‘Long autonomy or long delay?’ The importance of domain in opinion mining

Fermín L. Cruz *, José A. Troyano, Fernando Enríquez, F. Javier Ortega, Carlos G. Vallejo

Department of Languages and Computer Systems, University of Seville, Spain

Keywords:
Sentiment analysis
Opinion mining
Feature-based opinion extraction
User-generated contents
Information extraction

A B S T R A C T

Nowadays, people do not only navigate the web, but they also contribute contents to the Internet. Among other things, they write their thoughts and opinions in review sites, forums, social networks, blogs and other websites. These opinions constitute a valuable resource for businesses, governments and consumers. In the last years, some researchers have proposed opinion extraction systems, mostly domain-independent ones, to automatically extract structured representations of opinions contained in those texts. In this work, we tackle this task in a domain-oriented approach, defining a set of domain-specific resources which capture valuable knowledge about how people express opinions on a given domain. These resources are automatically induced from a set of annotated documents. Some experiments were carried out on three different domains (user-generated reviews of headphones, hotels and cars), comparing our approach to other state-of-the-art, domain-independent techniques. The results confirm the importance of the domain in order to build accurate opinion extraction systems. Some experiments on the influence of the dataset size and an example of aggregation and visualization of the extracted opinions are also shown.

1. Introduction

Internet users generate a large amount of information while surfing the web. This information can be used to extract useful knowledge, either from the implicit information contained in the logs of user interactions with the Web, or from the explicit information provided by user-generated content. User-generated content has become a centerpiece in Web 2.0. People contribute contents as videos, pictures and, mainly, texts. Among other things, people write their thoughts and opinions on various topics in forums, blogs, review sites and other websites. These opinions constitute a very valuable information for governments, companies or consumers. But reading this huge, fast-changing amount of information is virtually impossible. Some novel techniques within Natural Language Processing (NLP) try to solve this problem.

Sentiment analysis is a modern subdiscipline of NLP which deals with subjectivity, affects and opinions in texts (a good survey on this subject can be found in Pang & Lee (2008) and Liu & Zhang (2012)). Within sentiment analysis, the *feature-based opinion extraction* is a task related to information extraction, which consists in extracting structured representations of opinions on features of some object from subjective texts. For example, given the sentence *“The customer service is terrible”*, a negative opinion on feature *customer service* should be extracted. The task has many practical applications. For example, it allows a company to monitor real-time opinions on the Internet about their products. It also makes it easier for customers to choose between different products, based on the positive and negative opinions available online. Although one might think that these tasks can be performed manually, the large number of opinions on the Internet and the speed of appearance of new opinions encourage the use of automatic extraction methods.

Some researchers have proposed several approaches to this task, often unsupervised, domain-independent ones. In most cases, they select a few words from the sentence representing the feature affected by the opinion (*opinion target* or *feature words*, depending on authors). This approach entails some problems. First, sometimes the same feature can be named in different ways. For example, *customer service* is also known as *helpline* or *help desk* in some contexts. So a further matching problem must be solved in order to be able to aggregate opinions on the same feature. Besides, some features may include others; for example, someone looking for opinions about the *sound quality* of an audio system would be interested not only in those sentences explicitly referring to the sound quality (e.g., “The *sound quality* is superb”, “Very clean, outstanding *sound*”), but also in sentences talking about some other related features (e.g., “The *low end* is clear and the *high* is twangy”). Dealing with these issues is important in order to properly aggregate the extracted opinions and exploit the huge amount of available information.

In this work, we present a taxonomy-based approach to the opinion extraction task. We are interested in extracting feature-level opinions and mapping them into a *feature taxonomy*, a semantic

* Corresponding author. Address: Escuela Técnica Superior de Ingeniería Informática, Av. Reina Mercedes s/n, 41012 Sevilla, Spain. Tel.: +34 954 55 62 33.
E-mail address: fcruz@us.es (F.L. Cruz).

representation of the opinable parts and attributes of an object. As we rely on feature taxonomies specific for each class of objects being reviewed, our approach is domain-oriented. The importance of the domain in sentiment analysis is well known, and in this work we try to empirically demonstrate it for the opinion extraction task. We define a set of domain-specific resources which capture valuable knowledge about how people express opinions on a given domain. These resources are automatically induced from a set of annotated documents. We compare the results obtained by our opinion extraction system, using these domain-specific resources, to other state-of-the-art, domain-independent techniques in order to test our hypothesis.

The use of a feature taxonomy and the adaptation of the extraction system to a given domain are the main contributions of our approach. These aspects lead to better results than other state-of-art approaches.

The paper structure is as follows. In Section 2, we comment some related works on opinion extraction. In Section 3, we propose a redefinition of the opinion extraction problem, describe the domain-specific resources used by our system and the process to generate them, and briefly explain our system architecture. In Section 4, we report some experimental results, and finally we point out some conclusions in Section 5.

2. Previous works

The first work that deals with opinions from product reviews is Dave, Lawrence, and Pennock (2003). In this paper, they train a binary document classifier using a collection of reviews, and then use this model to classify individual sentences. Therefore, the granularity is at sentence level, rather than feature level (throughout the paper, they talk about "features", but meaning *linguistic features* used by the classifier). Anyway, they are the first to propose a certain type of summary composed of positive and negative terms found in the reviews, which is the main goal of the following works on feature-based opinion extraction.

The first definition of feature-based opinion extraction task appears in Hu and Liu (2004b). The task is divided into two steps: first, identifying the product features on which opinions are expressed, and second, identifying the sentences in which there are positive or negative opinions regarding those features. In the work concerned, they only describe a solution for the first task. The detection of product features is performed in an unsupervised manner, starting from a set of reviews of the product. Using association rule mining (Agrawal & Srikant, 1994), frequent features are sought from the noun phrases (up to three words long) that appear more frequently. After a pruning step to discard meaningless or redundant candidates, they search for adjectives appearing in the near context of the remaining features, which are then used in turn to find new ones. The main problem of this method is that a huge set of features is obtained, many of which are actually different names for the same feature. In Hu and Liu (2004a) they describe their approach to the second step of the feature-based opinion extraction task. The opinions extracted on the features obtained by the previous method are not very useful to build summaries, as opinions on the same feature but named in a different way are impossible to aggregate. In subsequent papers (Ding, Liu, & Yu, 2008; Zhai, Liu, Xu, & Jia, 2010), several solutions to this problem are proposed, e.g. building clusters of features. Anyway, these solutions do not consider the existence of features that are specializations of other features (e.g., *bass* or *treble* of a sound system would be sub-features of the *sound quality* feature).

There also exist some approaches to the opinion extraction problem based on modified versions of *Latent Dirichlet Allocation* (LDA) (Blei, Ng, & Jordan, 2003), as Titov and McDonald (2008a, 2008b), Brody and Elhadad (2010), Zhao, Jiang, Yan, and Li (2010) and Jo and Oh (2011). LDA is an unsupervised topic-based document modelling technique, which models an input document as a mixture of topics and is able to identify sets of words related to each topic. The different modified versions proposed in the context of opinion extraction are intended to detect abstract features (aspects) instead of topics. Note that this aspect-based opinion extraction task is radically different to the feature-based opinion extraction task, as the former consists in identifying the aspects reviewed in a piece of text based on a bag-of-words model of the document, rather than extracting individual feature mentions and their related opinions. The approach being discussed in this paper focuses on the feature-based opinion extraction task, so it is not directly comparable to these works.

