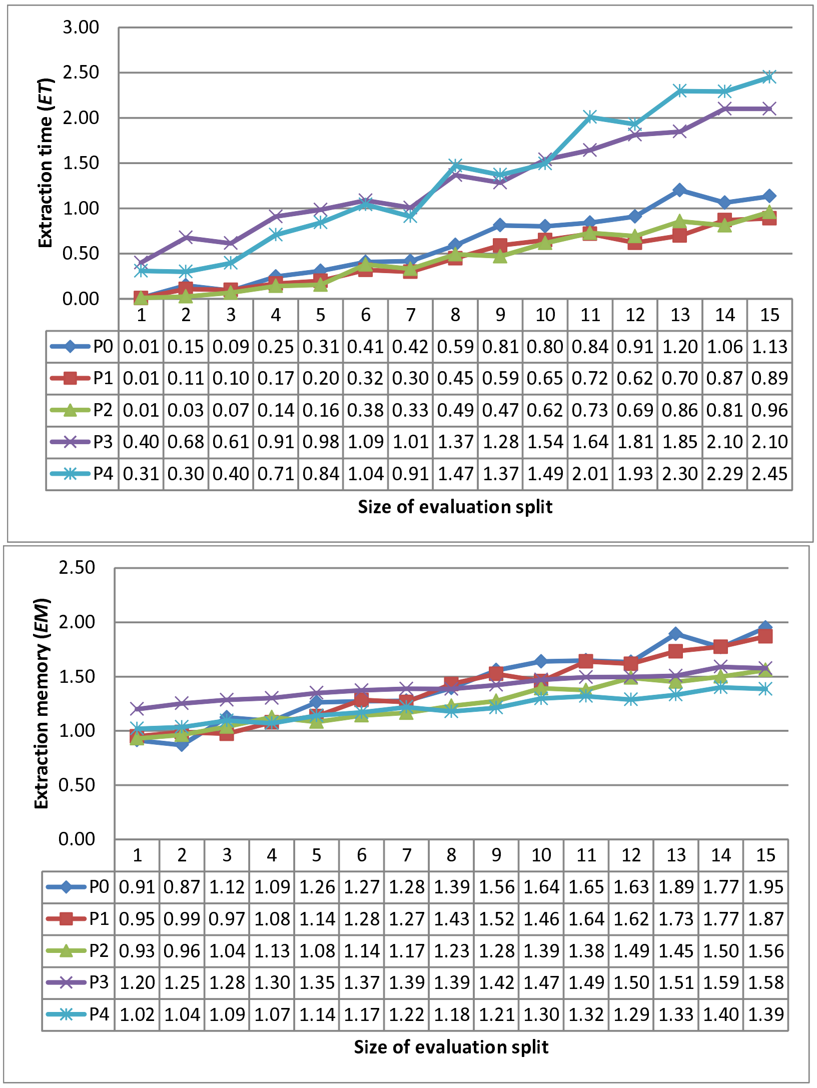
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
P0	0.44	0.54	0.68	0.80	0.84	0.88	1.07	1.20	1.30	1.18	1.31	1.39	1.66	1.88	1.89
P1	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
P2	0.01	0.04	0.31	0.37	0.31	0.34	0.65	0.55	0.75	0.75	1.06	0.91	1.22	1.25	1.30
P3	6.74	7.49	8.00	7.89	8.40	9.09	8.59	9.40	9.46	9.78	10.4	10.7	11.2	11.7	12.2
P4	8.00	9.20	9.32	10.0	9.90	10.7	11.0	11.4	12.1	11.9	13.9	14.1	14.1	14.2	14.6

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
P0	0.87	1.01	0.93	0.99	1.27	1.20	1.30	1.50	1.54	1.57	1.67	1.66	1.74	1.95	1.88
P1	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
P2	0.87	1.03	1.01	1.01	1.15	1.37	1.35	1.42	1.51	1.63	1.71	1.67	1.78	1.95	1.90
P3	1.26	1.23	1.36	1.44	1.61	1.61	1.72	1.78	1.89	1.94	2.15	2.17	2.23	2.28	2.45
P4	1.43	1.51	1.53	1.79	1.85	2.08	2.15	2.17	2.13	2.35	2.41	2.51	2.77	2.66	2.89

Table 4.4: *Non-derived learning-related efficiency measures computed in our case study.*

Non-derived measures. Tables §4.3, §4.4, and §4.5 report on the mean values that we gathered regarding the non-derived measures (before purging them) and Table §4.6 presents the best approximations that we have found.

- Regarding the area under the ROC curve, note that all of the proposals seem to behave logarithmically with respect to the size of

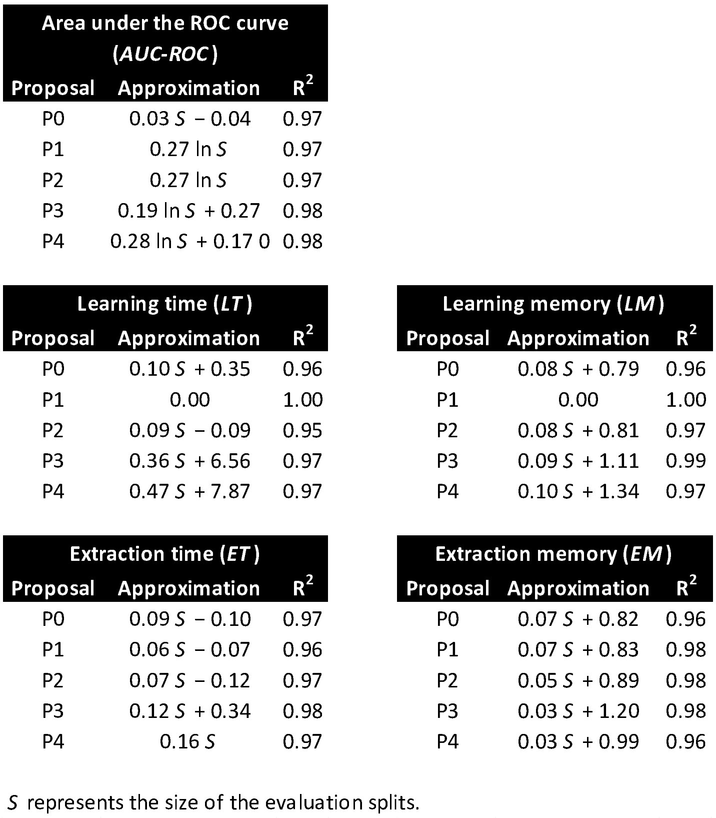


	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
P0	0.01	0.15	0.09	0.25	0.31	0.41	0.42	0.59	0.81	0.80	0.84	0.91	1.20	1.06	1.13
P1	0.01	0.11	0.10	0.17	0.20	0.32	0.30	0.45	0.59	0.65	0.72	0.62	0.70	0.87	0.89
P2	0.01	0.03	0.07	0.14	0.16	0.38	0.33	0.49	0.47	0.62	0.73	0.69	0.86	0.81	0.96
P3	0.40	0.68	0.61	0.91	0.98	1.09	1.01	1.37	1.28	1.54	1.64	1.81	1.85	2.10	2.10
P4	0.31	0.30	0.40	0.71	0.84	1.04	0.91	1.47	1.37	1.49	2.01	1.93	2.30	2.29	2.45

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
P0	0.91	0.87	1.12	1.09	1.26	1.27	1.28	1.39	1.56	1.64	1.65	1.63	1.89	1.77	1.95
P1	0.95	0.99	0.97	1.08	1.14	1.28	1.27	1.43	1.52	1.46	1.64	1.62	1.73	1.77	1.87
P2	0.93	0.96	1.04	1.13	1.08	1.14	1.17	1.23	1.28	1.39	1.38	1.49	1.45	1.50	1.56
P3	1.20	1.25	1.28	1.30	1.35	1.37	1.39	1.39	1.42	1.47	1.49	1.50	1.51	1.59	1.58
P4	1.02	1.04	1.09	1.07	1.14	1.17	1.22	1.18	1.21	1.30	1.32	1.29	1.33	1.40	1.39

Table 4.5: *Non-derived extraction-related efficiency measures computed in our case study.*

the evaluation splits (S), except for P0, which seems to behave linearly. Regarding the proposals that learn rules, the logarithmic behaviour was expected because the larger an evaluation split, the more learning documents are available, which increases the chances to learn extraction rules that are more general and effective. The behaviour of P0 is linear because the technique on which



Area under the ROC curve (*AUC-ROC*)		
Proposal	Approximation	R^2
P0	$0.03 S - 0.04$	0.97
P1	$0.27 \ln S$	0.97
P2	$0.27 \ln S$	0.97
P3	$0.19 \ln S + 0.27$	0.98
P4	$0.28 \ln S + 0.17$ 0	0.98

Learning time (*LT*)		
Proposal	Approximation	R^2
P0	$0.10 S + 0.35$	0.96
P1	0.00	1.00
P2	$0.09 S - 0.09$	0.95
P3	$0.36 S + 6.56$	0.97
P4	$0.47 S + 7.87$	0.97

Learning memory (*LM*)		
Proposal	Approximation	R^2
P0	$0.08 S + 0.79$	0.96
P1	0.00	1.00
P2	$0.08 S + 0.81$	0.97
P3	$0.09 S + 1.11$	0.99
P4	$0.10 S + 1.34$	0.97

Extraction time (*ET*)		
Proposal	Approximation	R^2
P0	$0.09 S - 0.10$	0.97
P1	$0.06 S - 0.07$	0.96
P2	$0.07 S - 0.12$	0.97
P3	$0.12 S + 0.34$	0.98
P4	$0.16 S$	0.97

Extraction memory (*EM*)		
Proposal	Approximation	R^2
P0	$0.07 S + 0.82$	0.96
P1	$0.07 S + 0.83$	0.98
P2	$0.05 S + 0.89$	0.98
P3	$0.03 S + 1.20$	0.98
P4	$0.03 S + 0.99$	0.96

S represents the size of the evaluation splits.

Table 4.6: *Best approximations of non-derived measures computed in our case study.*

it relies does not actually attempt to learn a rule set that can generalise the features of the information to be extracted; the more learning documents, the more patterns are available, but the technique is far too naive and roughly can extract information from documents that are very similar to the learning documents; as a conclusion, it is not surprising that it behaves linearly, with a very small slope. Before concluding, we would like to highlight that proposal P0 cannot achieve a value for AUC-ROC greater than 0.50, which means that it behaves worse than a random guess and can then be removed from our study.

- Regarding the learning time and the extraction time, the proposals behave linearly with small slopes, which confirms that they are very scalable. Furthermore, the learning times range from a

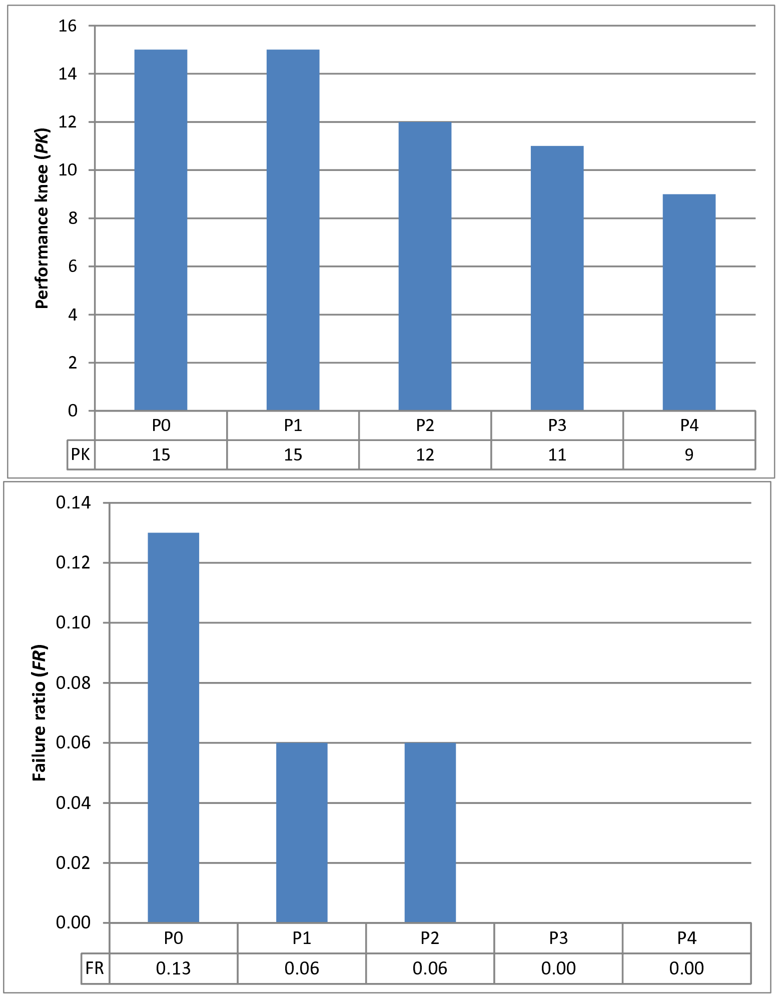


Table 4.7: *Derived measures computed in our case study.*

few milliseconds to quarter a minute and the extraction times range from a few milliseconds to a few seconds, which is reasonable in this context. Note that proposal P1 is based on heuristics, so it does not have a learning phase; thus the learning times are 0.00 seconds in every case.

