A Knowledge-Rich Approach to Feature-Based Opinion Extraction from Product Reviews

Fermín L. Cruz
University of Seville
Avda. Reina Mercedes s/n.
41012 Seville, Spain
fcruz@us.es

José A. Troyano
University of Seville
Avda. Reina Mercedes s/n.
41012 Seville, Spain
troyano@us.es

Fernando Enríquez
University of Seville
fenros@us.es

F. Javier Ortega
University of Seville
javierortega@us.es

Carlos G. Vallejo
University of Seville
vallejo@us.es

ABSTRACT

Feature-based opinion extraction is a task related to information extraction, which consists of extracting structured opinions on features of some object from reviews or other subjective textual sources. Over the last years, this prob-lem has been studied by some researchers, generally in an unsupervised, domain-independent manner. As opposed to that, in this work we propose a redefinition of the problem from a more practical point of view, and describe a domain-specific, resource-based opinion extraction system. We fo-cus on the description and generation of those resources, and briefly report the extraction system architecture and a few initial experiments. The results suggest that domain-specific knowledge is a valuable resource in order to build precise opinion extraction systems.

1. INTRODUCTION

Within *sentiment analysis*, a modern subdiscipline of *natural language processing* which deals with subjectivity, affects and opinions in texts (a good survey on this subject can be found in [10]), the *feature-based opinion extraction* is a task related to information extraction, which consists in extracting structured opinions on features of some object from subjective texts. Some researchers have proposed a few approaches to this task, often unsupervised, domain-independent ones. In this work, we present a domain-specific, resource-based approach. We are interested in the construction of reliable opinion extraction systems, with a practical point of view of the problem. In order to do that, we propose a redefinition of the problem, including a more specific characterization of the opinion concept, and a methodology to build opinion extraction systems for a given product class in a semisupervised way. We concentrate our attention on extracting opinions from product reviews, although the same ideas could be applied to extract opinions from any other textual sources, like blogs or forums.

Besides the problem definition, in this paper we are mainly focusing on describing the resources that capture the domain knowledge. A brief description of the system architecture and a few initial experimental results are also discussed.

2. FEATURE-BASED OPINION EXTRACTION

2.1 Previous works

Over the last six years, feature-based opinion extraction from product reviews has been studied by a few researchers. The first definition of the problem can be found in [4]: "*Given a set of customer reviews of a particular product, the task involves three subtasks: (1) identifying features of the product that customers have expressed their opinions on (called product features); (2) for each feature, identifying review sentences that give positive or negative opinions; and (3) producing a summary using the discovered information.*". This definition, with further formalization of the entities participating (features, opinions, feature words, opinion words, etc.), has been used in works from the same authors and others ([5],[11],[9],[14],[2]). Most experiments in these works were performed on FBS customer review datasets[1], but each researcher reports results for different subtasks, so these results are difficult to compare. Most of them measure the accuracy of feature extraction on the one hand, and the semantic orientation estimation for a given feature/sentence pair on the other. The semantic orientation [7] or polarity of a term indicates the positive or negative implications of that term being used in an opinion.

We think the biggest shortcoming of all these proposals,

[1] http://www.cs.uic.edu/ liub/FBS/sentiment-analysis.html

and what makes the problem a really hard one, is their generality. First, you are supposed to identify any feature appearing in a review, no matter which kind of product being reviewed. And still it is possible that the list of features obtained is too big to produce a useful summary of pros and cons of the product. None of these works take the domain into account, so that 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. Moreover, the opinion concept itself is vaguely defined, so you are supposed to deal with a wide range of subjective phenomena and to extract any kind of opinion expressions, from the simplest to the most complex. Finally, all of this work is tried to be done in a completely automatic way, without (nearly) any manually collected resources.

2.2 Our approach

As opposed to a general, domain-independent opinion extraction task, we propose a redefinition of the problem from a more practical point of view. The main guidelines of our approach are:

- Before building the system, a domain or product category must be chosen.
- Only features contained in a domain-specific taxonomy will be considered.
- We are not handling all types of opinion, but only a small, well-defined subset.
- The extraction process is assisted by domain-specific supporting resources. Some of these resources are automatically induced from a corpus of annotated reviews; some others are manually generated by an expert (including an annotated corpus and a feature taxonomy), with some computational assessment.