3. A domain-adaptable approach to feature-based opinion extraction

We think that the biggest shortcoming of the above approaches to feature-based opinion extraction is their generality. None of these works take the domain into account, so the same exact system must be able to extract opinions from a review of a digital camera, a hotel, a movie, or any other type of product. It is common to find terms that are used to express positive opinions in one domain and negative opinions in another (or even in the same domain but applied to different features). For example, the adjective *long* has positive implications when applied to the autonomy of a laptop and negative implications when applied to the average delay of an airline. So in our approach we are not only using domain-specific but also feature-specific semantic orientations (a measure of the positive or negative implications of the term when used in an opinion). We also take the domain into account when performing other involved subtasks, like finding feature mentions, and the related opinion words. In previous works, the possible relations of synonymy, specialization or inclusion between the feature mentions remain undiscovered. This makes the extracted opinions useless in order to aggregate them and build summaries. We address this problem by using feature taxonomies.

In summary, the main guidelines of our approach are (1) building a feature taxonomy for each new domain, so our system will extract opinions on those features and map them into the taxonomy, and (2) automatically generating domain-specific, feature-level resources which capture knowledge about how people express opinions on each feature for a given domain. Our approach leads to a higher quality opinion extraction, at the expense of a small manual effort to annotate some documents from the selected domain. A conceptual representation of our proposal is shown in Fig. 1. In the next sections, we define the problem, describe the supporting resources and system architecture, and show the results of some experiments.

3.1. Problem definition

We are focusing on extracting opinions from product reviews. A *product* is any object or service that can be consumed by users, e.g. a car, a movie, a hotel,etc. A *feature* is any property, component or aspect of a product; the product itself is considered a feature, and any property, component or aspect of a feature is in itself a feature. A *review* is a text document where an expert or an anonymous user critically analyzes the product, pointing out its pros and cons. An *opinion* is any piece of text with positive or negative implications on some feature of the product. It may be a subjective evaluation (e.g. "the soundtrack is beautiful"), an objective description (e.g."the case is made of plastic") or an affective enunciation (e.g. "I love this movie"), among others.

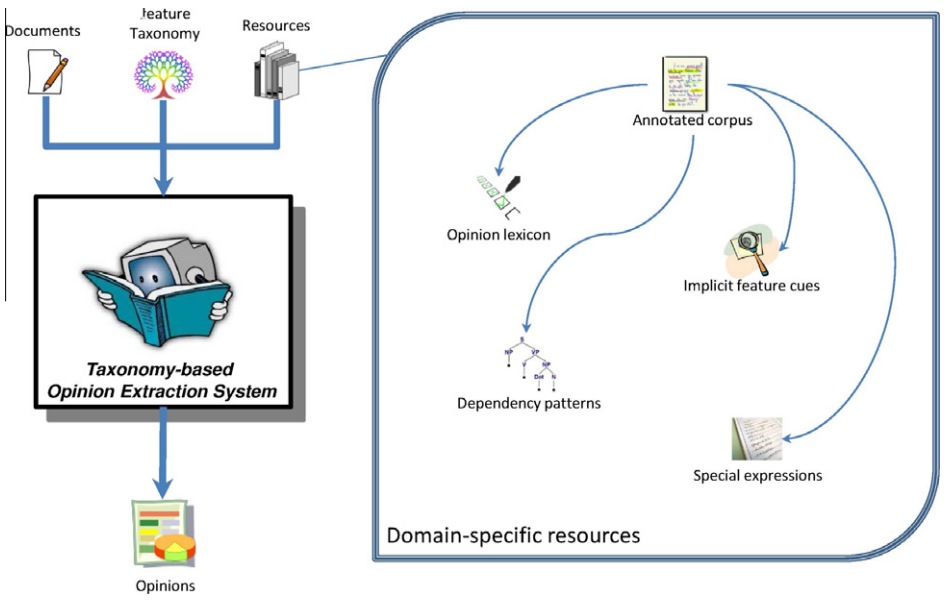

Fig. 1. A conceptual representation of our approach.

Let p be a concrete product, an instance of a product class P. Let $F_P = \{f_1, f_2, \ldots, f_n\}$ be a set of *features* of P. As each feature can be decomposed into sub-features, F_P is indeed a tree, rather than a set. The root of this tree is the product, and the children of a particular node are sub-features of the feature represented by that node. Anyway, for the purposes of defining the task, we call F_P the set of features and sub-features of the product, including all levels of the taxonomy. Let $R_p = \{r_1, r_2, \ldots, r_n\}$ be a set of reviews of p, with each review $r = \{s_1, s_2, \ldots, s_n\}$ being a list of sentences s_j. Let $o_k = (f_i, s_j, polarity)$ be an opinion on feature $f_i \in F_P$ contained in sentence s_j, with *polarity* being *positive* or *negative*.

Our main goal is to discover $O_r = \{o_1, o_2, \ldots, o_n\}$, the set of *opinions* o_k on any $f_i \in F_P$, appearing on any sentence from any review from R. Note that two or more opinions in the same sentence and on the same feature at the same time are allowed. This goal can be divided into two main subproblems: *opinion recognition* and *opinion classification*. Given a sentence, *opinion recognition* consists in identifying the existence of opinions, including determining which features those opinions refer to. *Opinion classification* consists in deciding the polarity of previously recognized opinions.

3.1.1. Opinion evidences

Although the main components of the opinions that we intend to extract are the feature and the polarity, our extraction system uses a more detailed representation of opinions that includes some important words from the sentence containing the opinion: *feature words* and *opinion words*. First, each f_i in F_P has an associated set of *feature words* $FW_{f_i} = \{fw_1, fw_2, \ldots, fw_n\}$, being the set of all the noun phrases that can be used to name f_i in a sentence. Second, given a sentence containing an opinion, let us name *opinion words* to the minimum set of words from the sentence containing that opinion from which you can decide the polarity of that opinion. Then, an *opinion evidence* oe_k is a tuple $(o_k = (f_i, s_j, polarity), fw_u, opw)$, where fw_u and opw are sets of words observed in s_j, being $fw_u \in FW_{f_i}$ some feature words refering the feature f_i and opw some opinion words related to the opinion o_k.

Given a review, our opinion extraction system will try to discover the set of opinion evidences $\{oe_1, oe_2, \ldots, oe_n\}$. Although we are mainly interested in opinions themselves, that is, the features on which opinions have been given and the polarities of those opinions, finding feature and opinion words is a previous step to correctly induce that information. Some examples of opinion evidences from a review about headphones are shown below:

Sentence 1: The sound quality is not impressive, with extremely powerful low frequencies but unclean, not well-defined high-end.

	(Feature, Feature words, Opinion words, Polarity)
oe_1:	(sound quality, sound quality, not impressive, negative)
oe_2:	(bass, low frequencies, powerful, positive)
oe_3:	(treble, high-end, unclean, negative)
oe_4:	(treble, high-end, not well-defined, negative)

Sentence 2: I love them, they are lightweight and look cool!

	(Feature, Feature words, Opinion words, Polarity)
oe_5:	(headphones, them, love, positive)
oe_6:	(size, *none*, lightweight, positive)
oe_7:	(appearance, *none*, looks cool, positive)

Features are usually mentioned by some feature words, but sometimes they are not (e.g. oe_6 and oe_7). Then we say that they are *implicit* features (and name them *implicit opinions*), which have to be deduced by context (opinion words seem to be a good indicator, as we will set out afterwards).