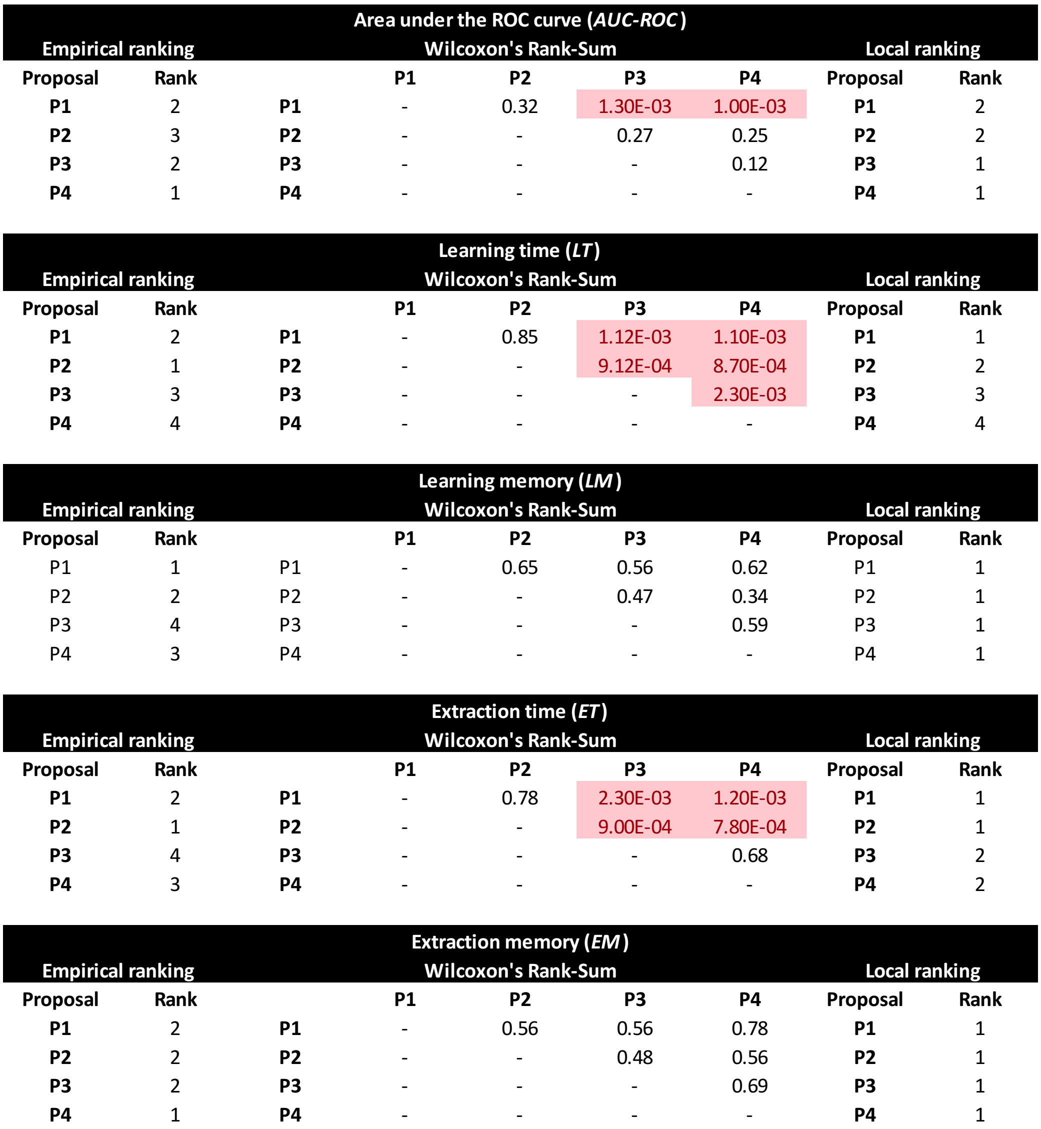
- Regarding the learning and extraction memory, they also seem to require an amount of memory that evolves linearly as the size of the evaluation splits increases; this is again a good piece of news since it confirms that all of the proposals are scalable in practice.

Note that it is commonly required a little more memory to learn a rule set than to apply it, but the overall memory footprint seems very small in every case. Again, the learning memory required by P1 is 0.00 GiB in every case because it is a heuristic-based proposal.

Derived measures. Table §4.7 reports on the derived measures computed.

- Regarding the performance knee, note that both proposals P0 and P1 seem to require 15 documents so that they are able to achieve their best performance, which seems to be a clear indication that they might improve a little more if more documents were available in the evaluation splits; note, however, that 15 documents can be considered a large number, chiefly because it is necessary to annotate all of the information to be extracted so that the effectiveness measures can be computed. P2 seems to achieve its best performance with 12 documents, P3 with 11 documents, and P4 with only 9 documents. Recall that P0 does not actually attempt to generalise rule sets, but uses prefixes and suffixes verbatim; thus, the more documents available, the more chances that the extraction rule set captures enough sequences of tokens so that the technique can extract correct information from the testing sets. On the contrary, P1 is a heuristic-based proposal that finds differences amongst documents, so the more documents, the more variability and the easier to infer which information has to be extracted. Proposal P2 is similar in spirit to P1, since it also compares differences amongst documents, so it also requires a relatively high number of documents to achieve its best performance. Proposals P3 and P4, which are based on standard machine-learning techniques seem to be the best at producing general-enough rules from as few as 11 or 9 documents, respectively.

- Regarding the failure ratio, note that proposals P0, P1, and P2 have failed on some datasets. P0 failed in 13.00% of the evaluation splits; after working this issue out, we found that the problem was the library that it uses to implement regular expressions, which did not work well with expressions of the form $\alpha|\alpha\beta$; the library implements regular expressions using a fixed-lookahead descending parser, which means that there are situations in which a sequence of tokens that matches a regular expression is not recognised as such. P1 and P2 failed in 6.00% of the evaluation splits because they cannot work on a single document, so there were many evaluation splits on which they could not work.



Area under the ROC curve (*AUC-ROC*)								
Empirical ranking		Wilcoxon's Rank-Sum					Local ranking	
Proposal	Rank		P1	P2	P3	P4	Proposal	Rank
P1	2	P1	-	0.32	1.30E-03	1.00E-03	P1	2
P2	3	P2	-	-	0.27	0.25	P2	2
P3	2	P3	-	-	-	0.12	P3	1
P4	1	P4	-	-	-	-	P4	1

Learning time (*LT*)								
Empirical ranking		Wilcoxon's Rank-Sum					Local ranking	
Proposal	Rank		P1	P2	P3	P4	Proposal	Rank
P1	2	P1	-	0.85	1.12E-03	1.10E-03	P1	1
P2	1	P2	-	-	9.12E-04	8.70E-04	P2	2
P3	3	P3	-	-	-	2.30E-03	P3	3
P4	4	P4	-	-	-	-	P4	4

Learning memory (*LM*)								
Empirical ranking		Wilcoxon's Rank-Sum					Local ranking	
Proposal	Rank		P1	P2	P3	P4	Proposal	Rank
P1	1	P1	-	0.65	0.56	0.62	P1	1
P2	2	P2	-	-	0.47	0.34	P2	1
P3	4	P3	-	-	-	0.59	P3	1
P4	3	P4	-	-	-	-	P4	1

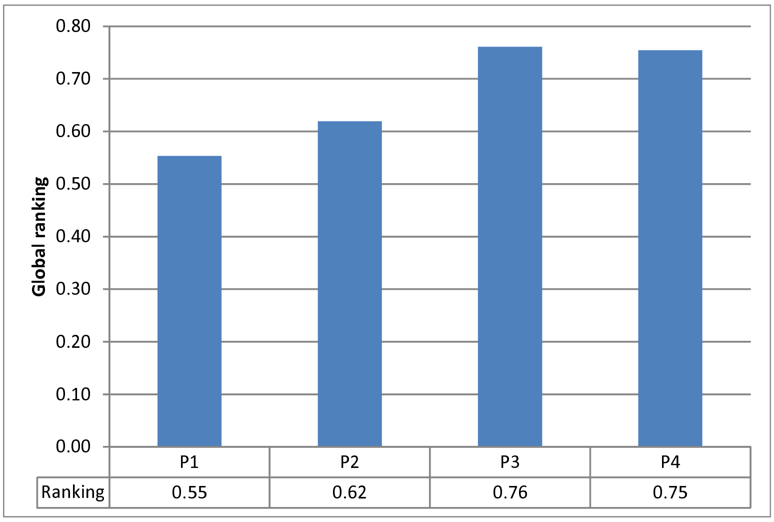
Extraction time (*ET*)								
Empirical ranking		Wilcoxon's Rank-Sum					Local ranking	
Proposal	Rank		P1	P2	P3	P4	Proposal	Rank
P1	2	P1	-	0.78	2.30E-03	1.20E-03	P1	1
P2	1	P2	-	-	9.00E-04	7.80E-04	P2	1
P3	4	P3	-	-	-	0.68	P3	2
P4	3	P4	-	-	-	-	P4	2

Extraction memory (*EM*)								
Empirical ranking		Wilcoxon's Rank-Sum					Local ranking	
Proposal	Rank		P1	P2	P3	P4	Proposal	Rank
P1	2	P1	-	0.56	0.56	0.78	P1	1
P2	2	P2	-	-	0.48	0.56	P2	1
P3	2	P3	-	-	-	0.69	P3	1
P4	1	P4	-	-	-	-	P4	1

Table 4.8: *Local rankings computed in our case study.*

Rankings. Next, we report on the local rankings and the global ranking that we have computed.

Local rankings. Table §4.8 reports on the local rankings that we have computed. The first column reports on the empirical ranks, which only require to average the values of the corresponding measures on the



	P1	P2	P3	P4
Ranking	0.55	0.62	0.76	0.75

Table 4.9: *Global ranking computed in our case study.*

normalised data. Then come the p-values that were computed using Wilcoxon's Rank-Sum test. Since we set the confidence level to its standard value $\alpha = 0.05$ and we have to compare 4 proposals, that means that we have to perform 6 comparisons on the same data. In other words, the decision boundary for the test is $\alpha/6 = 8.33\,10^{-3}$. In the table, we have highlighted the p-values that are below this decision boundary and thus indicate that there is enough evidence in the experimental data to conclude that the difference in rank amongst two given proposals is statistically significant. The last column reports on the resulting statistical rank. Note that Wilcoxon's Rank-Sum test does not lead to a total pre-order in the case of the area under the ROC curve: according to the test, there is not enough evidence to conclude that P1 behaves differently from P2; neither is there enough evidence to conclude that P2 behaves differently from P3; but there is enough evidence to conclude that P1 behaves differently from P3. To transform this into a total pre-order, we decided to break the chain at the comparison between P2 and P3, since this is the comparison for which Wilcoxon's Rank-Sum test returns the smallest p-value above the significance level, that is, the couple of proposals for which the data provide more evidence that they behave differently. The result is that we rank P3 and P4 in a group and P1 and P2 in a different group.

Global ranking. Table §4.9 reports on the global ranking that we have computed. According to the weights that we have assigned to the performance measures, the best-performing proposal is P3, which is closely followed by P4, and then come P2 and P1. This result is not surprising at all because the local rankings make a clear difference between P3 and P4 and the other proposals regarding effectiveness. Although there is a clear difference regarding learning time, too, they all range within a few seconds, which does not make an actual difference according to our preferences. There is also a difference regarding the extraction time, but not large enough to compensate for the superior effectiveness of P3 and P4.

Conclusions. In this case study, we have evaluated and compared five proposals in the literature.

P0 was a simple baseline, and it did not prove to be competitive enough with regard to the other proposals. It was removed from the comparison because it could not achieve an area under the ROC curve better than 0.50, which means that it performs worse than a random guess.

The best performing proposal was P3, which is a hybrid attempt to leverage standard machine-learning techniques that has proven to learn rule sets that are very effective and efficient. This proves that the idea of using such techniques, which has not been explored too much in the literature, is very promising. P4, which is based on inductive logic programming, ranks very close to P3, which is also an indication that trying to leverage standard machine-learning techniques is a good idea.

P1 and P2 rank at the bottom. None of them requires the user to provide an annotated learning set and they do their best at finding the differences amongst the documents on which they work, which is very likely a superset of the information to be extracted. However, their inability to take the user knowledge into account, has a clear impact on their performance.

Bibliography. We do not provide any bibliography references since we decided to keep the proposals anonymous.

4.4 Related work

In the following subsections, we first summarise the proposals that we have found in the literature; then we discuss on their key features and how they address the key questions that we have identified regarding a good ranking method; in every case we make a point of highlighting how our method advances the state of the art.

4.4.1 Overview of related proposals

We have found many informal methods in the general literature on information extraction, plus a few formal ones. In this section, we summarise our key findings regarding them.

The literature on information extractors. We have surveyed roughly one hundred proposals on web information extraction [1, 2, 4–8, 11, 13, 14, 18–27, 29, 31, 32, 35–40, 48, 49, 52, 53, 55, 61–64, 72, 73, 75, 76, 82, 83, 85, 86, 88, 91, 93, 94, 97, 103, 105–107, 116–123, 127, 128, 130, 134, 136–139, 141, 144, 145, 148, 150–152, 154, 155, 157–164, 167, 169, 171, 172, 174, 178, 180, 182, 185–187]. Our conclusion is that they provide a foundation and some guidelines to evaluate and compare information extraction proposals, but not formal methods that have the key features or address the key questions that we have identified. Obviously, their focus was not to provide such a method, but to support the idea that the new proposals that they introduced were better than others in the literature.

There are a few proposals that are a little surprising because they do not report on any experimental results [1, 7, 14, 37, 76, 127, 141, 150] or report on very few [11, 94], which does not contribute at all to drawing solid conclusions. Most of the remaining proposals provide enough empirical results, which helps support the conclusions better, but the methods used to evaluate and compare them were not solid enough.

Regarding the experimental environment, only a few proposals have paid attention to describing the hardware and the software used in the evaluation process [39, 154, 160, 162]. Regarding comparing the results, it was surprising that many proposals were not compared to others at all, but to some variations of themselves that resulted from changing the values of their configuration parameters [2, 5, 18, 19, 26, 29, 48, 49, 73, 82, 85, 91, 106, 120, 128, 144, 145, 151, 155, 163]. It was also surprising that not many proposals were evaluated on at least 20 datasets, which is the minimum that we recommend [5, 6, 20, 26, 72, 73, 82, 85, 119, 134, 137, 154, 162, 167]; a few other proposals were evaluated on 10–20 datasets, which still amounts to a significant number of experiments [13, 29, 39, 88, 118]; the others were evaluated on less than 10 datasets, which we do not think is acceptable to draw conclusions.

Regarding effectiveness, all of the proposals report on precision, recall, and the F_1 score. Only a few report on the learning curves as a means to assess their ability to learn good extraction rules from as few documents as possible [8, 21, 85, 86, 138, 164, 167]. A few ones, reported on the minimum number of documents that they require to learn effective extraction rules [35, 49, 82, 134, 152]. Unfortunately, very few proposals report on efficiency measures [8, 13, 24, 29, 39, 49, 88, 93, 97, 120, 154, 162].

It is not commonly clear how the evaluation splits were created, since the procedure to create them is not mentioned at all; in some cases, it is unclear if the evaluation sets were different from the learning sets. According to the few cases in which this information is provided, it seems that the favourite method is 10-fold cross validation [21, 97, 138, 164] or a variant [97, 155, 167]. A few authors also used repeated random splits in which the documents available were randomly selected as learning or evaluation documents multiple times. In many cases, the partition was 50%–50% [31, 55, 62, 64, 148, 154], but there are cases in which the learning set was smaller than the evaluation set [19, 23, 24, 26, 29, 73, 88, 134] and vice versa [2, 35, 36, 85, 86, 138, 151, 163, 164]. Summing up, most proposals used testing sets that were not larger than the corresponding learning sets. A few proposals used a single random split, that is, did not repeat the procedure multiple times [2, 19, 23, 24, 36, 73, 86, 88, 151, 154]. Many proposals that work on free-text documents used the official testing sets that were released at the MUC conferences [8, 32, 128, 144, 145, 164].

Regarding how the experimental data are used, it seems that every of the previous proposals analyses the data themselves, without cooking them. Furthermore, the results are analysed from a statistical point of view in very few cases [160, 162] which makes it difficult to assess if the differences found amongst a number of proposals are statistically significant or not. Finally, no global rankings are computed; the proposals are compared according to different measures in isolation, but no attempt is made to compute a global ranking.