A conceptual representation of our proposal is shown in figure 1. In the next sections, we define the problem and which kind of opinions we are focusing on, describe the supporting resources and system architecture, and show the results of some early experiments.

3. A KNOWLEDGE-RICH OPINION EXTRACTION SYSTEM

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 cellular phone, a movie, a hotel,etc. 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 a positive or negative evaluation on a feature of the product (e.g. "the soundtrack is beautiful"), or a description about that feature with positive or negative implications (e.g. "the rooms are large"). This definition of opinion is just an intuitive idea, but it is too abstract in order to deal with, and will be refined later.

Let p be a concrete product, an instance of a product class P. Let $F_P = \{f_1, f_2, ..., f_n\}$ be a set of *features* of P, including components and attributes. F represents the key parts and properties we are interested in. Let $R_p = r_1, r_2, ..., r_n$ be a set of reviews of p, with each review $r = \{s_1, s_2, ..., s_n\}$, formed by a list of sentences s_j. Let $o_k = (f_i, s_j, soLabel)$ be an opinion on feature f_i contained in sentence s_j, with $soLabel$ being *positive* or *negative* according to the polarity of the opinion.

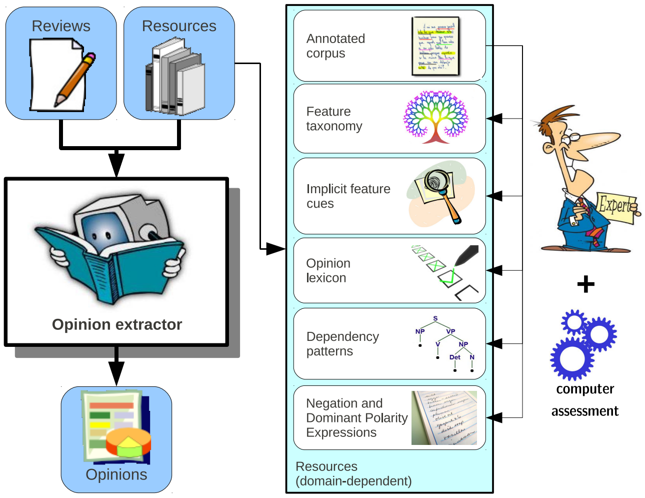

Figure 1: A knowledge-rich approach: Conceptual summary

Our main goal is to discover $O_r = \{o_1, o_2, ..., o_n\}$, the set of *opinions* o_k on any f_i of F_P, appearing on any sentence of any review from R. This goal can be divided into two main subproblems: *opinion recognition* and *opinion classification* (using a nomenclature similar to the *named-entity recognition* and *classification* in *natural language processing*). Given a sentence, *opinion recognition* consists in identifying the existence of opinions, including determining the feature that opinion refers to. *Opinion classification* consists in deciding the polarity of previously recognized opinions.

3.1.1 Opinion evidences

In order to characterize which kind of opinions will be suitable for extraction, and also to enable the induction of lexical and syntactic knowledge from an annotated corpus, let us define an *opinion evidence* as an specialization of an *opinion*, as defined before. An opinion evidence can be seen as a syntactic realization of an opinion. First, each f_i in F_P has an associated set of *feature words* $FW_{f_i} = \{fw_1, fw_2, ..., fw_n\}$, being the set of all the noun phrases that can be used to name f_i in a sentence. Second, given an opinion, let us name *opinion words* to the minimun 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 $(o_k = (f_i, s_j, soLabel), fw_u, opw)$, where $fw_u \in FW_{f_i}$, fw_u is observed in s_j, and opw are the *opinion words* contained in s_j and related to the opinion o_k.

Given a review, an *opinion extraction system* will try to discover the set of opinion evidences $OE = \{oe_1, oe_2, ..., oe_n\}$. 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 will help us to correctly induce that information.

Some examples of opinion evidences, corresponding to a review of a pair of headphones, are shown in figure 2. In the first sentence, an opinion evidence on feature *sound quality* has been annotated, with those exact words referring the

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

	(Feature,	*Feat. words,*	*Op. words,*	*SO label)*
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 looks cool!

	(Feature,	*Feat. words,*	*Op. words,*	*SO label)*
oe_5:	(headphones,	them,	love,	positive)
oe_6:	(size,	*none*,	lightweight,	positive)
oe_7:	(appearance,	*none*,	looks cool,	positive)

Sentence 3:
The cord is durable but too long: I always have to untwist it every time I get them out off my pocket.