3.2. Domain-specific resources

The central idea of our approach is the availability of resources that capture knowledge about a particular product class and the way people write their reviews on it. To generate these resources, we start from a manual effort (although computer assisted) in order to describe a feature taxonomy and annotate opinion evidences in a corpus of reviews. Then we apply some algorithms in order to extract relevant information about key concepts of the annotated opinion evidences (e.g. opinion words that have been used, correlations between those opinion words and implicit features, syntactic patterns more frequently used, etc.). This knowledge is storaged into a set of domain-specific resources to be used later on by the opinion extraction system. In this section we present a brief

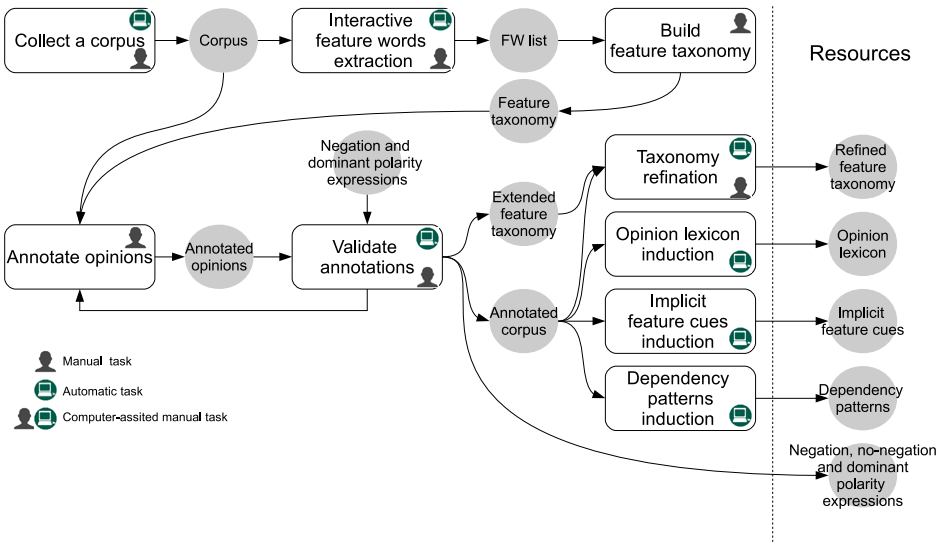

Fig. 2. Resource generation process.

overview of these resources, and briefly explain the process we follow to generate them. A graphical scheme of the whole process is shown in Fig. 2.

3.2.1. Corpus

The first step is to collect a large enough set of reviews of products of the domain we are interested in. There are a lot of good review sites on the Internet, where users write their analysis on products of diverse nature. We are using a corpus extracted from epinions.com, where reviews are written by anonymous users. That means low quality texts: expect a lot of mispellings, out-of-topic reviews, all capital texts, little context due to the short length of texts, questionable grammatical constructions, etc.

The reviews are processed using some NLP external tools: a tokenizer, a part-of-speech tagger, a sentence segmentator and a dependency parser. We are using MaltParser (Hall, 2006) for dependency parsing and Freeling (Atserias et al., 2006) for the rest of processing.

3.2.2. Feature taxonomy

The *feature taxonomy* F_P contains the set of product features for which opinions will be extracted. Besides, each feature f_i comes with a set of feature words, a subset of FW_{f_i}. All these pairs (f_i, FW_{f_i}) are hierarchically organized: the product class itself is the root node of the taxonomy, with a set of features hanging on it. Each feature can be recursively decomposed into a set of subfeatures. A piece of the feature taxonomy for product class *headphones* is shown in Fig. 3. The taxonomy hierarchy will be useful to aggregate opinions to produce summaries. For example, using the headphones taxonomy shown in Fig. 3, you could not only obtain independent summaries of opinions on *bass*, *mids* and *treble* features, but also a summary of opinions on *frequency response*, including the previous ones.

The feature taxonomy is built in two steps. First, a list of feature words is generated from the corpus using a semi-automatic method. Then, an expert produces the taxonomy, grouping feature words by feature and building a hierarchy. The whole process takes no more than a few minutes.

The identification of feature words is made in a semi-automatic way. Our active-learning algorithm is based on two principles:

1. The set of feature words used by people to name a given feature tends to converge (Hu & Liu, 2004b).
2. As we are dealing with opinion texts, *feature words* are often contained in opinions, and therefore near opinion words.

A diagram of the algorithm is shown in Fig. 4. Starting from a few opinion word seeds,[1] it looks for the most frequent feature word candidates appearing in some simple part-of-speech patterns near any of those seeds.[2] Then, an expert is expected to accept or refuse each candidate, beginning with candidates that appear more frequently. When the expert refuses a few candidates, the algorithm looks for new opinion words to be used as seeds, starting from already accepted feature words, and using the same previous patterns. These new seeds are then used to extract new feature word candidates, and the expert is asked again to accept or refuse them. The process continues until the expert refuses a certain number of consecutive candidates.

3.2.3. Annotated corpus

The *annotated corpus* is the most important resource, as all the remaining resources will be extracted from it. It will also be used for evaluation purposes in an experimental setup. Therefore, as many reviews as possible should be annotated. It is desirable to have a uniform distribution of evaluation ratings over the reviews chosen to be annotated. The annotation process consists in marking out opinion evidences, as defined before. When an annotator finishes annotating a few reviews, a validation application is run. It checks annotations for possible errors. For example, the application detects opinion words being employed with semantic orientation opposite to previous annotations, and informs the annotator to prevent possible mistakes. It also warns the annotator about some probably missed opinion evidences, when some words corresponding to (1) feature words or (2) opinion words previously used in opinion evidences with implicit feature have not been annotated.

3.2.4. Negation, non-negation and dominant polarity expressions

When inducing the semantic orientation of opinion words, it is necessary to give an special treatment to some expressions that influence the semantic orientation in a particular way. *Negation expressions* invert the polarity of the semantic orientation of an opinion (e.g., not, hardly, barely,...). *Non-negation expressions* are

[1] *excellent*, *good*, *bad* and *poor*.

[2] The patterns are "*#F* is/are/was/were *#O*" or "*#O #F*", where *#F* is the feature candidate (must be a noun phrase) and *#O* is the opinion seed (must be an adjective).

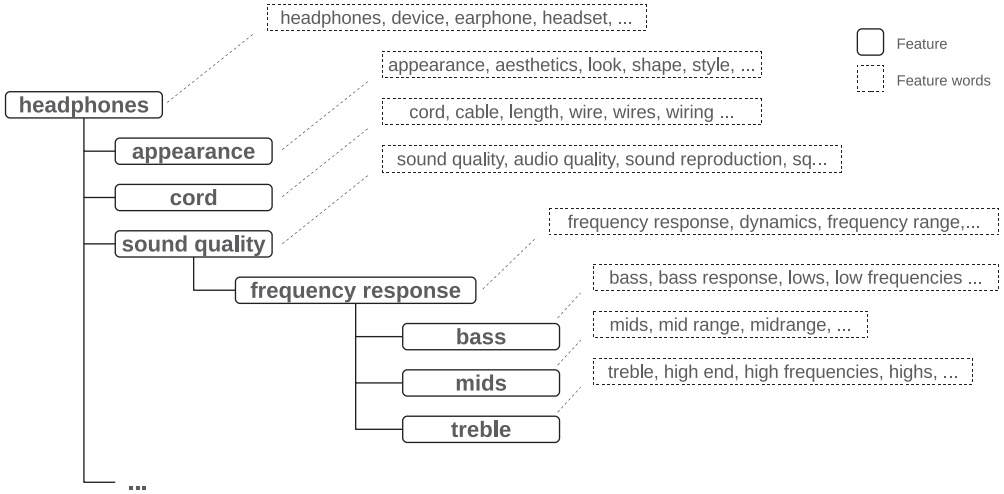

Fig. 3. An extract from the feature taxonomy of the *headphones* domain.