The literature on formal methods. We have also surveyed the few existing formal ranking methods. Lehnert and Sundheim [115], Chinchor and others [33], and Hirschman [81] range amongst the first authors who worked on this topic. They worked in the context of the well-known MUC conferences, whose focus was on extracting information from free-text documents; they published a number of datasets so that the proposals that were presented at these conferences could be evaluated using a semi-automatic software tool that computed precision, recall, over-generation, and fallout [45]. They proposed to analyse these measures in isolation with the help of tables and charts; the only exception was recall and precision, which could be analysed together since they can be easily combined thanks to the well-known F_1 score. They proposed to use an approximate randomisation method to find groups of proposals that rank equally or differently according to the values of the performance measures [135]; this method is not intended to produce a ranking automatically, but to help a researcher handcraft a set of per-measure rankings by analysing the corresponding tables and charts.

Lavelli and others [112] criticised the previous work and highlighted some common mistakes that authors make when they evaluate and compare their proposals. Their work was extended and updated by Ireson and others [87] and Lavelli and others [113], who reported on the conclusions from The Pascal Network of Excellence. They provided a repository that was composed of 1 100 annotated free-text documents that were intended to evaluate different systems as homogeneously as possible. They also explored some new ideas regarding experimentation, namely: studying how brittle a proposal is by using test documents that are sampled from a time frame that is different from the time frame used to collect the learning documents; studying the impact of 4-fold cross validation on the performance results; studying the learning curve, that is, how a proposal behaves as new documents are available in the learning sets; analysing active learning strategies, that is, studying how adding new documents to a learning set using a given heuristic may have an impact on the learning curve; and studying how enriching a learning set with data that come from unannotated documents may have an impact on the results. The proposals were evaluated on the basis of precision, recall, and the F_1 score using a version of the software tool used in the MUC conferences [45]. They proposed to use the same approximate randomisation method as in the MUC conferences to help produce the resulting rankings; they also proposed to use the bootstrap method to compute confidence intervals for every performance measure in an attempt not to draw conclusions from their raw mean values, but to take the effects of randomness into account. Note that the statistical tests are not intended to produce a ranking automatically, but to help a researcher handcraft a number of per-measure rankings.

4.4.2 Key features

The papers that introduce a new web information extractor do not provide enough details regarding whether the informal methods that they use are automated or not. We implicitly assume that the authors had some automated support to run their experiments and to compute the performance measures, but we do not think that the methods can be automated because they mostly rely on a researcher commenting on the experimental data and producing the resulting rankings. Neither are they open, since most of them compute precision, recall, and the F_1 score, or agnostic, since they were devised in the context of a specific proposal. Obviously, the authors' goal was not to devise a ranking method, just to evaluate their proposals and to provide some evidence that they could beat others in the literature.

The formal methods that we have surveyed are also supported by tools to compute the performance measures. However, they cannot be considered automated methods since they just provide some guidelines to help a researcher elaborate on the results of the experiments. They propose to use some statistical tests that help compute the rankings, but they require a researcher to interpret some tables and charts. Neither are these methods open, since they commit to using a number of performance measures and provide specific guidelines to produce per-measure rankings. Furthermore, all of the methods focus on ranking supervised free-text proposals, which leaves out many other proposals in the literature.

The conclusion seems to be that the available informal or formal methods in the literature can be considered guidelines that are intended to help researchers produce per-measure rankings. We have managed to devise a method that is automated, which reduces the bias introduced by the researcher, open, since it can accommodate new performance measures as they are published and proven to be appropriate in our context, and agnostic, since it can be applied to any kind of proposal.

4.4.3 Setting up the experimental environment

The papers on web information extraction do not generally put an emphasis on describing the hardware or the software. They generally provide a list with the datasets that were used, but few other details are presented. They commonly commit to precision, recall, and the F_1 scores as effectiveness measures; unfortunately, almost none of them reports on efficiency measures.

The specific ranking methods do not put an emphasis on describing the hardware or the software. They all are accompanied with collections of datasets that were devised by a community of researcher. They focus almost exclusively on precision- and recall-based measures; efficiency measures are not taken into account.

The common theme in the literature seems to be that describing the hardware and the software is usually paid very little attention and that the performance measures used just provide a partial view of how good an information extractor is.

Regarding the description of the hardware and the software, we think that it is very important that a good ranking provides a good description, chiefly if efficiency measures are taken into account. Clearly, it is not the same to run an experimentation on a mid-class computer than running it on a super-computing facility. Thus, our method puts an emphasis on the researcher describing the hardware and the software.

Regarding the performance measures, it is surprising that most of the rankings in the literature focus exclusively on effectiveness measures, and do not provide a clue on efficiency measures. Our method proposes to use both kinds of measures since, otherwise, the resulting rankings would not provide an overall picture of how a proposal performs. We propose to use a small set of orthogonal measures, that is, measures that focus on different performance issues and are complementary to each other. In our survey of the literature, we have also found that precision, recall, and the F_1 score are the most common measures, but, unfortunately, it has also revealed that they are not the most appropriate in our context. The reason is that precision and, therefore, the F_1 score are skewed by unbalanced datasets, which are very common in our context; note that the information to extract is a small fraction of the information that a typical document provides (which includes field tags, menus, advertisements, navigation bars, copyright messages, and the like) and it is also common that some slots are optional or multi-valued, which also contributes to unbalancing the datasets.

In our method, we have carefully studied the effectiveness measures in the literature, and we have selected the ones that are most appropriate in our context, cf. Sections §B.1 and §B.2. Note, however, that the method itself is not bound with these particular measures; it is open to accommodate new measures that might appear in the literature and prove to be appropriate in our context.

4.4.4 Computing raw experimental data

The papers on web information extraction do not generally put an emphasis on explaining how the experimental data are computed. The specific ranking methods rely on software tools that set a standard format for the data and allow to compute the performance measures automatically, but the user is allowed to make some corrections interactively.

The effectiveness measures in the literature [165] are commonly computed from confusion matrices that record the number of true positives, true negatives, false positives, and false negatives that are computed in a testing set regarding each slot; such per-slot measures must later be combined into per-extractor measures. Confusion matrices are well-known in the literature, but there are some issues that make computing them difficult in our context.

The first issue is regarding how matchings are computed. A correct matching happens when a piece of information that actually belongs to a slot is extracted as belonging to that slot. Intuitively, correct matchings should be exact, but this interpretation is very restrictive in practice. It is common that an information extractor produces inexact matchings, that is, that it extracts a part of the information of interest or some spurious information; this is particularly true in the case of extractors that work on DOM trees, since the text contained in a DOM node is not usually aligned with the information that is expected to be extracted. The papers that introduce new information extraction proposals do not report on how matchings are dealt with. Regarding the formal methods, the idea of inexact matching was first introduced by Chinchor and others [33]; later, Lavelli and others [113] emphasised that the way that matchings are computed may have an impact on the results of an evaluation and then on the final ranking, but they did not elaborate more on this issue. Chinchor and others [33] used a simple approach in which they compute confusion matrices using exact matchings, but record the number of partial and incorrect matchings; then, they compute their effectiveness measures using customised formulae. Their approach is very simple, because they count a partial matching as half a true positive and an incorrect matching as one false positive and one false negative. Unfortunately, they reported on problems to compute true negatives in the case of multi-valued slots; consequently, they had trouble to compute the effectiveness measures that depend on this count. They had to resort to an interactive post-processing phase in which a user could mend the measures that were computed automatically, thus increasing the chances to introduce a bias in the results. We think that how matchings are computed must be taken into account when computing confusion matrices, since, otherwise, they do not actually reflect the effectiveness of an information extractor, but just provide an approximation. In VENICE, we have devised a method that basically takes into account the ratio of tokens that must have been extracted or discarded with regard to the tokens that have been extracted; it has proven to be both simple and effective at dealing with the problem of inexact matchings.

The second issue is regarding how effectiveness measures that are computed on a per-slot basis are generalised to per-extractor measures. In the literature, effectiveness measures generalised by using macro-, micro-, or weighted averages. Unfortunately, the authors have not commonly paid attention to the problem that micro- and weighted averages are skewed in the context of unbalanced datasets, which makes them of little interest in our context [57]. In VENICE we recommend using macro-averages because they are known not to be skewed in our context; furthermore, we suggest using the area under the ROC curve as an effectiveness measure and we have found an efficient means to extend it to a per-extractor level [77].

The third issue is regarding how to compute confusion matrices when evaluating an unsupervised or a heuristic-based proposal. If a proposal is supervised, then it is trained to return pieces of text as belonging to a specific user-defined slot; contrarily, if a proposal is unsupervised, then it learns to extract as much information as possible, which is automatically assigned to computer-generated slots; heuristic-based proposals do not learn extraction rules, but assign the information that they extract directly to computer-generated slots. The problem is how to map the computer-generated slots onto the appropriate user-defined slots so that confusion matrices can be computed. In the papers in which an unsupervised or a heuristic-based proposal has been evaluated, the authors have handcrafted these mappings, but this is a time consuming task, not to mention error-prone. In VENICE, we provide a specific automated method to compute confusion matrices for unsupervised and heuristic-based proposals.

4.4.5 Cooking the experimental data

Unfortunately, both the papers on web information extraction and the formal methods that we have found in the literature use the experimental data as they are gathered from running the experiments. However, we support the idea that the experimentation would be less biased and more stringent if each proposal was compared in its best experimental conditions. Otherwise, the rankings might be biased because a proposal might seem to perform better than another, but the latter might perform better if different evaluation splits were chosen. Furthermore, there are derived performance measures that cannot be computed on a per-evaluation-split basis, but have to be computed from the raw experimental data. It is important that the data be normalised, since, otherwise, the differences in range or deviation regarding some performance measures might have an impact on the resulting rankings. Note that unless the experimental data are normalised, it is very difficult that a ranking method can work with arbitrary sets of performance measures.

In our proposal, we purge the raw experimental data so as to remove the data that correspond to very bad proposals according to a given purging measure, and to remove the evaluation splits that do not correspond to the best performing-splits for each proposal. Simply put, we choose the smallest evaluation splits on which a proposal achieves its best effectiveness results. We look for those smallest evaluation splits because the less documents a set has, the less annotation effort is required (in supervised proposals) and the faster it is expected to work (generally speaking). This is the reason why we call them the best-performing splits. We also take into account that there can be performance measures that cannot be computed on a per-evaluation-set basis, but are derived from other measures that are computed on that basis. Finally, we normalise the data so that all of the measures range within the same interval and the interpretation of this interval is homogeneous.

4.4.6 Computing rankings

The informal ranking methods simply average the experimental data and use the results to rank the proposals. Since they usually focus on precision and recall, the final ranking can be computed in terms of the F_1 score, which combines them both. The problem with such rankings is that they do not take into account the deviations so it is not clear whether the results are skewed by the data distribution; that is, it is not clear if the differences in the mean measures are significant from a statistical point of view. There are only a few papers that perform a statistical analysis. Furthermore, the rankings regarding each measure are produced and studied in isolation; that is, no attempt to derive a global ranking is made.

All of the formal ranking methods use statistical tests to make sure that the resulting rankings are statistically sound. The problem is that they rely on tests that are computationally intensive and outdated; furthermore, there are many cases in which such tests do not lead to a total ranking, but the problem has not been studied further because these methods are not intended to be automated, but require a person to interpret the results and draw conclusions. Furthermore, none of the methods provides a means to compute a global ranking from the experimental data.

Our proposal also computes local rankings on a per-measure basis and it also makes sure that the results are statistically sound. The difference is that we rely on Wilcoxon's Rank-Sum test, which is efficient and there are two versions that are specifically adapted to small and large datasets. This is a non-parametric test because it does not assume that the experimental data have a pre-defined distribution and it works on non-paired samples, so that it can be applied to the experimental data computed from the best-performing evaluation splits of each proposal. Furthermore, we propose a method to compute a global ranking that relies on all of the performance measures, instead of studying them in isolation. It is novel in that we take both the researcher's preferences and a combination of the means and the deviations of the performance measures into account.

4.4.7 Reporting on the results

The informal methods that we have surveyed typically report on the experimental data and then provide some conceptual explanations. The formal methods provide some intuitive guidelines, but they do not make a proposal regarding how to write a report.

In our proposal, we have carefully studied how to organise such a report. Our emphasis was on organising it as effectively as possible and on providing the information that researchers need to understand how a proposal compares to others, without providing spurious information or information that is of little interest for practical purposes.

4.5 Summary

In this chapter, we have presented a method to rank web information extractors that overcomes the deficiencies that we have found in the literature. It is automated, so that researchers can bias the conclusions as little as possible, open, so that it can easily accommodate new performance measures, and agnostic, so that it can be applied to as many different kinds of proposals as possible. Furthermore, it addresses the following questions: how to set up the experimental environment, how to create appropriate evaluation splits, how to compute the experimental data, how to cook them, how to compute the rankings, and how to report on the results.

We have also analysed the performance measures that have typically been used in the literature and we have concluded that they are skewed in contexts in which the datasets are unbalanced, which are common in our context. We have made a recommendation regarding a set of performance measures that are appropriate in this context. The set of measures takes into account both the effectiveness and the efficiency of a proposal, it is small so that a researcher can easily decide on the relative weight of each measure, and the measures themselves are orthogonal, so that they provide a good overview of how a proposal performs. We have also supported the idea that each proposal must be compared regarding its best experimental conditions and we have reported on a method to find them that is based on computing a so-called performance knee.

Chapter 5
Conclusions

The information that the Web provides is usually buried into semi-structured web documents, which have become the standard for companies to provide catalogues of products and/or services. They are commonly generated using a template that specifies how the data that are retrieved from a back-end database to fulfil a user request is rendered in a user-friendly format. This makes it very difficult to extract that information so that it can be used to feed typical automated business processes. Web information extractors are intended to extract information from human-friendly web documents in a structured format that is amenable to feed automated business processes. We think that companies shall rely on an increasing number of such automated business processes at an ever-increasing pace, which shall require web information extractors to feed them with web information so that they can perform Business Intelligence. We also think that companies should benefit from an automated method that allows them to compare the existing web information extractors to find the most appropriate ones for a particular purpose.

We have tackled the problem of web information extraction from an inductive logic programming perspective and we have devised TANGO, which is a new proposal to learn web information extraction rules in the context of semi-structured web documents. It is able to learn very effective, general and expressive rules, but its learning process is very slow. It is a very flexible system and we have proven it by means of an exhaustive experimentation to configure it so that it performs the best as possible. Furthermore, it relies on a open catalogue of features that is decoupled from the learning process, which helps evolve it as the Web does.

We then decided to improve on TANGO's efficiency to learn rules. We came to the conclusion that we should have to resort to a propositio-relational approach and developed ROLLER. It is relational in that it is able to explore an unbounded neighbourhood of every node, but it is propositional since the rules are learnt by using propositional techniques. It is also a highly configurable proposal: it relies on an open catalogue of features, which helps adapt it as the Web evolves; furthermore, it does not commit to a specific propositional technique, but can leverage many proposals in the literature and thus benefit from the continuous advances in the general field of machine learning. The experiments confirm that our proposal is very effective and efficient in practice. It can outperform state-of-the art proposals in terms of effectiveness and it is very competitive in terms of efficiency; although it is a little more inefficient than others regarding learning times, it can still learn a rule in a matter of seconds, which we do not think is a serious shortcoming; the rules that it learns can, however, be executed as efficiently as the rules learnt by other state-of-the-art proposals.