	(Feature,	*Feat. words,*	*Op. words,*	*SO label)*
oe_8:	(cord,	cord,	durable,	positive)
oe_9:	(cord,	cord,	too long,	negative)

Figure 2: Examples of opinion evidences

feature. The opinion words are *not impressive*, because the polarity of the opinion, that is negative, can be deduced observing only those two words. Note that in oe_2, *powerful* is the only opinion word, because it forms the *minimum* set of words which you can decide polarity from. In this case, *extremely* affects the intensity of the opinion, but not the polarity [2]. This "minimalistic" approach will allow us to obtain a better statistical representativity in order to generate some of our resources. Finally, note that opinion words are not limited to adjectives and adverbs (see oe_5 in sentence 2).

Features are usually *explicitly* mentioned by some feature words, but sometimes they are not (as in oe_6 and oe_7). Then we say they are *implicit* features, which have to be deduced by context (opinion words seem to be a good indicator, as we will set out afterward). Another issue you have to deal with is the frequent use of pronominal references, that have to be solved in order to decide what feature the author is talking about (see oe_5). Following on features, the same feature can be shared by two or more opinions in the same sentence. The third and fourth opinion evidences are a good example of it. Even though you could argue that they may be an unique positive opinion, this atomic scheme fits better in our opinion words definition, and solves some problematic cases (see examples oe_8 and oe_9).

3.1.2 Lexico-syntactic extractable opinions

Observe this clause from sentence 3: "*I always have to untwist it [the cord] every time I get them out off my pocket.*". You can dictamine that we have missed an opinion. Indeed, there is a negative evaluation on a feature of the product, according to our intuitive definition of opinion. But correctly inducing a negative opinion is a really hard task: it involves a deep semantic parsing of text, and a significative world knowledge reasoning in order to capture the negative implications of this particular fact. Undoubtedly, this is a very challenging and interesting problem. But the practical nature of our approach leads us to avoid the extraction of this type of opinions, since semantic parsing and automatic reasoning are not solved problems. Because of similar practical considerations, we are only interested in those opinions having not too complex syntactic structures. Syntactic parsers are more commonly used than semantic ones, but their accuracy drops considerably when analyzing large, syntactically complex sentences.

So, we will focus on those opinions where:

- You can identify a few words from which you can decide the polarity of the opinion (*opinion words*).
- The syntactic relations between opinion words, and between these and feature words, must be simple enough to be correctly parsed by state-of-art syntactic analysers.

3.2 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 develop these resources, we start from a manual effort (though computer assisted) in order to describe a feature taxonomy and annotate opinion evidences in a corpus of reviews. Then we apply some algorithms which try to extract important information about key concepts of the annotated opinion evidences, like the opinion words that have been used, correlations between those opinion words and implicit features or which syntactic patterns are more frequent, among others. This knowledge is saved into a set of domain-specific resources to be used later on by the opinion extraction system . In this section we present a brief overview of these resources, and shortly explain the process we follow to generate them. A graphical scheme of the whole process is shown in figure 5.

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 out there, where professional or, more frequently, anonymous reviewers 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, questionable grammatical constructions , etc.

3.2.2 Feature taxonomy

The *feature taxonomy* contains the set of product features for which opinions will be extracted, a subset of F_P. Besides, each feature f_i comes with a set of feature words, a subset of FW_{f_i} (hopefully nearly the complete set). 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 in a set of subfeatures. A piece of the feature taxonomy for product class *headphones* is shown in 3. The taxonomy hierarchy is not exploited by the extraction system, but we think it will be a useful resource when aggregating opinions to produce summaries. For example, using taxonomy from figure 3, you could not only obtain independent summaries of opinions on *bass*, *mids* and *treble* features, but also a

[2] In the corpus annotation process being described later, we decided to annotate these *intensity words* to be used in the future, but we are not working with this information yet as we are only concerned about binary opinion classification

summary of opinions on *frequency response*, including the previous ones.