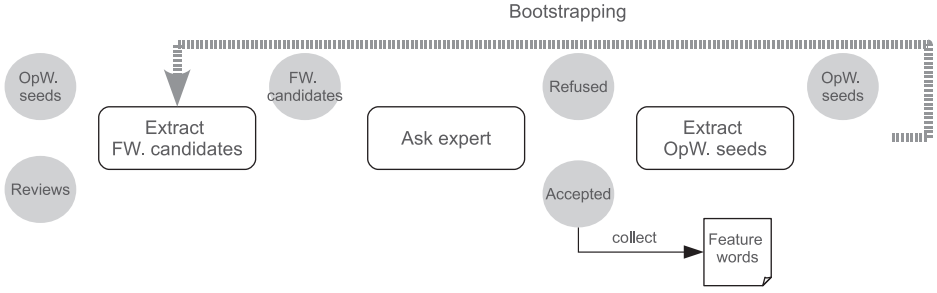

Fig. 4. Interactive identification of feature words.

those containing a negation expression that must be ignored (e.g, not only, not just,...). *Dominant polarity expressions* completely determine the polarity of the semantic orientation of the opinion, no matter which other opinion words take part (e.g., "*enough*" implies positive polarity and "*too*" implies negative polarity). Unlike the rest of resources, negation, non-negation and dominant polarity expressions are domain-independent lists. We start from a small manually-collected list of expressions. The validation application previously introduced allows annotators to add new expressions to the existing ones.

3.2.5. Opinion lexicon

Opinion words are as important or even more than feature words; their presence may help to detect the appearance of opinions, and their semantic orientation may allow us to classify them. The *opinion lexicon* contains some useful information about opinion words and their semantic orientations. For each term (individual word or phrase) whose words have been annotated as opinion words, the opinion lexicon contains two main sets of measures estimated from the annotated corpus: on one side the probabilities of taking part in an opinion evidence for every feature from the taxonomy; and on the other the semantic orientations of terms being used as opinion words for every feature from the taxonomy. A term that always appears in positive opinions receives a semantic orientation of 1.0 (−1.0 if always appears in negative opinions). Other values indicate some level of ambiguity. Note that the absolute value of the measure is not correlated with the intensity of the positive or negative implications of a term; it is rather correlated with the likelihood of that term having positive or negative implications; note that the semantic orientation of a term for a given feature is estimated from all the annotated opinion evidences on that feature or any subfeature of it. Most of the times, an ambiguous value on a feature indicates opposite, unambiguous values on some subfeatures of it. For example, the semantic orientation of *cheap* being used in an opinion on feature *headphones* was estimated as 0.4693, as the result of having negative implications on most subfeatures (*appearance*, *durability*, *sound quality*, etc.) and positive implications on a single but more frequently observed one (*price*). All the entries in the resource include a *support* value (the number of occurrences of the term in the corpus).

Unlike feature words, the set of opinion words for a given domain does not converge easily, so we apply an automatic expansion method to the obtained lexicon, in order to increase recall. The expansion method is detailed in Cruz, Troyano, Ortega, and Enríquez (2011), and is based on a random-walk algorithm explained in Cruz, Vallejo, Enríquez, and Troyano (2012).

3.2.6. Implicit feature cues

If you analyse opinion evidences with implicit features, you will surely notice some correlations between opinion words and features. For example, *comfortable* and *affordable* are positive opinion words commonly used to describe *comfort* and *price* features, respectively. The *implicit feature cues* resource intends to collect this kind of information, that can be very useful in order to discover opinions on implicit features. For each term (individual word or phrase) whose words have been annotated as opinion words, the resource contains estimations of the probabilities of being used as opinion word in an implicit opinion on every feature from the taxonomy. It also includes a *support* value (the number of occurrences of the term in the corpus).

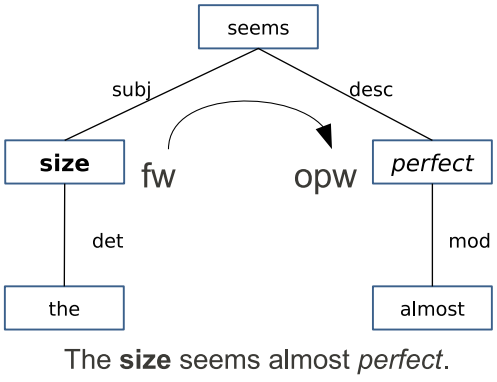

Fig. 5. An example of dependency tree.

3.2.7. Dependency patterns

This resource contains a list of syntantic dependency patterns connecting feature words with opinion words, opinion words between them and opinion words with negation and dominant polarity expressions. Dependency relations connect each word (called *head word*) with its grammatically dependent ones in a sentence. Each relation is tagged with a syntactic function (e.g., *subj* for subjects, or *mod* for modifiers). Given a sentence containing an annotated opinion evidence, the dependency pattern linking a source word to a destination word contains a list of part-of-speech classes and dependency relation tags, corresponding to the path from the first word to the second in the dependency tree. For example, given the sentence *"The size seems almost perfect."*, with the dependency tree shown in Fig. 5, the dependency pattern linking the feature word *seems* to the opinion word *perfect* is $N \rightarrow subj \rightarrow V \rightarrow desc \rightarrow J$,[3] where *N*, *V* and *J* are the part-of-speech classes for *size*, *seems* and *perfect*, respectively. Using this pattern, and given a new feature word, we will be able to discover its potentially related opinion words, whenever the syntactic structure is the same.

The resource includes estimations of precision and recall of each pattern, both real values between 0.0 and 1.0, and also the number of occurrences of each pattern in the corpus. The patterns are feature-specific, so an independent set of patterns (including the previous estimations) are induced for each feature from the taxonomy.

3.3. The opinion extraction system

Our opinion extraction system is comprised of a set of independent abstract components, each one dealing with an independent subtask, which can be combined in a wide variety of pipelines in order to complete the extraction task (Fig. 6). This modular design together with the multiple implementations of each component make up an experimental setup that enables us to test domain-independent *versus* domain-oriented approaches.

Let us give a brief description of these components. The *feature word annotators* discover features explicitly mentioned in the input reviews. Feature words are annotated in a new tentative opinion evidence. On the other hand, the *implicit feature annotators* discover implicitly mentioned features, annotating the opinion words related to that feature in a new tentative opinion evidence. Given some annotated feature words, the *opinion word linkers* intend to link them to dependent tentative opinion words. The *negation expression* and *dominant polarity expression linkers* start from previously identified opinion words and look for negative or dominant polarity expressions which might be associated with them. The *opinion classifiers* decide the polarity of tentative opinion evidences. Some of these tentative opinion evidences will be deleted by the *opinion filters* (e.g., those with a semantic orientation value lower than a certain threshold). Some others may be also deleted or modified by the *overlapping opinion fixers*, that solve conflicts between several opinion evidences (e.g., two opinion evidences using the same opinion or feature words). Finally, the *opinion extractor pipeline* component allows to define any combination of concrete implementations of components to perform the opinion recognition and classification. It takes a list of reviews as input, processed using the same NLP external tools that were previously introduced. It may also take the domain-specific resources, if some of the concrete components participating in the pipeline make use of them.

We have implemented a full set of resource-based concrete components, and also a few domain-independent, resource-free concrete components, in order to experimentally measure the contribution of the resources to the system.