Our results clearly support our idea that using standard machine-learning techniques to learn web information extraction rules is a promising approach. Note that this clearly deviates from the existing proposals in the literature, which build on ad-hoc machine-learning techniques that were specifically tailored to the problem of learning web information extraction rules. They have proved to be very effective, but the problem is that they tend to fade away because their learning components are not clearly differentiated, which makes it difficult to evolve them as the Web evolves and precludes re-using the many advances that are published in the general field of Machine Learning. Contrarily, both TANGO and ROLLER rely on standard machine-learning techniques, that are leveraged and configured for optimum performance, and an open catalogue of features, which can be easily replaced. This proves that it makes sense to keep working on trying to use general-purpose machine-learning techniques instead of working on new ad-hoc techniques.

Finally, we realised that there was a problem when comparing information extraction proposals. The papers in the literature use a variety of methods to prove that their approaches outperform the others, but our conclusion is that most of them use ad-hoc ranking methods that are not sufficiently specified and that, in some cases, have important deficiencies. Consequently, the existing web information extraction proposals have been ranked using quite heterogeneous methods, which makes comparing the results that have been published in the literature impossible.

In this dissertation, we have presented a method to rank web information extractors that overcomes the deficiencies that we have found in the literature. It is automated, so that researchers can bias the conclusions as little as possible, open, so that it can easily accommodate new performance measures, and agnostic, so that it can be applied to as many different kinds of proposals as possible. Furthermore, it addresses the following questions: how to set up the experimental environment, how to create appropriate evaluation splits, how to compute the experimental data, how to cook them, how to compute the rankings, and how to report on the results.

Regarding our future work, we would like to evolve towards Open Information Extraction, which is an area that remains unexplored in the context of semi-structured web documents. The underlying idea is to create learning sets with documents from several web sources and apply the rules learnt to new documents that belong to different web sites on the same topic. We started working on Open Information Extraction during our research visit to the Information Sciences Institute of the University of Southern California. There, we have been applying our techniques to quite a large repository of web documents on human trafficking, illicit gun trading, patent trolling, and autonomy research. Our proposal is to use TANGO or ROLLER on their datasets, but using a catalogue of features that are more general so that we can identify the commonalities of the relevant information across different web sites. Unfortunately, their heterogeneity prevents TANGO or ROLLER from achieving good results, so we are still working on a catalogue of general features that is more appropriate in this context. We have also found that our techniques being supervised might be a problem with such large repositories. We have also started working on a quite an unsupervised proposal that reduces the amount of human effort required: first, we need to apply unsupervised techniques to cluster the documents that are generated by the same server-side templates; then, we apply unsupervised techniques to extract the information from each cluster; finally, we discard the useless information and give semantics to the extracted information. We are trying to leverage some previous research results that were produced by our research group and the group at the Information Sciences Institute.

Summing up, assuming that our research hypothesis is accepted, we think that we have sufficiently proven our thesis. We hope that our results can effectively help companies reduce their integration costs by means of our new approaches to web information extraction. We also think that we have open up an interesting research path regarding Open Information Extraction.

Appendix A
The experimental environment

A.1 Hardware and software

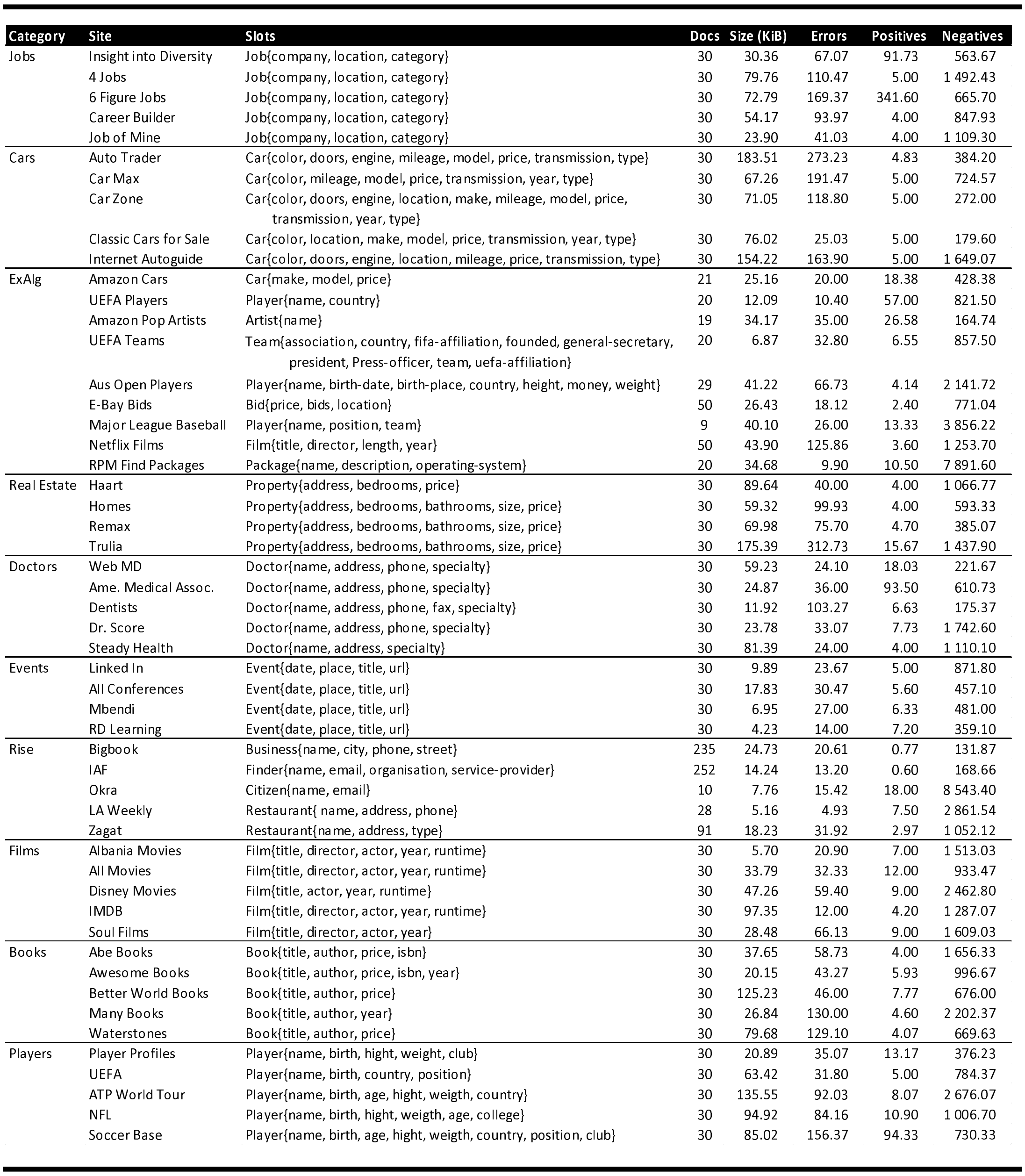
We performed our experiments on a virtual computer that was provided by our University cloud infrastructure. It is equipped with four Intel Xeon E7 4807 cores that run at 1.87 GHz, have 64 GiB of RAM, and 2 TiB of storage.

The operating system is Windows 7 Pro 64-bit and we used the following software packages: Oracle's Java Development Kit 1.7.9_02, Weka 3.6.8, JTidy 9.38, Jsoup 1.7.1, and Selenium 2.33.0.

No changes were made to the default configurations of the hardware or the software.

A.2 Repository of datasets

We used a collection 38 datasets on jobs, cars, real estate, doctors, events, films, books, and players, plus 9 datasets from the ExAlg repository and 5 datasets from the RISE repository that provide semi-structured documents. The categories regarding the first group of datasets were randomly sampled from The Open Directory sub-categories, and the web sites inside each category were randomly selected from the 100 best ranked web sites between December 2010 and March 2011 according to Google's search engine; we downloaded 30 documents from each web site and handcrafted a set of annotations with the slots that we wished to extract from each document. Table §A.1 describes our datasets; for each category, we report on the sites from which they were downloaded, the slots that model the information that they provide, the number of documents that they have, their average size in KiB, the average number of HTML errors that they have (as



Category	Site	Slots	Docs	Size (KiB)	Errors	Positives	Negatives
Jobs	Insight into Diversity	Job{company, location, category}	30	30.36	67.07	91.73	563.67
	4 Jobs	Job{company, location, category}	30	79.76	110.47	5.00	1 492.43
	6 Figure Jobs	Job{company, location, category}	30	72.79	169.37	341.60	665.70
	Career Builder	Job{company, location, category}	30	54.17	93.97	4.00	847.93
	Job of Mine	Job{company, location, category}	30	23.90	41.03	4.00	1 109.30
Cars	Auto Trader	Car{color, doors, engine, mileage, model, price, transmission, type}	30	183.51	273.23	4.83	384.20
	Car Max	Car{color, mileage, model, price, transmission, year, type}	30	67.26	191.47	5.00	724.57
	Car Zone	Car{color, doors, engine, location, make, mileage, model, price, transmission, year, type}	30	71.05	118.80	5.00	272.00
	Classic Cars for Sale	Car{color, location, make, model, price, transmission, year, type}	30	76.02	25.03	5.00	179.60
	Internet Autoguide	Car{color, doors, engine, location, mileage, price, transmission, type}	30	154.22	163.90	5.00	1 649.07
ExAlg	Amazon Cars	Car{make, model, price}	21	25.16	20.00	18.38	428.38
	UEFA Players	Player{name, country}	20	12.09	10.40	57.00	821.50
	Amazon Pop Artists	Artist{name}	19	34.17	35.00	26.58	164.74
	UEFA Teams	Team{association, country, fifa-affiliation, founded, general-secretary, president, Press-officer, team, uefa-affiliation}	20	6.87	32.80	6.55	857.50
	Aus Open Players	Player{name, birth-date, birth-place, country, height, money, weight}	29	41.22	66.73	4.14	2 141.72
	E-Bay Bids	Bid{price, bids, location}	50	26.43	18.12	2.40	771.04
	Major League Baseball	Player{name, position, team}	9	40.10	26.00	13.33	3 856.22
	Netflix Films	Film{title, director, length, year}	50	43.90	125.86	3.60	1 253.70
	RPM Find Packages	Package{name, description, operating-system}	20	34.68	9.90	10.50	7 891.60
Real Estate	Haart	Property{address, bedrooms, price}	30	89.64	40.00	4.00	1 066.77
	Homes	Property{address, bedrooms, bathrooms, size, price}	30	59.32	99.93	4.00	593.33
	Remax	Property{address, bedrooms, bathrooms, size, price}	30	69.98	75.70	4.70	385.07
	Trulia	Property{address, bedrooms, bathrooms, size, price}	30	175.39	312.73	15.67	1 437.90
Doctors	Web MD	Doctor{name, address, phone, specialty}	30	59.23	24.10	18.03	221.67
	Ame. Medical Assoc.	Doctor{name, address, phone, specialty}	30	24.87	36.00	93.50	610.73
	Dentists	Doctor{name, address, phone, fax, specialty}	30	11.92	103.27	6.63	175.37
	Dr. Score	Doctor{name, address, phone, specialty}	30	23.78	33.07	7.73	1 742.60
	Steady Health	Doctor{name, address, specialty}	30	81.39	24.00	4.00	1 110.10
Events	Linked In	Event{date, place, title, url}	30	9.89	23.67	5.00	871.80
	All Conferences	Event{date, place, title, url}	30	17.83	30.47	5.60	457.10
	Mbendi	Event{date, place, title, url}	30	6.95	27.00	6.33	481.00
	RD Learning	Event{date, place, title, url}	30	4.23	14.00	7.20	359.10
Rise	Bigbook	Business{name, city, phone, street}	235	24.73	20.61	0.77	131.87
	IAF	Finder{name, email, organisation, service-provider}	252	14.24	13.20	0.60	168.66
	Okra	Citizen{name, email}	10	7.76	15.42	18.00	8 543.40
	LA Weekly	Restaurant{ name, address, phone}	28	5.16	4.93	7.50	2 861.54
	Zagat	Restaurant{name, address, type}	91	18.23	31.92	2.97	1 052.12
Films	Albania Movies	Film{title, director, actor, year, runtime}	30	5.70	20.90	7.00	1 513.03
	All Movies	Film{title, director, actor, year, runtime}	30	33.79	32.33	12.00	933.47
	Disney Movies	Film{title, actor, year, runtime}	30	47.26	59.40	9.00	2 462.80
	IMDB	Film{title, director, actor, year, runtime}	30	97.35	12.00	4.20	1 287.07
	Soul Films	Film{title, director, actor, year}	30	28.48	66.13	9.00	1 609.03
Books	Abe Books	Book{title, author, price, isbn}	30	37.65	58.73	4.00	1 656.33
	Awesome Books	Book{title, author, price, isbn, year}	30	20.15	43.27	5.93	996.67
	Better World Books	Book{title, author, price}	30	125.23	46.00	7.77	676.00
	Many Books	Book{title, author, year}	30	26.84	130.00	4.60	2 202.37
	Waterstones	Book{title, author, price}	30	79.68	129.10	4.07	669.63
Players	Player Profiles	Player{name, birth, hight, weight, club}	30	20.89	35.07	13.17	376.23
	UEFA	Player{name, birth, country, position}	30	63.42	31.80	5.00	784.37
	ATP World Tour	Player{name, birth, age, hight, weigth, country}	30	135.55	92.03	8.07	2 676.07
	NFL	Player{name, birth, hight, weigth, age, college}	30	94.92	84.16	10.90	1 006.70
	Soccer Base	Player{name, birth, age, hight, weigth, country, position, club}	30	85.02	156.37	94.33	730.33

Table A.1: *Description of our datasets.*

beginsWithNumber
beginsWithParenthesis
beginsWithPunctuation
countOfAlphaNum
countOfBlanks
countOfCapitals
countOfDigits
countOfIntegers
countOfLetters
countOfLowercaseBigrams
countOfLowercaseTokens
countOfLowercaseTrigrams
countOfTokens
countOfTrigrams
countOfUppercaseBigrams
countOfUppercaseTokens
countOfUppercaseTokens
countOfUppercaseTrigrams
endsWithNumber
endsWithParenthesis
endsWithPunctuation
secondToken
firstToken
hasBlanks
hasBracketedAlphaNum
hasBracketedNumber
hasCurrencySymbol
hasQuestionMark
isAlphaNum
isBlank
isCapitalised
isCurrency
isDate
isEmail
isISBN
isLowerCase
isNumber
isPhone
isUppercase
isURL
isYear
penultimateToken
lastToken

(a) Some attributive features.

ancestor
children
firstSibling
lastSibling
leftSibling
parent
rightSibling

(b) Some relational features.

Table A.2: *Partial catalogue of user-defined features.*

reported by JTidy), the average number of positive examples, and the average number of negative examples. The datasets were split ten times; in each split, we randomly selected six documents for learning purposes and the remaining ones for testing purposes. The results on which we report were obviously computed on the testing sets.

A.3 Catalogue of features

The catalogue of features that we have used relies on the standard HTML features and the standard rendering features of the input documents, as they are defined in the corresponding W3C recommendations [80, 176]. Additionally, it also includes some user-defined features. Table §A.2 shows only the user-defined features that have proven to be useful in our experiments.