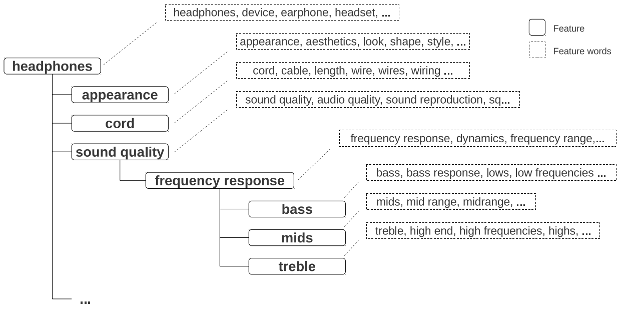
headphones, device, earphone, headset, ...
Feature
Feature words
appearance, aesthetics, look, shape, style, ...
headphones
cord, cable, length, wire, wires, wiring ...
appearance
sound quality, audio quality, sound reproduction, sq...
cord
sound quality
frequency response, dynamics, frequency range,...
frequency response
bass, bass response, lows, low frequencies ...
bass
mids, mid range, midrange, ...
mids
treble, high end, high frequencies, highs, ...
treble
...

Figure 3: Headphones feature taxonomy

This resource is built in two steps. First, a list of feature words is generated from the corpus, as shown in figure 4. Then, an expert should produce the taxonomy, grouping feature words by feature and building a hierarchy.

Feature words extraction is made in a semi-supervised way. Starting from a few opinion words seeds [3], the algorithm looks for feature words candidates appearing in some simple part-of-speech patterns near any opinion word. 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; those 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.

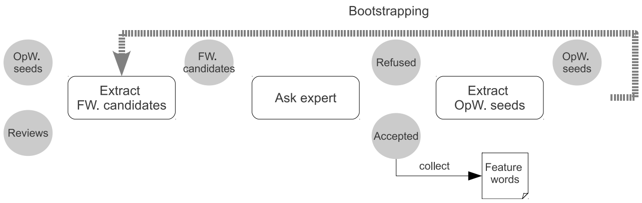

Figure 4: Interactive feature words extraction

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 to evaluate in an experimental setup. So, you should annotate as many reviews as possible. It is desirable to have a uniform distribution of evaluation ratings over the reviews chosen to annotate.

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, such as required fields missing (feature, opinion words and polarity). If an unknown feature or feature word has been annotated and it is not a mistake (annotator is asked to confirm), those elements are added to the feature taxonomy. The application also detects opinion words being employed with semantic orientation opposite to previous annotations, and informs the annotator to prevent possible mistakes. Besides error checkings, the application warns the annotator about some probably missed opinion evidences, if some words corresponding to *(a)* feature words, or *(b)* opinion words previously used in opinion evidences with implicit feature, are not annotated.

3.2.4 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. We have identified two of these types of expressions. First, *negation expressions* invert the polarity of the semantic orientation of an opinion (e.g., not, hardly, barely,. . .). Second, *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 and dominant polarity expressions are domain-independent. We start from a small manually-collected list of expressions. The validation application previously described 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 can help us to recognize opinions, and their semantic orientation to classify them. The *opinion lexicon* contains some useful information about opinion words and their semantic orientations.

For each term (individual word or phrase) annotated as opinion words, the opinion lexicon contains the following measures, which are estimated from the annotation corpus:

- Support: number of occurences of the term in the annotated corpus. The greater value the more accurate the rest of estimated measures are.
- Opinion word probability: an estimation of the probability of this term being used as opinion word.
- Feature-based opinion word probabilities: given a feature, an estimation of the probability of this term being used as opinion word in an opinion on that feature.
- Feature-based semantic orientations: given a feature, an estimation of semantic orientation of this term being used as opinion word in an opinion on that feature. Each estimation is a real number between -1.0 and 1.0.

A term being used in opinions with always positive or negative implications is assigned 1.0 or -1.0 respectively. Other values indicate some level of ambiguity. Note that the absolute value of SO is not correlated with the intensity of the positive or negative implications of a term; it is rather correlated with the probability of that term having positive or negative implications. The SO of a term for a given feature is estimated from annotated opinion evidences on that feature or any subfeature of it. Most of the times, an ambiguous SO value on a feature indicates opposite, unambiguous SO values on some subfeatures of it. For example, SO of *cheap* being used in an opinion on feature *headphones* was estimated as 0.4693, being -1.0 on most subfeatures

[3] *excellent*, *good*, *bad* and *poor*

(*appearance*, *durability*, *sound quality*,etc.) and 1.0 on a single, but more frequently observed one (*price*).

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 in opinions on *comfort* and *price* features, respectively. The *implicit feature cues* resource intends to collect this kind of information, which can be very useful in order to discover opinions on implicit features.