3.3.1. Resource-based components

There are resource-based implementations of the explicit and implicit feature annotators, the opinion word linker and the opinion classifier:

- *Taxonomy-based feature word annotator*: it uses the feature words contained in the feature taxonomy to annotate tentative explicit opinion evidences. The component can be configured to require the feature words found to be a noun phrase.
- *Cue-based implicit feature annotator*: it uses the implicit feature cues to annotate tentative implicit opinion evidences. It can be configured to ignore cues with a value of support or probability lower than some thresholds.
- *Dependency-based opinion word linker*: it uses the dependency patterns to find new opinion words, starting from previously annotated feature words or from previously annotated opinion words. It can be configured to ignore those patterns with a value of support, precision or recall lower than some thresholds.
- *Dependency-based special expression linker*: it uses dependency patterns to find negation and dominant polarity expressions related to previously annotated opinion words. As the previous component, it can be configured to ignore those patterns with a value of support, precision or recall lower than some thresholds.
- *Lexicon-based opinion classifier*: it uses the opinion lexicon to decide the polarity of previously annotated opinion evidences. The terms composed of consecutive opinion words are searched in the lexicon; if the value of probability is greater than a configurable threshold, the semantic orientation value is used to decide polarity. If more than one term have been found for a given opinion evidence, the mean of their semantic orientation estimations is used. The component also takes into account the negation and dominant polarity expressions participating, if any. Besides, it can be configured to use WordNet (Fellbaum, 1998) for semantic expansion; in this case, the synonyms and antonyms of a term appearing in the lexicon will receive the same values of probability and semantic orientation.

3.3.2. Domain-independent components

We have implemented domain-independent versions of each previous component, based on the methods used in some of the related works:

- *PMI-based implicit feature annotator*: it uses the Pointwise Mutual Information algorithm (PMI) (Turney, 2002) to compute the semantic proximity of adjectives in the reviews and the features from the feature taxonomy. If the PMI between an adjective from the review and some feature is greater than a configurable threshold, a tentative implicit opinion evidence is annotated.

[3] The pattern is actually represented by two lists, one corresponding to the ascending path and the other to the descending one: $N \rightarrow subj \rightarrow V$ and $V \rightarrow desc \rightarrow J$.

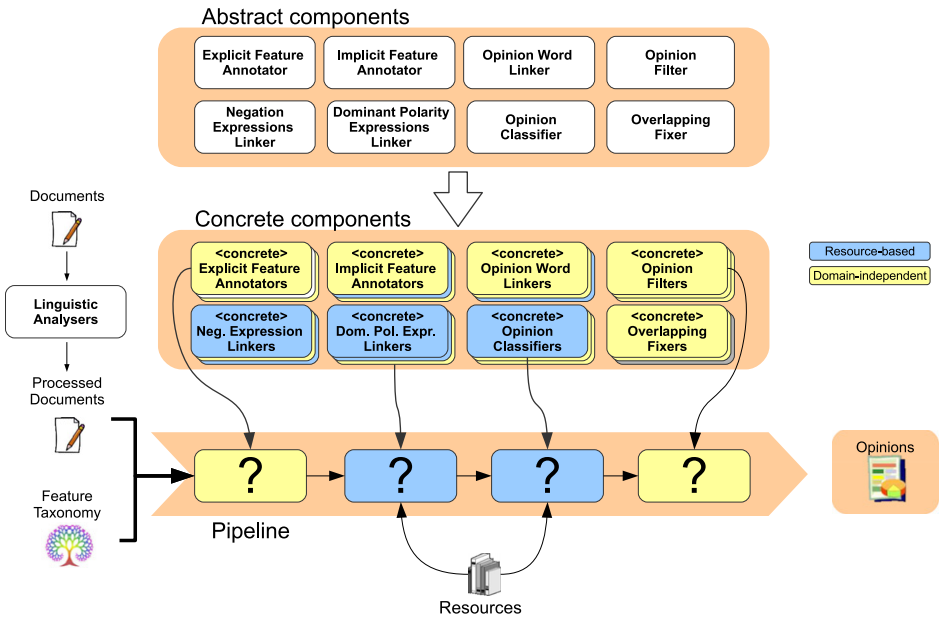

Fig. 6. System architecture.

- *Window-based opinion word linker*: it uses a context window around previously annotated feature words to find related opinion words. The size of the context window and the allowed part-of-speech tags can be set.
- *Window-based special expression linker*: it uses a context window around previously annotated opinion words to find related negation and dominant polarity expressions. The size of the context window and the allowed part-of-speech tags can be configured.
- *Opinion classifiers*: they compute the semantic orientation and decide the polarity of previously annotated opinion evidences using different techniques. We have implemented three different components: the *WordNet-based opinion classifier* uses the distance between the opinion words and some seeds in WordNet, in a similar way to Kamps, Marx, Mokken, and De Rijke (2004); the *PMI-based opinion classifier* uses Pointwise Mutual Information algorithm (Turney, 2002) between the opinion words and some seeds; and the *SentiWordNet-based opinion classifier* uses SentiWordNet 3.0 (Baccianella, Esuli, & Sebastiani, 2010), a state of the art domain-independent opinion lexicon, containing positivity and negativity scores for WordNet synsets. A minimum semantic orientation threshold can be set in all cases.

4. Experiments

In this section we describe the experiments performed using a set of user-generated reviews of three different domains: headphones, hotels and cars. The definition of our task does not match with previously published resources, for example Hu and Liu (2004b), mainly due to the absence of a taxonomic organization of features. So we collected and annotated our own dataset. Since the results over this new dataset cannot be directly compared to the results previously reported by other works, we performed some experiments with pipelines exclusively composed of domain-independent, resource-free components, using techniques similar to previous related works; these pipelines serve as baseline and allow us to measure the benefits of using domain-specific resources to solve the opinion recognition and classification task.

4.1. Data

We collected user-generated reviews of three different domains (headphones, hotels and cars) from epinions.com, a website specialized in product reviews written by customers. We built a feature taxonomy for each domain, using the proposed method, what took less than an hour per domain. Then, all the opinion evidences in the reviews were annotated and validated. The annotation and validation process took about one hour each fifteen reviews. Some statistics about reviews, feature taxonomies and annotations are shown in Table 1. Note the low proportion of sentences containing opinions (about one out of four); in comparison, the datasets used in most of the previous works (Ding et al., 2008; Hu & Liu, 2004b; Liu, Hu, & Cheng, 2005; Popescu & Etzioni, 2005) contain a more balanced set of sentences with and without opinions (about one out of two). Although we could artificially balance the corpus, we prefer using the reviews just as they were extracted, as we are interested in measuring the accuracy of our approach when applied to real user-generated texts. The dataset is available for public use.[4]

4.2. Experimental setup

All the experiments were done using 10-fold cross-validation. The results reported for each experiment are the average results obtained in ten different runs, taking each time a different subset as testing set and the remaining nine subsets as training set (to induce the domain-specific resources and tune the configuration parameters of each component of the pipelines).

4.2.1. Pipelines

In order to measure the contribution of the domain-specific resources, we carried out two sets of experiments:

- *Resource-free experiments*: three pipelines were defined using the domain-independent, resource-free concrete components explained before (except for the taxonomy-based feature word annotator, which uses the domain-specific feature taxonomy).

[4] http://www.lsi.us.es/~fermin/index.php/Datasets.

Table 1
Dataset statistics.