The user-defined attributive features can be classified according to the prefixes of their names into the following groups: a) prefix beginsWith identifies some features that check if the text of a node begins with a token that belongs to a given lexical class, e.g., a number or a punctuation symbol; b) prefix countOf identifies some features that count the number of tokens in the text of a node that fulfil a given property, e.g., the count of alpha-numeric tokens or the count of lowercase tokens; c) prefix endsWith denotes features that check if the text in a node ends with a token that belongs to a given lexical class, e.g., a number or a punctuation symbol; d) prefixes first and second denote features that return the first two tokens of the text in a node, i.e., the first bigram; e) prefix has identifies features that check if there is a subsequence of tokens in the text of a node that fulfils a given property, e.g., there is a bracketed number or a question mark; f) prefix is denotes a feature that checks if the text in a node matches a given pattern, e.g., whether it is capitalised or a phone number; g) finally, prefixes penultimate and last denote features that return the last two tokens of the text in a node, i.e., the last bigram.

The catalogue of relational features provides common features to navigate from a node to its neighbours in a DOM tree, namely: ancestor, children, first sibling, last sibling, left sibling, right sibling, and parent.

A.4 Other proposals

We searched the Web and contacted many authors in order to have access to the implementation of as many proposals as possible. We managed to find an implementation for SoftMealy [85] and Wien [107], which are classical proposals, and RoadRunner [40], FiVaTech [93], and Trinity [162], which are recent proposals. We also experimented with an approach that is based on Aleph [166]. Below, we provide additional details:

SofMealy [85]. It takes a collection of web documents and their corresponding annotations as input and learns an extraction rule that is a non-deterministic finite-state transducer. The states of the transducer indicate the slots to extract, the transitions account for the possible orderings of the slots in the input documents, and the conditions indicate when the extraction of an slot should start or end. Transition conditions are learnt using a token alignment and generalisation algorithm.

WIEN [107]. It takes a collection of annotated web documents as input and learns simple regular expressions that contain the delimiters of the information that should be extracted. These delimiters are the longest common prefix of characters, and the longest common suffix of characters for each type of slot.

RoadRunner [40]. It takes two or more web documents as input and tries to learn a union-free regular expression that describes them. It considers the first web document as a base template and then iterates through the other web documents; in each iteration, it compares the current web document with the base template using a string alignment algorithm, then collapses mismatches, and applies a backtracking algorithm to detect optional and repetitive patterns.

FiVaTech [93]. It takes one or more web documents as input and tries to learn the template that was used to generate them. It uses a clustering algorithm that applies a tree-edit distance to the DOM nodes of the input web documents; it then uses a matrix alignment algorithm to align the previous nodes on a per-cluster basis; then, it applies an algorithm to mine repetitive patterns in the aligned matrix, and finally, it applies some heuristics to detect optionality.

Trinity [162]. It works on two or more web documents. It finds and removes shared token sequences amongst them until finding the information that varies from document to document. It relies on the assumption that repetitive patterns are likely to belong to the template used to generate the web documents, and therefore, they only contain the non-relevant information that should be discarded. It starts with a collection of input web documents, and it then tries to find a shared pattern by using a sliding window of a given size. When a pattern is found, it splits the documents into three parts that contain the prefixes, the separators, and the suffixes into which the shared pattern partitions the initial documents. The algorithm is applied as many times as necessary until no more shared patterns are found.

Aleph [166]. Since our proposals work on datasets that can be very easily translated into first-order representations, this means that it is relatively easy to learn rules using general purpose inductive logic programming techniques. We have tried Aleph, which is a well-known proposal in the literature that is very effective and efficient in the context of classical machine learning problems. It relies on a bottom-up learning process, which first learns overly-specific rules and then tries to generalise and merge them.

Appendix B

Measuring performance

B.1 Effectiveness measures

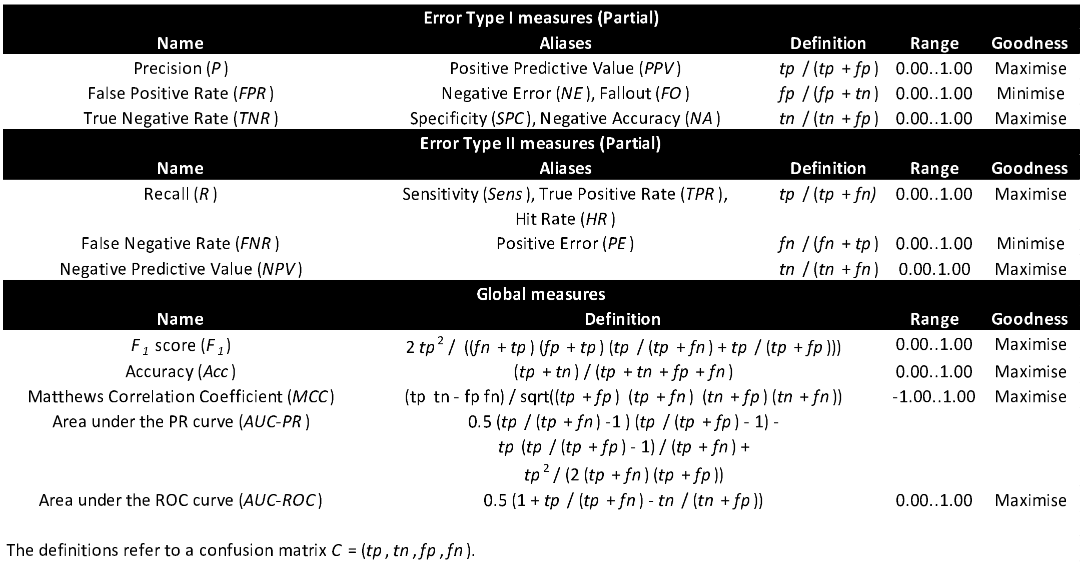
Our proposal regarding effectiveness measures is to classify them into error-related measures, generalisation-related measures, and failure-related measures. Next, we summarise our findings regarding them.

B.1.1 Error-related measures

We have surveyed the literature, and we have found many error-related measures [54, 165], cf. Table §B.1. Some of them are partial because they focus on either how good a proposal is at either extracting or ignoring slots; the others are global because they were designed to report on both abilities at the same time.

The partial measures can be further classified as follows: a) measures that assess the error type I (aka false alarms), that is, the number of pieces of information that are incorrectly extracted as belonging to a given slot or its complement; these measures include precision (P), the false positive rate (FPR), and the true negative rate (TNR); b) and measures that assess the error type II (aka misses), that is, the number of pieces of information that are not extracted as belonging to a given slot or its complement; these measures include recall (R), the false negative rate (FNR), and the negative predictive value (NPV).

The global measures that we have found are the following: the F_1 score (F_1), accuracy (Acc), the Matthews correlation coefficient (MCC), the area under the PR curve ($\mathsf{AUC\text{-}PR}$), and the area under the ROC curve ($\mathsf{AUC\text{-}ROC}$).



Error Type I measures (Partial)				
Name	**Aliases**	**Definition**	**Range**	**Goodness**
Precision (*P*)	Positive Predictive Value (*PPV*)	*tp* / (*tp* + *fp*)	0.00..1.00	Maximise
False Positive Rate (*FPR*)	Negative Error (*NE*), Fallout (*FO*)	*fp* / (*fp* + *tn*)	0.00..1.00	Minimise
True Negative Rate (*TNR*)	Specificity (*SPC*), Negative Accuracy (*NA*)	*tn* / (*tn* + *fp*)	0.00..1.00	Maximise

Error Type II measures (Partial)				
Name	**Aliases**	**Definition**	**Range**	**Goodness**
Recall (*R*)	Sensitivity (*Sens*), True Positive Rate (*TPR*), Hit Rate (*HR*)	*tp* / (*tp* + *fn*)	0.00..1.00	Maximise
False Negative Rate (*FNR*)	Positive Error (*PE*)	*fn* / (*fn* + *tp*)	0.00..1.00	Minimise
Negative Predictive Value (*NPV*)		*tn* / (*tn* + *fn*)	0.00.1.00	Maximise

Global measures			
Name	**Definition**	**Range**	**Goodness**
F_1 score (F_1)	2 tp^2 / ((*fn* + *tp*) (*fp* + *tp*) (*tp* / (*tp* + *fn*) + *tp* / (*tp* + *fp*)))	0.00..1.00	Maximise
Accuracy (*Acc*)	(*tp* + *tn*) / (*tp* + *tn* + *fp* + *fn*)	0.00..1.00	Maximise
Matthews Correlation Coefficient (*MCC*)	(tp tn - fp fn) / sqrt((*tp* + *fp*) (*tp* + *fn*) (*tn* + *fp*) (*tn* + *fn*))	-1.00..1.00	Maximise
Area under the PR curve (*AUC-PR*)	0.5 (*tp* / (*tp* + *fn*) -1) (*tp* / (*tp* + *fp*) - 1) - *tp* (*tp* / (*tp* + *fp*) - 1) / (*tp* + *fn*) + tp^2 / (2 (*tp* + *fn*) (*tp* + *fp*))		
Area under the ROC curve (*AUC-ROC*)	0.5 (1 + *tp* / (*tp* + *fn*) - *tn* / (*tn* + *fp*))	0.00..1.00	Maximise

The definitions refer to a confusion matrix *C* = (*tp*, *tn*, *fp*, *fn*).

Table B.1: *Common error related measures.*

```
method $\mathit{computePerformanceKnee}(\mathit{data}, \mathit{proposal}, \mathit{measure})$
  if $\mathit{proposal}$ is rule-based then
    $T = \{s \mid \exists l, t, d \bullet (\mathit{proposal}, (l, t), d) \in \mathit{data} \land s = |l|\}$
    $n = \max T$
    for each $i \in [1 \mathinner{..} n]$ do
      let $X_i = \{v \mid \exists l, t, d \bullet (\mathit{proposal}, (l, t), d) \in \mathit{data} \land |l| = i \land v = d(\mathit{measure})\}$
    end
  else
    $T = \{s \mid \exists l, t, d \bullet (\mathit{proposal}, (l, t), d) \in \mathit{data} \land s = |t|\}$
    $n = \max T$
    for each $i \in [1 \mathinner{..} n]$ do
      let $X_i = \{v \mid \exists l, t, d \bullet (\mathit{proposal}, (l, t), d) \in \mathit{data} \land |t| = i \land v = d(\mathit{measure})\}$
    end
  end
  let $r : \{1, 2, \ldots n\} \mapsto \{1, 2, \ldots, n\}$ be a permutation such that
    $\overline{X_{r(i)}} \leq \overline{X_{r(i+1)}}$ for every $i \in [1 \mathinner{..} n-1]$
  let $\mathit{result}$ be the smallest $r(i)$ such that
    $\mathit{Wilcoxon\text{-}Rank\text{-}Sum\text{-}Test}(X_{r(i)}, X_{r(n)}) \geq \alpha/(n-1)$
return $\mathit{result}$
```

Figure B.1: *Method to compute the performance knee.*

B.1.2 Generalisation-related measures

Regarding generalisation measures, our survey of the literature suggests that so-called learning curves should be used. Such curves display how the performance of a supervised proposal evolves as the learning set is grown from a relatively small set of documents up to an arbitrarily large set. There is typically a size of the learning set at which the performance achieves its maximum value and becomes stable; that size is a knee in the learning curve, that is, an inflection point that can be compared to others in order to assess how good a proposal is at generalising good extraction rules from a small set of input documents. Our proposal is to use this performance knee (PK) as a measure to assess the effort required to assemble the set of documents from which a proposal must learn an extraction rule set. Although the idea is conceptually simple, we have found two important problems, which we have addressed in our method.

The first problem is regarding heuristic-based proposals. They do not have a learning phase, so we can select the minimum number of documents that allows a proposal to work at its maximum performance as its corresponding performance knee.

The second problem is that we have not found any results regarding how to compute the performance knee; the results in the literature suggest that the learning curves be compared intuitively, which is not appropriate to devise an automated method. We have devised a method to compute the exact performance knee, which is presented in Figure §B.1. It gets some raw experimental data, a proposal, and a measure as input, and it returns the corresponding performance knee. The idea is to map the problem onto a statistical problem as follows: given a proposal, we create n new variables $X_1, X_2, \ldots, X_n$, where each X_i ranges over the values of the selected measure when it is computed on the evaluation splits of size i, where i ranges from 1 to n. Realise that these variables can be viewed as experimental samples of some unknown random variables. Note that prior to initialising variables X_i, our method needs to compute the set of evaluation split sizes in the experimental data, to which we refer to as T; n simply denotes the maximum evaluation split size. How the size is computed depends on whether the proposal being analysed is rule-based or heuristic-based: in the former case, it is computed as the size of the learning sets; in the latter case, it is computed as the size of the testing sets. After computing the X_i variables, we have to find a permutation r that ranks them according to their average value, in increasing order; in cases in which there are ties, we suggest that they should be broken by putting the variable that corresponds to the smallest evaluation split first. We then can apply the well-known Wilcoxon's Rank-Sum test [153] to find the first variable that is statistically indistinguishable from $X_{r(n)}$; in other words, to find the variable that corresponds to the smallest evaluation split on which the input proposal achieves a performance regarding the input measure that is statistically indistinguishable from the maximum. Given two samples of two random variables, this test computes a p-value that must be compared to $\alpha/(n-1)$, where α denotes the statistical significance level set by the researcher as a parameter of VENICE; note that we cannot compare it to α since we need to perform several tests on the same data, so it is necessary to apply Bonferroni's correction [153]; when the p-value is equal to or greater than the $(n-1)$-th part of the significance level, the variables are indistinguishable from a statistical point of view; if all of the variables are indistinguishable from $X_{r(n)}$, that means that we are in an exceptional case in which a technique performs the same in any situation, which is, obviously, not expected to be very frequent in practice.

Before concluding, we would like to emphasise that the method that we have proposed is generic, since it can compute the performance knee of an arbitrary measure and proposal. However, our study of the literature proves that the only measure that seems appropriate in our context is the area under the ROC curve. We have, however, decided to propose a generic method since many authors are working on new performance measures and VENICE is open to accommodate them as they are devised and proved to be appropriate in our context.

B.1.3 Failure-related measures

Regarding the failure-related measures, our survey of the literature reveals that they have not been paid attention. Authors have basically ignored that the implementations are far from perfect and, thus, may fail, which we think is very important from a practitioner's point of view.

Our proposal is to use a measure called failure ratio (FR), which is defined as follows:

$$FR = \frac{F}{D},$$

where F denotes the number of evaluation splits on which an alternative or proposal did not work, and D denotes the number of evaluation splits on which the alternative was run. Intuitively, the closer to 0.00 the better and the closer to 1.00 the worse.

B.2 Efficiency measures

Unfortunately, efficiency measures have not been paid much attention in the literature. Almost no author reports on them, but we think that they are very important to provide an actual overall picture of how a proposal performs in practice. They are of uttermost importance to practitioners who have to make a decision regarding which the most appropriate proposal is regarding a particular problem.