For each term (individual word or phrase) annotated as opinion words in an opinion evidence with implicit feature, the resource contains the following measures:

- Support.
- Feature-based implicit feature probabilities: given a feature, an estimation of the probability of this term being used as opinion word in an opinion on that implicit feature.
- Feature-based implicit feature, not explicit, conditional probabilities: given a feature, an estimation of the probability of this term being used as opinion word in an opinion on that implicit feature, knowing that it has not been used in another opinion with an explicit feature.

The last two values are used by our system in two different points of the extraction process: before and after the extraction of opinions on explicit features. If you are trying to discover opinions on implicit features as an stand-alone task, the feature-based implicit feature probabilities should be used. But once opinions on explicit features have been extracted, the conditional probabilities work better.

3.2.7 Dependency patterns

The feature taxonomy helps us to identify potencial feature words, and the opinion lexicon allows us to find and classify potencial opinion words. But it is also necessary to correctly link related feature and opinion words, in order to completely fill in a new opinion evidence. We will address this problem by making use of the dependency relations between words, as parsed by Minipar [8]. The *dependency patterns* resource contains a list of 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 dependents. 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 is formed by a list of part-of-speech tags 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 figure 6, the dependency pattern linking the feature word *seems* to the opinion word *perfect* is $N \rightarrow subj \rightarrow V \rightarrow desc \rightarrow J$ [4] , where N, V and J are the part-of-speech tags 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.

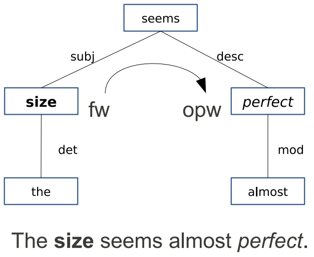

Figure 6: An example of dependency tree

We extract five dependency pattern lists from annotated corpus, formed by different types of patterns: 1. patterns linking any feature word with any opinion word; 2. patterns linking the head of feature words with the head of opinion words; 3. patterns linking the head of opinion words with any other opinion words; 4. patterns linking the head of opinion words with a negation word; and 5. patterns linking the head of opinion words with a dominant polarity word. Each type of pattern will be used by different components of our system (see section 3.3). For each dependency pattern, the resource also contains the following information:

- Support.
- Feature on which this pattern is applicable: the pattern can only be applied from any source word contained in a opinion evidence on this feature. If this field is undefined, the pattern can be applied from any source word in any opinion evidence. When extracting patterns, both feature-restricted and feature-independent patterns are learned.
- Precision: being a real value between 0.0 and 1.0, it measures the precision of this pattern linking source words to destination words which play the expected role (according to the type of the pattern).
- Recall: being a real value between 0.0 and 1.0, it measures the completeness of this pattern in linking words with the expected role from a given set of source words.

The precision and recall of each pattern is computed by applying it from each annotated source word corresponding to the appropiate role (depending on the type of the pattern) and counting correct and incorrect destination words extracted. Then, each list is sorted by descending precision and descending recall. Finally, for each list, all patterns are consecutively applied, in order to compute accumulated precision and recall values. These values are also saved together with each pattern in the resource.

3.3 System architecture

Our opinion extraction system is formed by 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.

[4] 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 $V \rightarrow desc \rightarrow J$

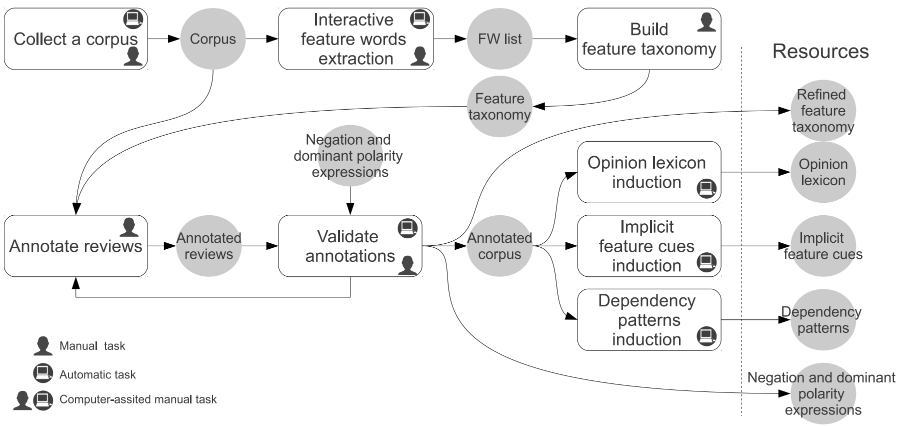

Figure 5: Resource generation process

This modular design together with the multiple implementations of each component make up an experimental setup that enable us to test different approaches.