	Headphones	Hotels	Cars
Reviews	587	988	972
Words	139331	631442	493459
Sentences	8151	33853	26307
Sentences containing opinions	2545	7339	5989
Number of features in taxonomy	31	60	91
Opinion evidences...	3897	11054	8519
...positive/negative polarity	72.24%/27.76%	69.41%/30.59%	74.36%/25.64%
...implicit/explicit feature	36.62%/63.38%	13.24%/86.76%	37.25%/62.75%

We conducted experiments with three similar pipelines, but using a different domain-independent opinion classifier in each one.

- *Resource-based experiments*: we defined a pipeline using the resource-based concrete components explained before.

The pipelines are shown in Fig. 7.

4.2.2. Evaluation measures

For each run, after applying each pipeline to the testing set, we evaluated the opinion recognition and classification as individual tasks. First, we measured the proportion of correctly recognized opinions from the total of extracted opinions (*precision*) and from the total of annotated opinions (*recall*); an extracted opinion is a correctly recognized one if it matches an annotated opinion on the same feature (and that opinion has not been used to validate another extracted opinion). Then, we measured the proportion of correctly classified opinions (*accuracy*).

For each pair of precision and recall scores, we calculate F_1 and $F_{\frac{1}{2}}$ scores, which are weighted averages of them. In the F_1 score, precision and recall are evenly weighted. The $F_{\frac{1}{2}}$ score weights precision higher than recall.

4.2.3. Parameter tuning

For each pipeline, the configuration parameters of the components were tuned using the training set to optimize F_1, in one hand, and $F_{\frac{1}{2}}$, in the other. In the first case, we search for opinion extraction systems with a balanced precision and recall; in the second case, we obtain systems with a higher precision, but a lower recall. We carried out experiments with both versions of pipelines; we think that, for some applications, it would be better to extract some reliable opinions, although missing some others, rather than

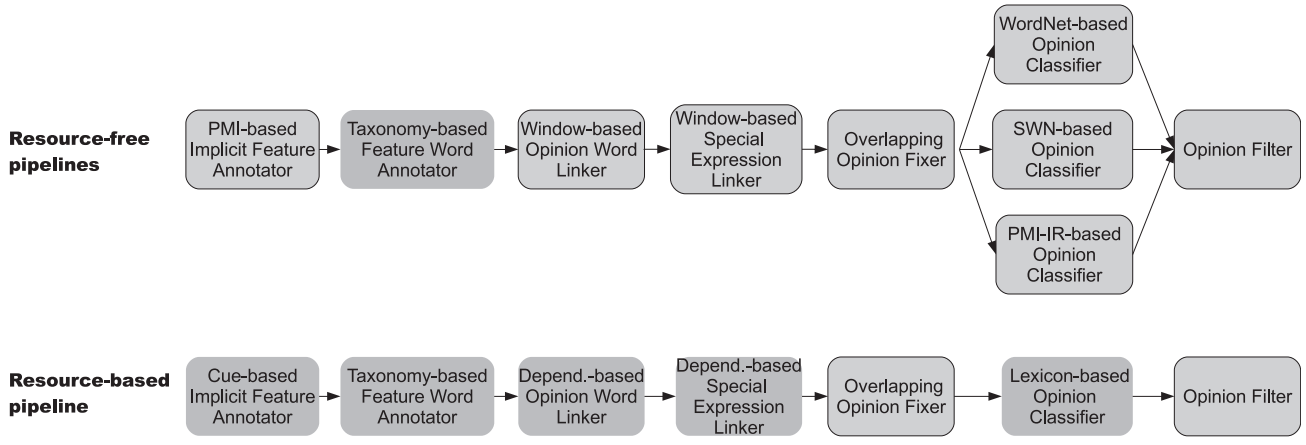

Fig. 7. Experimental pipelines.

Table 2
Results for the opinion recognition and classification task. Components' parameters optimized for F_1.

Domain	Pipeline	Opinion Clasiffier	Opinion Recognition			Opinion Classif. Acc.
			P	R	F_1	
Headphones	Resource-free	WordNet	0.5424	0.5053	0.5232	0.7793
		SWN	0.5373	0.5375	0.5374	0.7821
		PMI	0.5882	0.4794	0.5283	0.8354
	Resource-based	Lexicon	**0.7115**	**0.6617**	**0.6857**	**0.9345**
Hotels	Resource-free	WordNet	0.4811	0.6012	0.5345	0.743
		SWN	0.5126	0.6137	0.5586	0.7521
		PMI	0.5136	0.5392	0.5261	0.7922
	Resource-based	Lexicon	**0.6782**	**0.7388**	**0.7072**	**0.9114**
Cars	Resource-free	WordNet	0.4742	0.5433	0.5064	0.7717
		SWN	0.4854	0.5536	0.5173	0.7577
		PMI	0.4872	0.4918	0.4895	0.8235
	Resource-based	Lexicon	**0.7169**	**0.7296**	**0.7232**	**0.9287**

Table 3
Results for the opinion recognition and classification task. Components' parameters optimized for $F_{\frac{1}{2}}$.

Domain	Pipeline	Opinion Clasiffier	Opinion Recognition			Opinion Classif. Acc.
			P	R	$F_{\frac{1}{2}}$	
Headphones	Resource-free	WordNet	0.6534	0.2589	0.5008	0.9033
		SWN	0.6758	0.2992	0.5399	0.8937
		PMI	0.6744	0.3643	0.5763	0.8688
	Resource-based	Lexicon	**0.7869**	**0.5662**	**0.73**	**0.9503**
Hotels	Resource-free	WordNet	0.5854	0.2426	0.4565	0.8503
		SWN	0.7104	0.2853	0.5473	0.8731
		PMI	0.5924	0.4092	0.5437	0.8323
	Resource-based	Lexicon	**0.7673**	**0.584**	**0.722**	**0.9366**
Cars	Resource-free	WordNet	0.6224	0.2395	0.4716	0.8853
		SWN	0.6709	0.259	0.509	0.8972
		PMI	0.5534	0.38	0.5071	0.8611
	Resource-based	Lexicon	**0.7836**	**0.609**	**0.7411**	**0.95**

to obtain a big set of opinions, containing almost all the existent opinions but including a high number of inexistent ones.

4.2.4. Computational issues

The resource-inducing algorithms, the extraction system and the parameter tuning were all implemented in Java 1.6 and run in a Intel Core 2 Duo 2.13 MHz with 4 GB of RAM. The parameter tuning algorithm is the most time-consuming (about an hour per pipeline), although it is only run once before the exploitation of the extraction system. The resource-inducing algorithms and the extraction system are both really fast, being the preprocessing linguistic tools the only bottleneck; even so, we have obtained processing rates of about 50–100 sentences per second. We think they are low enough time to process large amounts of text on-the-fly, as required in the context of the fast-growing Web 2.0.

4.3. Results

The results obtained by each pipeline, F_1 and $F_{\frac{1}{2}}$ optimized versions, are shown in Tables 2 and 3, respectively. For each domain, the first three rows correspond to resource-free pipelines, each one including an opinion classifier based on WordNet, SentiWordNet or PMI algorithm. The fourth row corresponds to the resource-based pipeline.

4.3.1. Resource-free vs. Resource-based results

Our initial hypothesis is fully supported by the results. The resource-based pipelines, both F_1 and $F_{\frac{1}{2}}$ optimized versions, obtain much better results than the resource-free pipelines in the three domains, with average increments of +0.181 in F_1, +0.214 in $F_{\frac{1}{2}}$ and +0.072 in accuracy. The improvement seems to be greater in those domains with a more complex feature taxonomy. The reason why our resource-based approach works better than other domain-independent state-of-art approaches is that the different values included in the resources are specifically estimated for each domain and each feature of the taxonomy. For example, the term *small* receives positive values of semantic orientation in the *headphones* opinion lexicon, and negative values in the *hotels* opinion lexicon (especially when the term is applied to the feature *room*). However, domain-independent state-of-art classifiers compute a single value of semantic orientation, which tries to agglutinate all the possible uses of the term.