We suggest using the following ones: learning time (LT) and learning memory (LM), which refer to the time taken and the memory required to learn a rule set, respectively; and extraction time (ET) and extraction memory (EM), which refer to the time taken and the memory required to extract information from a document. If a proposal is based on heuristics, then its learning time and its learning memory can be trivially set to zero, since it does not learn any rules, but extracts information directly from a dataset.

Regarding the learning time and the extraction time, it is worth mentioning that it is common to distinguish between computer and user time. The former refers to the time that the CPU or the IO devices are allocated to running a process, whereas the latter refers to the total time that elapses since a process is started until it finishes, which includes the time that the computer is running other processes. Computer times tend to be quite stable, i.e., when an algorithm is repeatedly executed on the same input they do not vary largely; contrarily, user times are not so stable because they depend on many other processes that can run concurrently on the same machine. As a conclusion, our proposal is to measure computer times only.

B.2.1 A note on global error-related measures

Previously, we have reported on the error-related measures that we have found in the literature. Obviously, we recommend using the global ones since they are the only that report on how good a proposal is at both extracting the information in which we are interested and ignoring the rest. The standard is to use the F_1 score, which combines precision and recall. Unfortunately, our study reveals that this measure is not appropriate in our context.

The reason is that the F_1 score is skewed when it is computed on unbalanced datasets. A dataset is said to be unbalanced when the number of instances of a slot deviates from the number of instances of the remaining slots. In our context, the datasets are naturally unbalanced because the amount of information to be ignored in a web document typically exceeds the amount of information to be extracted; furthermore, some slots are optional and some others are multi-valued, which also contributes to making the datasets naturally unbalanced.

To understand the reason why using the F_1 score in the context of unbalanced datasets is problematic, we use the following example: assume that a proposal is evaluated on a dataset that has 15 documents that provide a total of 15 instances of a given slot; assume, too, that the resulting confusion matrix is $(tp_1, tn_1, fp_1, fn_1) = (15, 2371, 98, 0)$, which implies that precision is 0.13, recall is 1.00, and the F_1 score is 0.23. In other words, it does not seem to be a good proposal because precision is very low, but realise it is actually very good because it makes very few mistakes. Assume that another proposal is evaluated on a dataset that provides 15 documents, but only 13 instances of the slot being considered; assume, too, that the corresponding confusion matrix is $(tp_2, tn_2, fp_2, fn_2) = (13, 960, 40, 0)$. That is, its precision is 0.25, its recall is 1.00, and its F_1 score is 0.39. Neither seems this proposal to be excellent, but a little better than the previous one. Note however, that a deeper analysis can easily reveal that both proposals behave very similarly because they successfully extract every instance of the slot being considered and roughly 4% of the examples to be ignored are mistakenly extracted as belonging to that slot. In other words, they behave very similarly regarding their ability to extract or ignore information. The problem is that the F_1 score provides a distorted view of these proposals because they have been evaluated of different testing sets with different skews.

A good global error-related measure must depend only on the proposal being evaluated, not on the dataset used to evaluate it being balanced or unbalanced. That is, it should be possible to maximise the measure by improving the techniques that lie at the core of a proposal, not by changing the proportion of information to be ignored in a dataset.

To find out which of the global error-related measures that we have presented before is not skewed in the presence of unbalanced datasets, we have re-written their formulations in terms of the following measures: the true positive rate ($TPR = tp/(tp + fn)$), which measures the proportion of instances of a slot that are extracted as belonging to that slot with regard to the total number of actual instances of that slot, the false positive rate ($FPR = fp/(fp + tn)$), which measures the proportion of information that is extracted as belonging to a given slot with regard to the information that must be ignored, and the skew of the dataset ($S = (tn + fp)/(tp + fn)$), which measures the proportion of information to be ignored with regard to the information to be extracted as belonging to a given slot. Note that we have selected the true positive rate and the false positive rate because these measures provide a clear picture of how good a proposal is at extracting or ignoring information and they have been proven not to be skewed in the context of unbalanced datasets [51]. Next, we present the results of re-writing the global error-related measures in terms of the previous measures:

$$\begin{aligned}
F_1 &= \frac{2\ TPR}{FPR\ S + TPR + 1}\\
Acc &= -\frac{(FPR - 1)\ S - TPR}{S + 1}\\
MCC &= \frac{\alpha\ (FPR - TPR)\ \sqrt{FPR\ S + TPR}}{(FPR^2 - FPR)\ S^2 - FPR\ S + ((2\ FPR - 1)\ S - 1)\ TPR + TPR^2}\\
&\quad \text{where } \alpha = \sqrt{-FPR\ S + S - TPR + 1}\ \sqrt{S}\\
AUC\text{-}PR &= \frac{FPR\ S\ TPR + FPR\ S + TPR^2}{2\ (FPR\ S + TPR)}\\
AUC\text{-}ROC &= \frac{1}{2}\ (1 + TPR - FPR)
\end{aligned}$$

Realise that the area under the ROC curve is the only measure that does not depend on S when it is re-written, which analytically proves that it is the only measure that is not skewed in the context of unbalanced datasets. In other words, it is the only that we can recommend in our context. Regarding our previous examples, the area under the ROC curve is 0.98 in both cases, which reflects that the corresponding proposals were not that bad and that a dataset being unbalanced does not have an impact on the results.

B.2.2 A note on computing per-extractor measures

To compute a ranking, we need to compute the performance measures on a per-extractor level. It is very easy to compute per-extractor efficiency measures because we just need to measure the time that elapses since an experiment starts running until it finishes or to probe the maximum amount of memory requested. Regarding effectiveness measures, the problem is a little more involved because we can compute per-slot measures and we then have to combine them in a manner that makes sense in the context of unbalanced datasets.

Typically, the problem has been addressed using macro-, micro-, or weighted averages. Macro averages are calculated by computing the effectiveness measures in a per-slot basis and then computing their unweighted averages; micro averages are computed from a global confusion matrix that is, in turn, computed by adding the confusion matrices that correspond to each slot; weighted averages are computed like macro averages, but the measures are weighted by the number of actual instances of each slot. Our general recommendation is to use macro averages because micro- and weighted averages have been proven to be skewed in the context of unbalanced datasets [57].

We have also found some specific research results regarding the area under the ROC curve, which is the most appropriate effectiveness measure that we have found so far [51, 77, 109]. The only one that seems both effective and computationally tractable is the one by Hand and Till [77], which computes it as follows:

$$\text{AUC-ROC} = \frac{\sum\limits_{i,j \in S, i \prec j} \text{AUC-ROC}_{i,j}}{(|S|^2 - |S|)/2}$$

where $\text{AUC-ROC}_{i,j}$ refers to a new pairwise measure that combines every two slots, $\prec$ denotes an arbitrary ordering of the slots, e.g., a lexicographic ordering, and S denotes the set of slots to be extracted. Note that there are $(|S|^2 - |S|)/2$ pairs of slots if their order is not taken into account. Simply put, the proposal amounts to macro averaging the pairwise area under the ROC curve, which has proven to work very well.

B.2.3 A note on implementation-related measures

The failure ratio, the timings, and the amounts of memory are related to a particular implementation of a proposal. Some researchers might argue that they are not appropriate as performance measures because they might lead to a distorted view of a proposal. The reason is that they depend on a programmer's ability to produce efficient code, on the implementation language, on the hardware and the software used to run the experiments. In other words, an intrinsically very efficient proposal might seem worse than another one because it was not well implemented or because the experimentation environment was not configured properly. However, we think that the failure ratio, the timings, and the amounts of memory are the only way for a practitioner to have a good overall picture of how a proposal performs in practice.

Some researchers might argue that we should evaluate the efficiency of a proposal building on its theoretical time or space complexity, but we do not think that such an approach is realistic because only a few authors have characterised the theoretical complexity of their proposals; furthermore, many of them have characterised an upper bound to the actual theoretical complexity to prove that their proposals are computationally tractable, not their actual complexity; even worse: even if we knew the exact theoretical complexity of every proposal, the relationships amongst most theoretical complexity classes are still open problems in computer science [68].

Thus our conclusion is that the failure ratio, the timings, and the amounts of memory that we propose to compute are very appropriate from a practitioner's point of view.

B.3 Statistical analysis

We have made a point of using sound methods to support our conclusions. Next, we report on the ranking heuristic that we have used to configure our proposals and on the hypothesis testing procedures that we have used to find differences in empirical ranks.

B.3.1 A ranking heuristic

When comparing several proposals or several alternatives to configure a proposal, we need to combine the performance measures that we have collected into a single rank per proposal or alternative. Unfortunately, there is not a widely accepted proposal in the literature to combine effectiveness and efficiency measures in a single rank. Next, we present a proposal that takes into account the means and deviations of the performance measures, as well as the relative weights that the experimenter has assigned to them.

Let M denote a set of performance measures. We assume that the experimenter provides a map β that assigns a weight in range $[0.00 \mathrel{..} 1.00]$ to every measure in M. Obviously, the weights must sum up to 1.00 so that they are consistent. Now, assume that we are dealing with a performance measure m, that we have gathered a set of values W regarding it, that a denotes the minimum value in set W, and that b denotes the maximum value. If m has to be maximised, then we define the set of normalised values of m as $W' = \{w' \mid \exists w \bullet w \in W \land w' = (w - a) \operatorname{div} (b - a)\}$; if m has to be minimised, then we define the set of normalised values of m as $W' = \{w' \mid \exists w \bullet w \in W \land w' = 1.00 - (w - a) \operatorname{div} (b - a)\}$. ($x \operatorname{div} y$ equals x/y if $y \neq 0.00$; otherwise, it equals 1.00.) The values in W' range in interval $[0.00 \mathrel{..} 1.00]$, so that the closer a value to the lower bound the worse and the closer to the upper bound the better. Let M' denote a set of new measures that are in one-to-one correspondence with the measures in M, but are normalised according to the previous procedure. Our proposal is to compute the rank of alternative p as follows:

$$K^p = \sum_{m' \in M'} \beta(m) \frac{\text{mdr}^p_{m'}}{\text{mdr}^{max}_{m'}}$$

where $\text{mdr}^p_{m'}$ denotes the mean-to-deviation ratio of alternative p with regard to normalised performance measure m' and $\text{mdr}^{max}_{m'}$ denotes the maximum mean-to-deviation ratio of performance measure m' across all of the alternatives. This ratio is defined as follows:

$$\text{mdr}^p_m = \begin{cases} \frac{(\mu^p_m)^2}{\sigma^p_m} \text{if } \sigma^p_m \neq 0.00 \\ \mu^p_m \quad \text{otherwise} \end{cases}$$

where m denotes an arbitrary performance measure, μ^p_m denotes its mean value regarding alternative p, and σ^p_m its standard deviation regarding alternative p. Note that this ratio maps every measure onto a value that weights its mean value with the inverse coefficient of variation ($\frac{\sigma^p_m}{\mu^p_m}$) as long as the standard deviation is not zero; intuitively, the smallest the coefficient of variation with respect to the mean value, the better that measure because it is more stable. If the standard deviation is zero, then the mean-to-deviation ratio is trivially defined as the mean value.

Before concluding, we would like to mention that our experimental analysis has revealed that some alternatives fail when they are applied to some datasets. Sometimes, the reason is that they consume too much memory; sometimes, they cannot learn a rule in a reasonable time (we set a deadline of 1 CPU day). That means that we also need to compute a failure ratio for every alternative under consideration, cf. Section §B.1. We, obviously, are not willing to accept an alternative whose failure ratio is different from 0.00, since that means that it is not generally applicable.

B.3.2 Hypothesis testing

In every case, we have conducted a statistical analysis to make sure that the differences in rank that our experiments have found are statistically significant at the standard significance level $\alpha = 0.05$. Following the results in Demšar [44] and García and Herrera [66], we have used Iman-Davenport's test to find out if there are statistically significant differences in the empirical ranks and then Hommel's test to compare the best ranked proposal to the remaining ones.

We have to resort to non-parametric tests because we found out that the distribution of the performance measures in our experiments was neither normal nor homoscedastic [153]. As an example regarding normality, consider WIEN's precision: Kolmogorov-Smirnov's test returns $D = 0.78$ with a p-value less than $2.20\,10^{-16}$, Shapiro-Wilk's test returns $W = 0.79$ with p-value $2.39\,10^{-07}$, and Arlinton-Darling's test returns $AD = 49.85$ with p-value $1.11\,10^{-05}$. Regarding homoscedasticity, consider WIEN's precision and Soft-Mealy's precision: Levene's test returns $F = 49.64$ with p-value $1.94\,10^{-10}$, Bartlett's test returns $K = 69.66$ with p-value less than $2.20\,10^{-16}$, and the F test returns $F = 0.08$ with p-value less than $2.20\,10^{-16}$. Note that the p-value is extremely close to 0.00 in every case, which provides a strong indication that the data do not behave normally and are not homoscedastic, which supports using non-parametric tests.

Bibliography

[1] B. Adelberg. *NoDoSE: a tool for semi-automatically extracting semi-structured data from text documents*. In *SIGMOD Conference*, pages 283–294, 1998.

[2] E. Agichtein and L. Gravano. *Snowball: extracting relations from large plain-text collections*. In *ICDL*, pages 85–94, 2000.

[3] D. Ajwani, A. Cosgaya-Lozano, and N. Zeh. *A topological sorting algorithm for large graphs*. *ACM Journal of Experimental Algorithmics*, 17 (1), 2011.

[4] M. Álvarez, A. Pan, J. Raposo, F. Bellas, and F. Cacheda. *Extracting lists of data records from semi-structured web pages*. *Data Knowl. Eng.*, 64(2):491–509, 2008.

[5] M. Álvarez, A. Pan, J. Raposo, F. Bellas, and F. Cacheda. *Finding and extracting data records from web pages*. *Signal Processing Systems*, 59 (1):123–137, 2010.

[6] A. Arasu and H. Garcia-Molina. *Extracting structured data from web pages*. In *SIGMOD Conference*, pages 337–348, 2003.

[7] G. O. Arocena and A. O. Mendelzon. *WebOQL: restructuring documents, databases, and webs*. *TAPOS*, 5(3):127–141, 1999.

[8] J. H. Aseltine. *WAVE: an incremental algorithm for information extraction*. In *AAAI*, 1999.

[9] F. Ashraf, T. Özyer, and R. Alhajj. *Employing clustering techniques for automatic information extraction from HTML documents*. *IEEE Transactions on Systems, Man, and Cybernetics, Part C*, 38(5):660–673, 2008.

[10] A. Atramentov, H. Leiva, and V. Honavar. *A multi-relational decision tree learning algorithm*. In *ILP*, pages 38–56, 2003.

[11] R. Basili, M. T. Pazienza, and M. Vindigni. *Corpus-driven learning of event recognition rules*. In *ECAI Workshop on Machine Learning for Information Extraction*, pages 1–7, 2000.

[12] G. E. A. P. A. Batista, R. C. Prati, and M. C. Monard. *A study of the behavior of several methods for balancing machine learning training data*. *SIGKDD Explorations*, 6(1):20–29, 2004.

[13] R. Baumgartner, S. Flesca, and G. Gottlob. *Visual web information extraction with Lixto*. In *VLDB*, pages 119–128, 2001.