Let us give a brief description of these components. The *explicit feature annotators* discover feature 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 *negative 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 (they actually set a real value for semantic orientation, although only the sign of that value will be taken into account in evaluation as our problem definition establishes). 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 try to solve conflicts between various 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 extraction. It takes a list of reviews as input, that are processed using some NLP external tools: a tokenizer, a part-of-speech tagger, a sentence segmentator and a dependency parser. We are using Minipar [8] for dependency parsing and Freeling [1] for the rest of processing. The opinion evidences obtained as output should be the input of an still in definition set of components, with aggregation, summarization and visualization tasks intended.

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. The latter include window-based versions of the *opinion word*, *negation expression* and *dominant polarity expression linkers*, and two *opinion classifiers*, one using *Wordnet*[3] in a similar way to [6], and one using the *PMI-IR* algorithm explained in [12]. A few parameters are available to configure each resource-based component; some of them are thresholds related with measures contained in the resources. By tuning this parameters you can get a greater value of precision, at the expense of recall, or viceversa. A more detailed explanation of each concrete component is avoided because of space considerations.

4. EXPERIMENTAL RESULTS

In this section we describe some initial experiments performed over a corpus of headphones reviews. We chose *headphones* as product class, as it has a relatively small set of features. We annotated 599 reviews, randomly chosen among all the headphones reviews available in *epinions.com*, but with uniformly distributed ratings. Some statistics about reviews 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 ([5],[11],[9],[2]) contain a more balanced set of sentences with and without opinions (about one out of two). Although we could artificially balance the corpus, we preffer using the reviews just as they are extracted, as we are interested in measuring the accuracy of our approach when being applied in a real enviroment.

Reviews	599
Words	142832
Sentences	8302
Sentences containing opinions	2554
Number of features in taxonomy	35
Opinion evidences	3887
Opinions with...	
...implicit feature	36,56%
...explicit feature	63,44%

Table 1: *Headphones* annotated corpus statistics

All the experiments below were done using 10-fold cross-validation, taking 500 and 99 reviews as training and test sets, respectively. For those experiments measured by precision and recall values, we also compute f_β [13]. We use $\beta = 1/2$, which weights precision twice as much as recall. An opinion extraction system will be useful to process a large number of reviews as input, generating some kind of

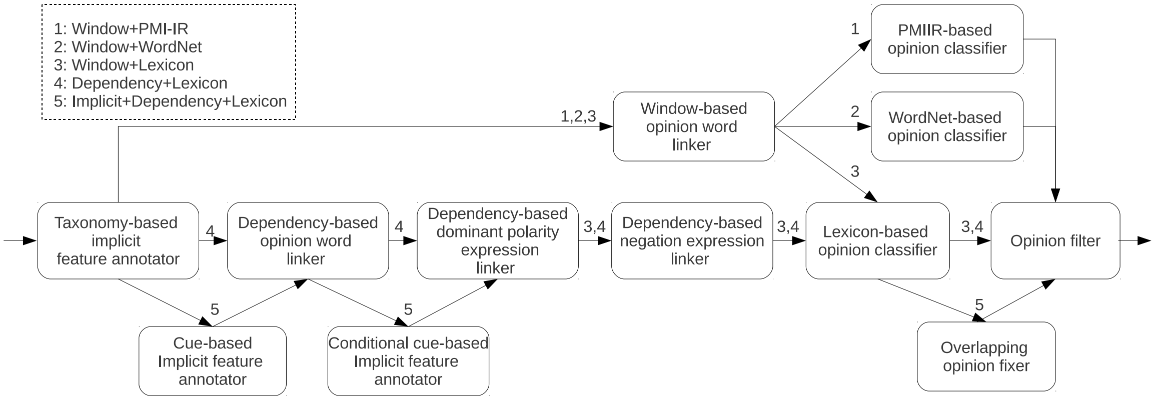
1: Window+PMI-IR
2: Window+WordNet
3: Window+Lexicon
4: Dependency+Lexicon
5: Implicit+Dependency+Lexicon
Taxonomy-based implicit feature annotator
Dependency-based opinion word linker
Dependency-based dominant polarity expression linker
Dependency-based negation expression linker
Lexicon-based opinion classifier
Opinion filter
Window-based opinion word linker
PMIIR-based opinion classifier
WordNet-based opinion classifier
Cue-based Implicit feature annotator
Conditional cue-based Implicit feature annotator
Overlapping opinion fixer
1,2,3
1
2
3
4
4
3,4
3,4
3,4
5
5
5