We think that the results obtained by our system using the domain-specific resources are fairly good, considering the complexity of the problem definition. Using the F_1 optimized version of the resource-based pipeline, about 70% of opinions are correctly recognized and mapped to the feature taxonomy, and about 92% of them are correctly classified into positive or negative classes. If we are more concerned about the accuracy of the system, about 58% of opinions can be correctly recognized, with a lower rate of false positives (using the $F_{\frac{1}{2}}$ version of the pipeline). In a following section we propose an example of use of the extracted opinions, in order to determine whether or not they are useful from a qualitative point of view.

Table 4
Influence of dataset size in headphones domain. First values of F_1 and accuracy which surpass the resource-free pipeline are highlighted.

Number of reviews	Estimated annotation time (hours)	Opinion recognition F_1	Opinion classification Acc.
9	0.6	0.48	0.9348
18	1.2	0.5561	0.9377
27	1.8	0.5916	0.9249
36	2.4	0.6061	0.9337
45	3	0.6162	0.9345
(best resource-free pipeline)		0.5283	0.8354

4.3.2. Influence of dataset size and temporal issues

The previous results confirm that the availability of domain-specific resources allows us to build much better feature-based opinion extraction systems. But it involves some manual effort to annotate a set of documents in order to induce those resources for a given domain. In our experience, it takes about 35 h to annotate the 528 reviews used to induce resources in the experiments with the headphones domain (note that we used 10-fold cross validation in our experiments, so 90% of the 587 reviews were used to induce resources in each run). From our point of view, this is a worthwhile effort, given the improvement achieved with respect to the domain-independent approaches.

We did some experiments for the headphones domain using training sets of different sizes, in order to measure the influence of the number of available annotated documents in the results obtained by the resource-based pipeline (see Table 4). Again, all the experiments were done using 10-fold cross-validation, but this time taking only one to five reviews from each of the nine training subsets used for training, and all the reviews from the remaining subset for testing. The results suggest a still significant improvement over the best resource-free pipeline using only 45 reviews to induce the resources (+0.0879 in F_1 and +0.0991 in accuracy). This implies only 3 h to annotate the documents involved in the induction of the resources.

4.3.3. Opinion aggregation and visualization

The precision and recall measures used in our tests allow us to compare the different solutions proposed, and to represent the

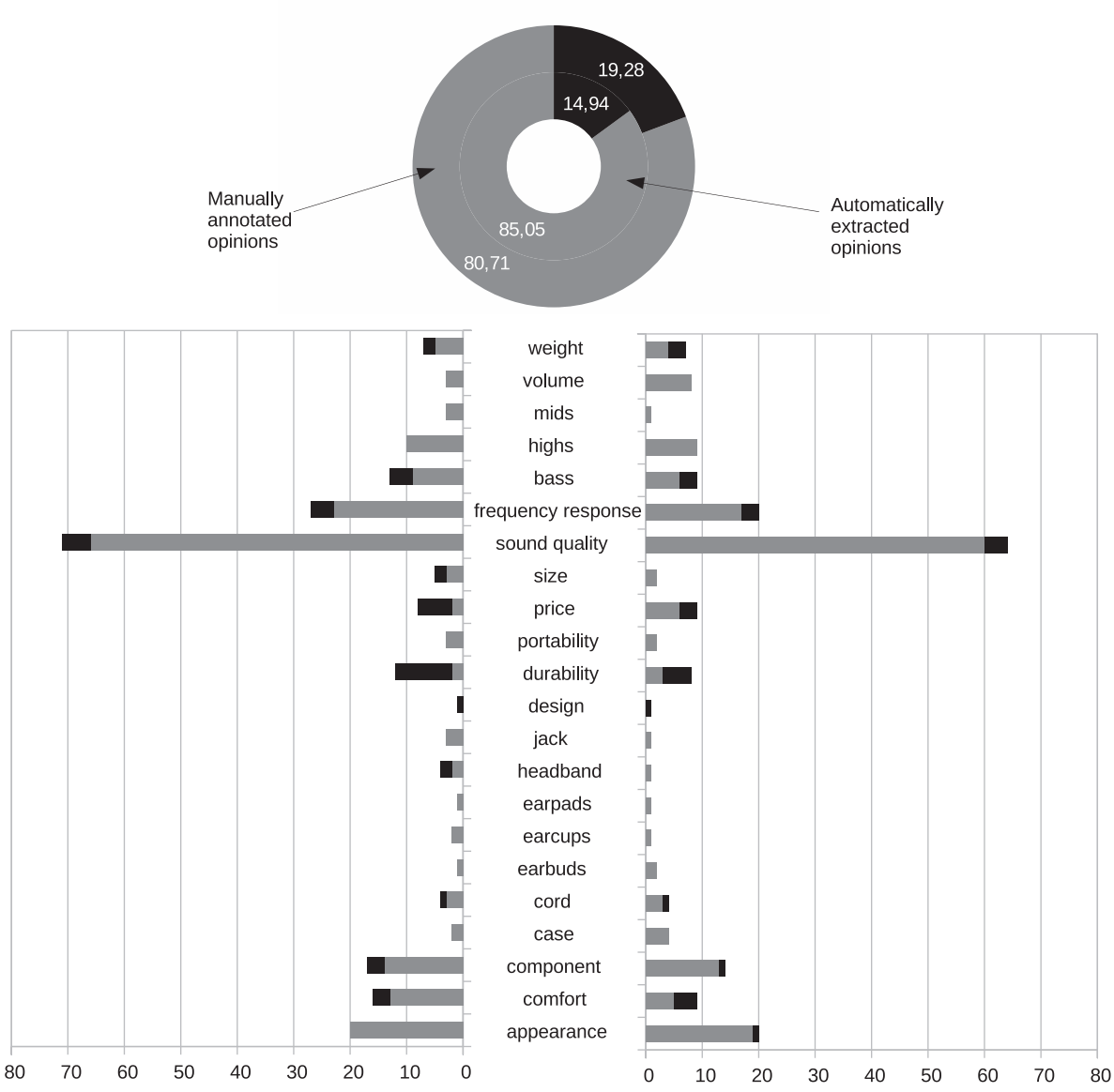

Fig. 8. Visualization of manually annotated opinions (left) vs. automatically extracted opinions (right) from 26 reviews of Sony MDR-V700DJ.

effectiveness of the system as a single value (for example, by the F_1 score). But at the sight of this single value, a question arises: are the extracted opinions good enough for being used in a real application? The answer to this question depends on the application. Let us consider the aggregation and visualization of the extracted opinions.

Fig. 8 shows a possible visual summarization of the opinions from 26 reviews about a concrete product (Sony MDR-V700DJ headphones). It represents the general opinion of users about each feature from the headphones feature taxonomy, showing the proportion of positive (grey) and negative (black) opinions. The size of the bars indicates the number of opinions found about the feature. The figure has been constructed by accounting positive and negative opinions about each feature, using the specialization relationships from the taxonomy. In this way, opinions on the feature *bass* are counted as opinions on feature *frequency response*, which in turn are counted as opinions on *sound quality*.