[14] R. Baumgartner, O. Frölich, and G. Gottlob. *The Lixto systems applications in business intelligence and the Semantic Web*. In *ESWC*, pages 16–26, 2007.

[15] H. Blockeel and L. D. Raedt. *Top-down induction of first-order logical decision trees*. *Artif. Intell.*, 101(1-2):285–297, 1998.

[16] H. Blockeel, L. D. Raedt, N. Jacobs, and B. Demoen. *Scaling up inductive logic programming by learning from interpretations*. *Data Min. Knowl. Discov.*, 3(1):59–93, 1999.

[17] B. Bos, T. Çelik, I. Hickson, and H. W. Lie. *Cascading style sheets specification*. Technical report, W3C, 2014.

[18] S. Brin. *Extracting patterns and relations from the World Wide Web*. In *WebDB*, pages 172–183, 1998.

[19] C. Bădică, A. Bădică, E. Popescu, and A. Abraham. *L-Wrappers: concepts, properties and construction*. *Soft Comput.*, 11(8):753–772, 2007.

[20] D. Buttler, L. Liu, and C. Pu. *A fully automated object extraction system for the World Wide Web*. In *ICDCS*, pages 361–370, 2001.

[21] M. E. Califf and R. J. Mooney. *Bottom-up relational learning of pattern matching rules for information extraction*. *Journal of Machine Learning Research*, 4:177–210, 2003.

[22] N. Català, N. Castell, and M. Martin. *A portable method for acquiring information extraction patterns without annotated corpora*. *Natural Language Engineering*, 9(2):151–179, 2003.

[23] J. Y. Chai and A. W. Biermann. *The use of lexical semantics in information extraction*. In *ACL Workshop on Automatic Information Extraction and Building of Lexical Semantic Resources for NLP Applications*, pages 61–70, 1997.

[24] J. Y. Chai, A. W. Biermann, and C. I. Guinn. *Two dimensional generalization in information extraction*. In *AAAI*, pages 431–438, 1999.

[25] C.-H. Chang, M. Kayed, M. R. Girgis, and K. F. Shaalan. *A survey of web information extraction systems*. *IEEE Trans. Knowl. Data Eng.*, 18(10):1411–1428, 2006.

[26] C.-H. Chang and S.-C. Kuo. *OLERA: semisupervised web-data extraction with visual support*. *IEEE Intelligent Systems*, 19(6):56–64, 2004.

[27] C.-H. Chang, Y.-L. Lin, K.-C. Lin, and M. Kayed. *Page-level wrapper verification for unsupervised web data extraction*. In *WISE (1)*, pages 454–467, 2013.

[28] C.-H. Chang and S.-C. Lui. *IEPAD: information extraction based on pattern discovery*. In *WWW*, pages 681–688, 2001.

[29] B. Chidlovskii. *Wrapping web information providers by transducer induction*. In *ECML*, pages 61–72, 2001.

[30] B. Chidlovskii. *Automatic repairing of web wrappers by combining redundant views*. In *ICTAI*, pages 399–406, 2002.

[31] H. L. Chieu and H. T. Ng. *A maximum entropy approach to information extraction from semi-structured and free text*. In *AAAI/IAAI*, pages 786–791, 2002.

[32] H. L. Chieu, H. T. Ng, and Y. K. Lee. *Closing the gap: learning-based information extraction rivaling knowledge-engineering methods*. In *ACL*, pages 216–223, 2003.

[33] N. Chinchor, L. Hirschman, and D. D. Lewis. *Evaluating message understanding systems: an analysis of the third Message Understanding Conference*. *Computational Linguistics*, 19(3):409–449, 1993.

[34] L. Chiticariu, Y. Li, S. Raghavan, and F. Reiss. *Enterprise information extraction: recent developments and open challenges*. In *SIGMOD Conference*, pages 1257–1258, 2010.

[35] W. W. Cohen, M. Hurst, and L. S. Jensen. *A flexible learning system for wrapping tables and lists in HTML documents*. In *WWW*, pages 232–241, 2002.

[36] C. Cox, J. Nicolson, J. R. Finkel, C. Manning, and P. Langley. *Template sampling for leveraging domain knowledge in information extraction*. In *PASCAL Challenges Workshop*, 2005.

[37] V. Crescenzi and G. Mecca. *Grammars have exceptions*. *Inf. Syst.*, 23(8): 539–565, 1998.

[38] V. Crescenzi and G. Mecca. *Automatic information extraction from large websites*. *J. ACM*, 51(5):731–779, 2004.

[39] V. Crescenzi, G. Mecca, and P. Merialdo. *RoadRunner: towards automatic data extraction from large web sites*. In *VLDB*, pages 109–118, 2001.

[40] V. Crescenzi and P. Merialdo. *Wrapper inference for ambiguous web pages*. *Applied Artificial Intelligence*, 22(1&2):21–52, 2008.

[41] C. M. Cumby and D. Roth. *On kernel methods for relational learning*. In *ICML*, pages 107–114, 2003.

[42] N. N. Dalvi, A. Machanavajjhala, and B. Pang. *An analysis of structured data on the Web*. *PVLDB*, 5(7):680–691, 2012.

[43] J. Davis, E. S. Burnside, I. de Castro Dutra, D. Page, and V. S. Costa. *An integrated approach to learning Bayesian networks of rules*. In *ECML*, pages 84–95, 2005.

[44] J. Demšar. *Statistical comparisons of classifiers over multiple data sets*. *Journal of Machine Learning Research*, 7:1–30, 2006.

[45] A. Douthat. *The Message Understanding Conference scoring software: A user's manual*, 1998. URL: http://www.itl.nist.gov/iaui/894.02/related_projects/muc/muc_sw/muc_sw_manual.html.

[46] S. Džeroski and N. Lavrač. *Inductive learning in deductive databases*. *IEEE Trans. Knowl. Data Eng.*, 5(6):939–949, 1993.

[47] L. Eikvil. *Information extraction from the World Wide Web: A survey*. Technical report 945, Norweigan Computing Center, 1999.

[48] H. Elmeleegy, J. Madhavan, and A. Y. Halevy. *Harvesting relational tables from lists on the Web*. *PVLDB*, 2(1):1078–1089, 2009.

[49] H. Elmeleegy, J. Madhavan, and A. Y. Halevy. *Harvesting relational tables from lists on the Web*. *VLDB J.*, 20(2):209–226, 2011.

[50] W. Emde and D. Wettschereck. *Relational instance-based learning*. In *ICML*, pages 122–130, 1996.

[51] T. Fawcett. *An introduction to ROC analysis*. *Pattern Recognition Letters*, 27(8):861–874, 2006.

[52] B. Fazzinga, S. Flesca, and A. Tagarelli. *Schema-based web wrapping*. *Knowl. Inf. Syst.*, 26(1):127–173, 2011.

[53] J. I. Fernández-Villamor, C. A. Iglesias, and M. Garijo. *First-order logic rule induction for information extraction from web resources*. *International Journal on Artificial Intelligence Tools*, 21(6), 2012.

[54] C. Ferri, J. Hernández-Orallo, and R. Modroiu. *An experimental comparison of performance measures for classification*. *Pattern Recognition Letters*, 30(1):27–38, 2009.

[55] A. Finn and N. Kushmerick. *Information extraction by convergent boundary classification*. In *AAAI Workshop on Adaptive Text Extraction And Mining*, 2004.

[56] P. A. Flach and N. Lachiche. *Naive Bayesian classification of structured data*. *Machine Learning*, 57(3):233–269, 2004.

[57] G. Forman. *A pitfall and solution in multi-class feature selection for text classification*. In *ICML*, 2004.

[58] E. Frank, M. A. Hall, G. Holmes, R. Kirkby, B. Pfahringer, I. H. Witten, and L. Trigg. *Weka: a machine learning workbench for data mining*. In *Data Mining and Knowledge Discovery Handbook*, pages 1269–1277. Springer, 2010.

[59] D. Freitag. *Information extraction from HTML: application of a general machine learning approach*. In *AAAI/IAAI*, pages 517–523, 1998.

[60] D. Freitag. *Multistrategy learning for information extraction*. In *International Conference on Machine Learning*, pages 161–169, 1998.

[61] D. Freitag. *Toward a general-purpose learning for information extraction*. In *COLING-ACL*, pages 404–408, 1998.

[62] D. Freitag. *Machine learning for information extraction in informal domains*. *Machine Learning*, 39(2/3):169–202, 2000.

[63] D. Freitag and A. McCallum. *Information extraction with HMM structures learned by stochastic optimization*. In *AAAI*, pages 584–589, 2000.

[64] D. Freitag and A. K. Mccallum. *Information extraction with HMMs and shrinkage*. In *AAAI Workshop on Machine Learning for Information Extraction*, pages 31–36, 1999.

[65] J. Fürnkranz. *Separate-and-conquer rule learning*. *Artif. Intell. Rev.*, 13 (1):3–54, 1999.

[66] S. García and F. Herrera. *An extension on 'statistical comparisons of classifiers over multiple data sets' for all pair-wise comparisons*. *Journal of Machine Learning Research*, 9:2677–2694, 2008.

[67] T. Gärtner, J. W. Lloyd, and P. A. Flach. *Kernels and distances for structured data*. *Machine Learning*, 57(3):205–232, 2004.

[68] W. I. Gasarch. *Classifying problems into complexity classes*. *Advances in Computers*, 95:239–292, 2015.

[69] L. Geng and H. J. Hamilton. *Interestingness measures for data mining: A survey*. *ACM Comput. Surv.*, 38(3), 2006.

[70] L. Getoor, N. Friedman, D. Koller, and B. Taskar. *Learning probabilistic models of relational structure*. In *ICML*, pages 170–177, 2001.

[71] D. G. Gregg and S. Walczak. *Exploiting the Information Web*. *IEEE Transactions on Systems, Man, and Cybernetics, Part C*, 37(1):109–125, 2007.

[72] T. Grigalis and A. Cenys. *Unsupervised structured data extraction from template-generated web pages*. *J. UCS*, 20(2):169–192, 2014.

[73] P. Gulhane, A. Madaan, R. R. Mehta, J. Ramamirtham, R. Rastogi, S. Satpal, S. H. Sengamedu, A. Tengli, and C. Tiwari. *Web-scale information extraction with Vertex*. In *ICDE*, pages 1209–1220, 2011.

[74] H. Guo and H. L. Viktor. *Multirelational classification: A multiple view approach*. *Knowl. Inf. Syst.*, 17(3):287–312, 2008.

[75] R. Gupta and S. Sarawagi. *Answering table augmentation queries from unstructured lists on the Web*. *PVLDB*, 2(1):289–300, 2009.

[76] J. Hammer, J. McHugh, and H. Garcia-Molina. *Semistructured data: the Tsimmis experience*. In *ADBIS*, pages 1–8, 1997.

[77] D. J. Hand and R. J. Till. *A simple generalisation of the area under the ROC curve for multiple class classification problems*. *Machine Learning*, 45(2):171–186, 2001.

[78] J. A. Hanley and B. J. McNeil. *The meaning and use of the area under a receiver operating characteristic (ROC) curve*. *Radiology*, 143: 29–36, 1982.

[79] H. He and E. A. Garcia. *Learning from imbalanced data*. *IEEE Trans. Knowl. Data Eng.*, 21(9):1263–1284, 2009.

[80] I. Hickson, R. Berjon, S. Faulkner, T. Leithead, E. D. Navara, E. O'Connor, and S. Pfeiffer. *HTML 5: A vocabulary and associated APIs for HTML and XHTML*. Technical report, W3C, 2014.

[81] L. Hirschman. *The evolution of evaluation: lessons from the Message Understanding Conferences*. *Computer Speech & Language*, 12 (4):281–305, 1998.

[82] A. W. Hogue and D. R. Karger. *Thresher: automating the unwrapping of semantic content from the World Wide Web*. In *WWW*, pages 86–95, 2005.

[83] J. L. Hong, E.-G. Siew, and S. Egerton. *Information extraction for search engines using fast heuristic techniques*. *Data Knowl. Eng.*, 69(2): 169–196, 2010.

[84] T. Horváth, S. Wrobel, and U. Bohnebeck. *Relational instance-based learning with lists and terms*. *Machine Learning*, 43(1/2):53–80, 2001.

[85] C.-N. Hsu and M.-T. Dung. *Generating finite-state transducers for semi-structured data extraction from the Web*. *Inf. Syst.*, 23(8):521–538, 1998.

[86] S. B. Huffman. *Learning information extraction patterns from examples*. In *Learning for Natural Language Processing*, pages 246–260, 1995.

[87] N. Ireson, F. Ciravegna, M. E. Califf, D. Freitag, N. Kushmerick, and A. Lavelli. *Evaluating machine learning for information extraction*. In *International Conference on Machine Learning*, pages 345–352, 2005.

[88] U. Irmak and T. Suel. *Interactive wrapper generation with minimal user effort*. In *WWW*, pages 553–563, 2006.

[89] M. Jaeger. *Probabilistic-logic models: Reasoning and learning with relational structures*. In *SCAI*, pages 197–200, 2008.

[90] P. Jiménez and R. Corchuelo. *On extracting information from semi-structured deep web documents*. In *Business Information Systems*, pages 140–151, 2015.

[91] N. Kambhatla. *Combining lexical, syntactic, and semantic features with maximum entropy models for extracting relations*. In *ACL (Interactive poster & demonstration sessions)*, pages 1–4, 2004.

[92] Y. Kavurucu, P. Senkul, and I. H. Toroslu. *A comparative study on ILP-based concept discovery systems. Expert Syst. Appl.*, pages 11598–11607, 2011.

[93] M. Kayed and C.-H. Chang. *FiVaTech: page-level web data extraction from template pages. IEEE Trans. Knowl. Data Eng.*, 22(2):249–263, 2010.

[94] J.-T. Kim and D. I. Moldovan. *Acquisition of linguistic patterns for knowledge-based information extraction*. *IEEE Trans. Knowl. Data Eng.*, 7(5):713–724, 1995.

[95] Y.-J. Kim. *Emerging trends: 2010 through 2015*, 2005. URL: http://www.kait.or.kr/filedb/051207-it839/KAIT-1.pdf.

[96] A. J. Knobbe, M. de Haas, and A. Siebes. *Propositionalisation and aggregates*. In *PKDD*, pages 277–288, 2001.

[97] R. Kosala, H. Blockeel, M. Bruynooghe, and J. V. den Bussche. *Information extraction from structured documents using k-testable tree automaton inference*. *Data Knowl. Eng.*, 58(2):129–158, 2006.

[98] S. Kramer, N. Lavrač, and P. Flach. *Propositionalization approaches to relational data mining*. In S. Džeroski and N. Lavrač, editors, *Relational Data Mining*, pages 262–291. Springer, 2001.

[99] S. Kramer, G. Widmer, B. Pfahringer, and M. de Groeve. *Prediction of ordinal classes using regression trees*. *Fundam. Inform.*, 47(1-2): 1–13, 2001.

[100] R. Krishnamurthy, Y. Li, S. Raghavan, F. Reiss, S. Vaithyanathan, and H. Zhu. *SystemT: a system for declarative information extraction*. *SIGMOD Record*, 37(4):7–13, 2008.