Figure 7: Experimental pipelines. Each of five paths represents a different experiment.

summary of the opinions contained in those reviews. In this context, we think it is better to extract some reliable opinions, although missing some others, rather than to obtain a big set of opinions, containing almost all the existent opinions but including a high number of non-existent ones.

4.1 Evaluation of independent components

We have conducted some experiments in order to measure the accuracy of the most relevant concrete components. Some results are shown in table 2.

Using the taxonomy-based explicit feature annotator, nearly every feature word in the annotated corpus is found; but many feature words not corresponding with annotated opinions are also tagged. So a majority of the tentative opinion evidences output by this component are incorrect ones. This is related with the low proportion of sentences in the corpus containing opinions, and it makes the opinion recognition task a harder problem.

Window-based and dependency-based opinion words linkers were tested starting from feature words annotated in the corpus. The dependency-based opinion word linker considerably outperforms the window-based one in terms of precision: more than 3 out of 4 suggested opinion words coincide with those ones in the annotated corpus. Recall is noticeably lower, though; but as explained before, we are more interested in exactness rather than completeness. Nevertheless, combining both approaches may be an interesting issue.

Three different opinion classifiers were tested over opinions in the annotated corpus. It seems that having an opinion lexicon is a clear advantage in order to decide the correct semantic orientation of a given opinion evidence.

	Precision	Recall	$F_{1/2}$
Feature word annotation	22,11	99,14	26,18
Opinion words linking:			
Window-based	31,2	80,87	35,57
Dependency-based	77,78	61	73,72
Opinion classification:			
PMIIR-based	73,63	-	-
WordNet-based	65,1	-	-
Lexicon-based	86,36	-	-

Table 2: Feature word annotation, opinion word linking and opinion classification independent evaluations.

4.2 Opinion recognition and classification

Five experimental pipelines were tested using the annotated corpus (see figure 7). All of them contain an explicit feature annotator that makes use of the feature taxonomy in order to annotate feature words occurrences, and an opinion filter to remove opinions with a semantic orientation equal to zero. Only the fifth pipeline deals with opinions on implicit features. The results obtained by these pipelines for opinion recognition and classification are shown in table 3.

4.2.1 Explicit opinion recognition and classification

In order to measure the gain in accuracy contributed by resources, the first two pipelines are formed by components which do not make use of any of them (except for the feature taxonomy). The third pipeline includes a lexicon-based opinion classifier, and the fourth one replaces the window-based opinion word linker by a set of three components based on dependency patterns (opinion word, negation expression and dominant polarity expression linkers). These four pipelines are not extracting any implicit feature opinion, which means that more than a third part of total opinions remain unreachable (see table 1). The *opinion recognition* column reports results in correctly discovering opinion evidences on explicit features, and the *opinion classification* one shows how many of those correctly discovered opinion evidences were correctly classified. Finally, the last column contains results for the complete task. Note that only those opinions on explicit features from the annotated corpus were used to compute recall.

The best results are obtained by the fourth pipeline, which makes use of the opinion lexicon and the dependency patterns. It seems that the contribution of the opinion lexicon to that results is greater than the contribution of the dependency patterns, as you can deduce from the results obtained by the previous pipelines. We think it can be related to the accuracy of the dependency parser, as we observed some feature/opinion word pairs for which a path in the dependency tree could not be found. These leads to a lower recall than expected when using the dependency-based opinion words linker. Nevertheless, the high precision obtained makes up for it.

4.2.2 Implicit opinion recognition and classification

The fifth pipeline adds two implicit feature annotators, one before the explicit feature annotator and one after it.