In order to answer the question about the practical usefulness of our system, we constructed two versions of the chart: one using the annotated opinions from the dataset, and another one using the opinions extracted by the resource-based pipeline. Although there are some differences, most of them are irrelevant: a person who wishes to get an idea about the benefits and shortcomings of the product in question, will draw similar conclusions from both versions of the chart. For example:

- Most of the opinions about the product are positive.
- The sound quality is the most commented feature by users, with overall positive opinions about it. Within it, the bass seem to be the weak point for a few users, while the midrange and treble are good for all the users who have commented on it.
- Most of the users like the appearance of the device.
- The weakest point seems to be the durability or quality of construction: a majority of users have a bad opinion about it.

We conclude that the opinions extracted by our system are undoubtedly useful for opinion aggregation and visualization applications.

5. Conclusions

The feature-based opinion extraction task is intended to extract structured representations of opinions from user-generated texts.

In order to facilitate the aggregation of the extracted opinions, we have proposed a redefinition of the task based on feature taxonomies, semantic representations of parts and attributes of objects. Since some of the entities involved in the problem are context-dependent (at domain or feature level), we have defined some domain-specific, feature-level resources which capture valuable knowledge about how people express opinions on each feature for a given domain. A modular system has been designed, consisting of a number of components that address different parts of the task and can be combined to form various pipelines. We have performed some experiments in three different domains with two different pipelines, one mainly formed by resource-free components and the other one by resource-based components. The results obtained by the latter were significantly better, which confirms the importance of the domain in the feature-based opinion extraction task. Although a set of documents must be annotated in order to automatically induce the resources from them, it does not require much effort: annotating only 45 documents (which takes about three hours) led to an improvement of nearly 10 percentage points in both F_1 and accuracy, with respect to the domain-independent approaches. One possible improvement would be to adapt the system from one domain to another. Starting from the resources for a particular domain, we think that they can be exploited to generate the resources for a new domain. Resolving these issues would lead to a diminishing effort in adapting the system to new domains.

The average results obtained by our system was 0.71 for opinion recognition F_1 and 0.9248 for opinion classification accuracy (0.7310 for $F_{\frac{1}{2}}$ and 0.9456 for accuracy if using the $F_{\frac{1}{2}}$ optimized pipeline). Beyond the quantitative results, we have shown an example of aggregation and visualization of opinions for a particular product, demonstrating the usefulness of the taxonomy-based opinions extracted by our system.

References

Agrawal, R., & Srikant, R. (1994). Fast Algorithms for Mining Association Rules. URL <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.40.7506>.

Atserias, J., Casas, B., Comelles, E., González, M., Padró, L., & Padró, M. (2006). Freeling 1.3: Syntactic and semantic services in an open-source nlp library. In *Proceedings of the international conference on language resources and evaluation*. ELRA, Genoa, Italy.

Baccianella, S., Esuli, A., & Sebastiani, F. (2010). Sentiwordnet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining. In *Proceedings of the seventh conference on international language resources and evaluation. ELRA*.

Blei, D. M., Ng, A. Y., & Jordan, M. I. (2003). Latent Dirichlet allocation. *Journal of Machine Learning Research, 3*, 993–1022.

Brody, S., & Elhadad, N. (2010). An unsupervised aspect-sentiment model for online reviews. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics. HLT '10* (pp. 804–812). Association for Computational Linguistics, Stroudsburg, PA, USA. URL <http://portal.acm.org/citation.cfm?id=1857999.1858121>.

Cruz, F. L., Troyano, J. A., Ortega, F. J., & Enríquez, F. (2011). Automatic expansion of feature-level opinion lexicons. In *Proceedings of the second workshop on computational approaches to subjectivity and sentiment analysis* (pp. 125–131). ACL, Portland, Oregon. URL <http://www.aclweb.org/anthology/W11-1716>.

Cruz, F. L., Vallejo, C. G., Enríquez, F., & Troyano, J. A. (2012). Polarityrank: Finding an equilibrium between followers and contraries in a network. *Information Processing and Management, 48*(2), 271–282.

Dave, K., Lawrence, S., & Pennock, D. M. (2003). Mining the peanut gallery: Opinion extraction and semantic classification of product reviews. In *Proceedings of WWW* (pp. 519–528).

Ding, X., Liu, B., & Yu, P. S. (2008). A holistic lexicon-based approach to opinion mining. In *WSDM '08: Proceedings of the international conference on Web search and web data mining* (pp. 231–240). New York, NY, USA: ACM.

Fellbaum, C. (Ed.). (1998). *WordNet: An electronic lexical database*. MIT Press.

Hall, J. (2006). MaltParser – An architecture for inductive labeled dependency parsing.

Hu, M., & Liu, B. (2004). Mining and summarizing customer reviews. In *Proceedings of the ACM SIGKDD conference on knowledge discovery and data mining (KDD)* (pp. 168–177).

Hu, M., & Liu, B. (2004). Mining opinion features in customer reviews. In *Proceedings of AAAI* (pp. 755–760).

Jo, Y., & Oh, A. H. (2011). Aspect and sentiment unification model for online review analysis. In *Proceedings of the fourth ACM international conference on Web search and data mining. WSDM '11* (pp. 815–824). New York, NY, USA: ACM. http://dx.doi.org/10.1145/1935826.1935932.

Kamps, J., Marx, M., Mokken, R. J., & De Rijke, M. (2004). Using wordnet to measure semantic orientation of adjectives. In *Proceedings of the 4th International Conference on Language Resources and Evaluation (LREC 2004), (Vol. 26, pp. 1115–1118)*. URL <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.6.2534>.

Liu, B., Hu, M., & Cheng, J. (2005). Opinion observer: Analyzing and comparing opinions on the web. In *Proceedings of WWW*.

Liu, B., & Zhang, L. (2012). A survey of opinion mining and sentiment analysis. In C. C. Aggarwal & C. Zhai (Eds.), *Mining text data* (pp. 415–463). US: Springer.

Pang, B., & Lee, L. (2008). Opinion mining and sentiment analysis. *Foundations and Trends in Information Retrieval, 2*(1–2), 1–135.

Popescu, A. -M., & Etzioni, O. (2005). Extracting product features and opinions from reviews. In *Proceedings of the human language technology conference and the conference on empirical methods in natural language processing (HLT/EMNLP)*.

Titov, I., & McDonald, R. (2008). A joint model of text and aspect ratings for sentiment summarization. In *Proceedings of ACL-08: HLT. Association for computational linguistics*, Columbus, Ohio. URL <http://www.aclweb.org/anthology/P/P08/P08-1036>.

Titov, I., & McDonald, R. (2008b). Modeling online reviews with multi-grain topic models. In *Proceeding of the 17th international conference on World Wide Web. WWW '08* (pp. 111–120). New York, NY, USA: ACM. http://dx.doi.org/10.1145/1367497.1367513.

Turney, P. (2002). Thumbs up or thumbs down? Semantic orientation applied to unsupervised classification of reviews. In: *Proceedings of the ACL* (pp. 417–424).

Zhai, Z., Liu, B., Xu, H., & Jia, P. (2010). Grouping product features using semi-supervised learning with soft-constraints. In *Proceedings of the 23rd international conference on computational linguistics. Coling 2010 Organizing Committee*, Beijing, China, (pp. 1272–1280). URL <http://www.aclweb.org/anthology/C10-1143>.

Zhao, W. X., Jiang, J., Yan, H., & Li, X. (2010). Jointly modeling aspects and opinions with a maxent-lda hybrid. In *Proceedings of the 2010 conference on empirical methods in natural language processing. EMNLP '10* (pp. 56–65). Association for Computational Linguistics, Stroudsburg, PA, USA. URL <http://portal.acm.org/citation.cfm?id=1870658.1870664>.