[101] M.-A. Krogel, S. Rawles, F. Zelezný, P. A. Flach, N. Lavrač, and S. Wrobel. *Comparative evaluation of approaches to propositionalization*. In *ILP*, pages 197–214, 2003.

[102] M.-A. Krogel. *On propositionalization for knowledge discovery in relational databases*. PhD thesis, Otto von Guericke Universität Magdeburg, 2005.

[103] S. Kuhlins and R. Tredwell. *Toolkits for generating wrappers*. In *NetObjectDays*, pages 184–198, 2002.

[104] N. Kushmerick. *Regression testing for wrapper maintenance*. In *AAAI/IAAI*, pages 74–79, 1999.

[105] N. Kushmerick. *Wrapper verification*. *World Wide Web*, 3(2):79–94, 2000.

[106] N. Kushmerick and B. Thomas. *Adaptive information extraction: Core technologies for information agents*. In *Intelligent Information Agents - The AgentLink Perspective*, pages 79–103, 2003.

[107] N. Kushmerick, D. S. Weld, and R. B. Doorenbos. *Wrapper induction for information extraction*. In *IJCAI (1)*, pages 729–737, 1997.

[108] A. Lally, K. Verspoor, and E. Nyberg. *Unstructured information managemen architecture (uima) version 1.0*, 2009. URL: http://docs.oasis-open.org/uima/v1.0/uima-v1.0.pdf.

[109] T. Landgrebe and R. P. W. Duin. *Approximating the multiclass ROC by pairwise analysis*. *Pattern Recognition Letters*, 28(13):1747–1758, 2007.

[110] N. Landwehr, K. Kersting, and L. D. Raedt. *Integrating Naïve Bayes and FOIL*. *Journal of Machine Learning Research*, 8:481–507, 2007.

[111] N. Landwehr, A. Passerini, L. D. Raedt, and P. Frasconi. *kFOIL: Learning simple relational kernels*. In *AAAI*, pages 389–394, 2006.

[112] A. Lavelli, M. E. Califf, F. Ciravegna, D. Freitag, C. Giuliano, N. Kushmerick, and L. Romano. *A critical survey of the methodology for IE evaluation*. In *LREC*, 2004.

[113] A. Lavelli, M. E. Califf, F. Ciravegna, D. Freitag, N. Kushmerick, C. Giuliano, L. Romano, and N. Ireson. *Evaluation of machine learning-based information extraction algorithms: Criticisms and recommendations*. *Language Resources and Evaluation*, 42(4):361–393, 2008.

[114] N. Lavrač and S. Džeroski. *Inductive logic programming: Techniques and applications*. Ellis Horwood, 1994.

[115] W. G. Lehnert and B. Sundheim. *A performance evaluation of text analysis technologies*. *AI Magazine*, 12(3):81–94, 1991.

[116] K. Lerman, S. Minton, and C. A. Knoblock. *Wrapper maintenance: a machine learning approach*. *J. Artif. Intell. Res.*, 18:149–181, 2003.

[117] L. Li, Y. Liu, A. Obregon, and M. Weatherston. *Visual segmentation-based data record extraction from web documents*. In *IRI*, pages 502–507, 2007.

[118] Q. Li, Y. Ding, A. Feng, and Y. Dong. *A novel method for extracting information from web pages with multiple presentation templates*. *JSW*, 5(5):506–513, 2010.

[119] B. Liu and Y. Zhai. *NET: a system for extracting web data from flat and nested data records*. In *WISE*, pages 487–495, 2005.

[120] L. Liu, C. Pu, and W. Han. *XWRAP: an XML-enabled wrapper construction system for web information sources*. In *ICDE*, pages 611–621, 2000.

[121] W. Liu, X. Meng, and W. Meng. *ViDE: a vision-based approach for deep web data extraction*. *IEEE Trans. Knowl. Data Eng.*, 22(3):447–460, 2010.

[122] A. Machanavajjhala, A. S. Iyer, P. Bohannon, and S. Merugu. *Collective extraction from heterogeneous web lists*. In *WSDM*, pages 445–454, 2011.

[123] R. McCann, B. K. AlShebli, Q. Le, H. Nguyen, L. Vu, and A. Doan. *Mapping maintenance for data integration systems*. In *VLDB*, pages 1018–1030, 2005.

[124] W. Mendenhall and T. T. Sincich. *A second course in statistics: Regression analysis*. Pearson, edition 7, 2011.

[125] W. Meng and C. T. Yu. *Advanced metasearch engine technology*. Synthesis Lectures on Data Management. Morgan & Claypool Publishers, 2010.

[126] X. Meng, D. Hu, and C. Li. *Schema-guided wrapper maintenance for web-data extraction*. In *WIDM*, pages 1–8, 2003.

[127] R. C. Miller and B. A. Myers. *Lightweight structured text processing*. In *USENIX Annual Technical Conference, General Track*, pages 131–144, 1999.

[128] S. Miller, H. Fox, L. A. Ramshaw, and R. M. Weischedel. *A novel use of statistical parsing to extract information from text*. In *ANLP*, pages 226–233, 2000.

[129] T. M. Mitchell. *Machine learning*. McGraw Hill, 1997.

[130] R. Mohapatra, K. Rajaraman, and S. Y. Sung. *Efficient wrapper reinduction from dynamic web sources*. In *Web Intelligence*, pages 391–397, 2004.

[131] S. Muggleton. *Learning stochastic logic programs*. *Electron. Trans. Artif. Intell.*, 4(B):141–153, 2000.

[132] S. Muggleton, L. D. Raedt, D. Poole, I. Bratko, P. A. Flach, K. Inoue, and A. Srinivasan. *ILP turns 20: Biography and future challenges*. *Machine Learning*, 86(1):3–23, 2012.

[133] I. Muslea. *RISE: repository of online information sources used in information extraction*, 1998. URL: http://www.isi.edu/info-agents/RISE.

[134] I. Muslea, S. Minton, and C. A. Knoblock. *Hierarchical wrapper induction for semistructured information sources*. *Autonomous Agents and Multi-Agent Systems*, 4(1/2):93–114, 2001.

[135] E. W. Noreen. *Computer intensive methods for testing hypotheses: An introduction*. John Wiley & Sons, 1989.

[136] N. Papadakis, D. Skoutas, K. Raftopoulos, and T. A. Varvarigou. *STAVIES: a system for information extraction from unknown web data sources through automatic web wrapper generation using clustering techniques*. *IEEE Trans. Knowl. Data Eng.*, 17(12):1638–1652, 2005.

[137] J. Park and D. Barbosa. *Adaptive record extraction from web pages*. In *WWW*, pages 1335–1336, 2007.

[138] L. Peshkin and A. Pfeffer. *Bayesian information extraction network*. In *IJCAI*, pages 421–426, 2003.

[139] J. R. Quinlan and R. M. Cameron-Jones. *Induction of logic programs: FOIL and related systems*. *New Generation Comput.*, 13(3&4):287–312, 1995.

[140] J. Raposo, A. Pan, M. Álvarez, and J. Hidalgo. *Automatically generating labeled examples for web wrapper maintenance*. In *Web Intelligence*, pages 250–256, 2005.

[141] J. Raposo, A. Pan, M. Álvarez, J. Hidalgo, and Á. Viña. *The Wargo System: semi-automatic wrapper generation in presence of complex data access modes*. In *DEXA Workshops*, pages 313–320, 2002.

[142] J. Raposo, A. Pan, M. Álvarez, and Á. Viña. *Automatic wrapper maintenance for semi-structured web sources using results from previous queries*. In *SAC*, pages 654–659, 2005.

[143] Reuters. *The reuters 21578 collection*, 1993. URL: http://www.daviddlewis.com/resources/testcollections/reuters21578/.

[144] E. Riloff. *Automatically constructing a dictionary for information extraction tasks*. In *AAAI*, pages 811–816, 1993.

[145] E. Riloff. *Automatically generating extraction patterns from untagged text*. In *AAAI/IAAI*, volume 2, pages 1044–1049, 1996.

[146] J. Rissanen. *The Minimum Description Length principle*. In *Encyclopedia of Machine Learning*, pages 666–668. Springer, 2010.

[147] J. Rivera and R. van der Meulen. *What's driving mobile data growth?* Technical report, Gartner, 2015.

[148] D. Roth and W.-T. Yih. *Relational learning via propositional algorithms: an information extraction case study*. In *IJCAI*, pages 1257–1263, 2001.

[149] N. Sager. *Medical language processing: computer management of narrative data*. Addison-Wesley, 1987.

[150] A. Sahuguet and F. Azavant. *Building intelligent web applications using lightweight wrappers*. *Data Knowl. Eng.*, 36(3):283–316, 2001.

[151] K. Seymore, A. McCallum, and R. Rosenfeld. *Learning hidden Markov model structure for information extraction*. In *AAAI*, pages 37–42, 1999.

[152] Y. K. Shen and D. R. Karger. *U-REST: an unsupervised record extraction system*. In *WWW*, pages 1347–1348, 2007.

[153] D. J. Sheskin. *Handbook of parametric and nonparametric statistical procedures*. Chapman and Hall/CRC, edition 5, 2012.

[154] K. Simon and G. Lausen. *ViPER: augmenting automatic information extraction with visual perceptions*. In *CIKM*, pages 381–388, 2005.

[155] M. Skounakis, M. Craven, and S. Ray. *Hierarchical hidden Markov models for information extraction*. In *IJCAI*, pages 427–433, 2003.

[156] H. A. Sleiman and R. Corchuelo. *Information extraction framework*. In *PAAMS (Workshops)*, pages 149–156, 2012.

[157] H. A. Sleiman and R. Corchuelo. *Towards a method for unsupervised web information extraction*. In *ICWE*, pages 427–430, 2012.

[158] H. A. Sleiman and R. Corchuelo. *An unsupervised technique to extract information from semi-structured web pages*. In *WISE*, pages 631–637, 2012.

[159] H. A. Sleiman and R. Corchuelo. *A survey on region extractors from web documents*. *IEEE Trans. Knowl. Data Eng.*, 25(9):1960–1981, 2013.

[160] H. A. Sleiman and R. Corchuelo. *TEX: an efficient and effective unsupervised web information extractor*. *Knowl.-Based Syst.*, 39:109–123, 2013.

[161] H. A. Sleiman and R. Corchuelo. *A class of neural-network-based transducers for web information extraction*. *Neurocomputing*, 135: 61–68, 2014.

[162] H. A. Sleiman and R. Corchuelo. *Trinity: on using trinary trees for unsupervised web data extraction*. *IEEE Trans. Knowl. Data Eng.*, 26(6): 1544–1556, 2014.

[163] S. Soderland. *Learning to extract text-based information from the World Wide Web*. In *KDD*, pages 251–254, 1997.

[164] S. Soderland. *Learning information extraction rules for semi-structured and free text*. *Machine Learning*, 34(1-3):233–272, 1999.

[165] M. Sokolova and G. Lapalme. *A systematic analysis of performance measures for classification tasks*. *Inf. Process. Manage.*, 45(4):427–437, 2009.

[166] A. Srinivasan. *The Aleph manual*. Technical report, University of Oxford, 2004.

[167] W. Su, J. Wang, and F. H. Lochovsky. *ODE: ontology-assisted data extraction*. *ACM Trans. Database Syst.*, 34(2), 2009.

[168] F. M. Suchanek, M. Sozio, and G. Weikum. *SOFIE: a self-organizing framework for information extraction*. In *WWW*, pages 631–640, 2009.

[169] A. Sun, M.-M. Naing, E.-P. Lim, and W. Lam. *Using support vector machines for terrorism information extraction*. In *ISI*, pages 1–12, 2003.

[170] B. Sundheim. *TIPSTER/MUC-5: information extraction system evaluation*. In *MUC*, pages 27–44, 1993.

[171] C. Tao and D. W. Embley. *Automatic hidden-web table interpretation by sibling page comparison*. In *ER*, pages 566–581, 2007.

[172] C. Tao and D. W. Embley. *Automatic hidden-web table interpretation, conceptualization, and semantic annotation*. *Data Knowl. Eng.*, 68 (7):683–703, 2009.

[173] J. Turmo, A. Ageno, and N. Català. *Adaptive information extraction*. *ACM Comput. Surv.*, 38(2), 2006.

[174] J. Turmo and H. Rodríguez. *Learning rules for information extraction*. *Nat. Lang. Eng.*, 8:167–191, 2002.

[175] R. van der Meulen and J. Rivera. *Gartner says power shift in business intelligence and analytics will fuel disruption*, 2015. URL: http://www.gartner.com/newsroom/id/2970917.

[176] A. van Kesteren, A. Gregor, A. Russell, and R. Berjon. *Document Object Model 4*. Technical report, W3C, 2014.

[177] J. Wang and F. H. Lochovsky. *Data extraction and label assignment for web databases*. In *WWW*, pages 187–196, 2003.

[178] Y. Xia, Y. Yang, S. Zhang, and H. Yu. *Automatic wrapper generation and maintenance*. In *PACLIC*, pages 90–99, 2011.

[179] Y. Yamada, N. Craswell, T. Nakatoh, and S. Hirokawa. *Testbed for information extraction from the Deep Web*. In *WWW (Alternate Track Papers & Posters)*, pages 346–347, 2004.

[180] R. Yangarber. *Counter-training in discovery of semantic patterns*. In *ACL*, pages 343–350, 2003.

[181] X. Yin, J. Han, J. Yang, and P. S. Yu. *Efficient classification across multiple database relations: A crossmine approach*. *IEEE Trans. Knowl. Data Eng.*, 18(6):770–783, 2006.

[182] D. Zelenko, C. Aone, and A. Richardella. *Kernel methods for relation extraction*. *Journal of Machine Learning Research*, 3:1083–1106, 2003.

[183] Y. Zhai and B. Liu. *Structured data extraction from the Web based on partial tree alignment*. *IEEE Trans. Knowl. Data Eng.*, 18(12):1614–1628, 2006.

[184] H. Zhang and J. Su. *Conditional independence trees*. In *ECML*, pages 513–524, 2004.

[185] H. Zhao, W. Meng, Z. Wu, V. Raghavan, and C. T. Yu. *Fully automatic wrapper generation for search engines*. In *WWW*, pages 66–75, 2005.

[186] H. Zhao, W. Meng, and C. T. Yu. *Automatic extraction of dynamic record sections from search engine result pages*. In *VLDB*, pages 989–1000, 2006.

[187] S. Zhao and R. Grishman. *Extracting relations with integrated information using kernel methods*. In *ACL*, 2005.

[188] J. Zhu, Z. Nie, J.-R. Wen, B. Zhang, and W.-Y. Ma. *Simultaneous record detection and attribute labeling in web data extraction*. In *KDD*, pages 494–503, 2006.

This document was typeset on September 11, 2015 at 17:03 using class RC-BOOK α2.14 for LATEX2ε. As of the time of writing this document, this class is not publicly available since it is in alpha version. Only members of The Distributed Group are using it to typeset their documents. Should you be interested in giving forthcoming public versions a try, please, do contact us at contact@tdg-seville.info. Thanks!