	Opinion Recognition			Opinion Classification	Opinion Recognition + Classification		
Pipeline	Precision	Recall	$F_{1/2}$	Precision	Precision	Recall	$F_{1/2}$
Window+PMI-IR	0.257	0.682	0.293	0.792	0.203	0.542	0.232
Window+WordNet	0.298	0.813	0.341	0.843	0.251	0.685	0.287
Window+Lexicon	0.501	0.773	0.539	0.953	0.477	0.729	0.513
Dependency+Lexicon	0.662	0.599	**0.648**	**0.98**	0.649	0.583	**0.634**
Implicit+Dependency+Lexicon	0.689	0.593	**0.667**	0.978	0.674	0.577	**0.652**

Table 3: Opinion recognition and classification results. Recall values for the first four pipelines were computed using only those opinions on explicit features from the annotated corpus.

The first one uses feature-based implicit feature probabilities estimated in the implicit feature cues resource, and the second one uses the conditional probabilities instead (see section 3.2.6). An overlapping opinion fixer is also added to this pipeline in order to deal with possible implicit/explicit overlapping opinions [5]. In these experiments, all the opinions from the annotated corpus, both the explicit and the implicit ones, were used to compute recall.

This pipeline achieves an f-score equal to 0.652 for the opinion recognition and classification task. We think this is a good result , taking into account the low precision of the initial explicit feature annotation (≈ 22%, which means that one of each five feature words occurrences are not corresponding to real opinions). Besides, there are some configurable thresholds available in every component, that allow us to tune the system in order to obtain a better precision at the expense of a lower recall; in this way, we expect to be able to build reliable and useful opinion-based summaries from product reviews.

5. CONCLUSIONS

In this work, we introduced a domain-specific, resource-based approach to the problem of opinion extraction from product reviews. We proposed a redefinition of the problem and a methodology to build opinion extraction systems for a given product class in a semisupervised way. We have described the resources which captures the domain knowledge and given a brief description of the system architecture. Finally, we reported results of some initial experiments performed over a set of headphones reviews. These results demonstrate that domain-specific knowledge is a valuable resource in order to build precise opinion extraction systems.

We are currently ending the annotation of a corpus composed by a thousand reviews of hotels. We are interested in applying the techniques explained here in a new, more complex domain, and observing if they fit well. We also intend to figure out if a part of the effort employed in the construction of one of these systems for an specific domain can be exploited in the construction of a system for a new domain (e.g., adapting some of the resources from one domain to another).

6. REFERENCES

[1] J. Atserias, B. Casas, E. Comelles, M. González, L. Padró, and M. Padró. Freeling 1.3: Syntactic and semantic services in an open-source nlp library. In *Proceedings of the fifth international conference on Language Resources and Evaluation (LREC 2006)*, Genoa, Italy, May 2006. ELRA. http://www.lsi.upc.edu/ nlp/freeling.

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

[3] C. Fellbaum, editor. *WordNet: An Electronic Lexical Database.* MIT Press, 1998.

[4] M. Hu and B. Liu. Mining and summarizing customer reviews. In *KDD '04: Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 168–177, New York, NY, USA, 2004. ACM.

[5] M. Hu and B. Liu. Mining opinion features in customer reviews. In *Proceedings of AAAI*, pages 755–760, 2004.

[6] J. Kamps, M. Marx, R. J. Mokken, and M. De Rijke. Using wordnet to measure semantic orientation of adjectives. In *National Institute for*, volume 26, pages 1115–1118, 2004.

[7] . Lehrer, Adrienne. *Semantic fields and lexical structure / A. Lehrer.* North-Holland ; American Elsevier, Amsterdam : New York :, 1974.

[8] D. Lin. Dependency-based evaluation of minipar. In *Proc. Workshop on the Evaluation of Parsing Systems*, Granada, 1998.

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

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

[11] A.-M. Popescu and O. Etzioni. 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)*, 2005.

[12] P. Turney. Thumbs up or thumbs down? Semantic orientation applied to unsupervised classification of reviews. In *Proceedings of the Association for Computational Linguistics (ACL)*, 2002.

[13] C. van Rijsbergen and P. D. Information retrieval, 1979.

[14] L. Zhuang, F. Jing, X. Zhu, and L. Zhang. Movie review mining and summarization. In *Proceedings of the ACM SIGIR Conference on Information and Knowledge Management (CIKM)*, 2006.

[5] E.g., in sentence *"The headphones are too small"*, an explicit opinion on feature *headphones* and an implicit opinion on feature *size* might be found by the system; only the latter should be extracted.