A Feature Extraction Process for Sentiment Analysis of Opinions on Services

Henrique Siqueira and Flavia Barros

Centro de Informática (CIn) - Universidade Federal de Pernambuco (UFPE)
Recife - PE - Brazil
hbas@cin.ufpe.br, fab@cin.ufpe.br

Abstract. The growing number of blogs, forums and social networks in the Web drastically increased the amount of texts conveying not just facts but opinions. A large number of opinions are customers' reviews about products and services, as e-commerce became more popular. This trend motivated several research works and market applications aiming at the automatic analysis of the available opinions. Clearly, this information is crucial for managers, who should improve the quality of the offered services based on customers' opinions. Concerning services provided by enterprises such as stores or hotels, it is particularly difficult to identify the features being commented on (e.g., quality of service, delivery cost, etc). This work presents a domain-free process for feature extraction which was used in the construction of WhatMatter. The prototype system was used for feature extraction from opinions on services (a domain not yet covered by the literature). The obtained results were very satisfactory.

1 Introduction

The advent of the Web drastically changed the way people communicate, and also express their opinions and preferences. As e-commerce is becoming more and more popular, the number of customers' reviews about products grows rapidly in specialized sites, blogs, forums and social networks. Clearly, it is a hard task for the users to identify relevant opinions on a particular product (or a product's feature -e.g., a notebook's keyboard), or on a particular service in order to make informed decisions on whether to purchase the product or to use the service. This information is also crucial for enterprises managers, who should consider the customers' opinions when trying to improve the quality of the offered services.

Trying to fill in this gap, we witness a growing number of research works and market applications focusing on the automatic processing of opinions [1]. This research area, known as Sentiment Analysis (or Opinion Mining), focuses on the automatic summarization and classification of opinions polarity [2].

Sentiment Analysis (SA) usually counts on four tasks: opinion identification (identifying the text which contains an opinion), feature extraction (identifying the aspects being commented on - e.g., the notebook's price), sentiment classification (i.e., whether the opinion polarity is positive, negative or neutral), and visualization and summarization of results. See Section 2 for more details.

The majority of the works available in the literature focuses on the automatic analysis of opinions on products [3], leaving aside opinions about enterprises and their services (e.g., hotels, travel agencies, shops, etc). Different from opinions on products (which have a limited number of well known features), opinions on services may be much more subjective, and the features being commented on are not always clearly stated in the text. This way, it is harder to automatically identify which aspect or event pleased or displeased the customer.

In SA, feature extraction is one of the most complex tasks, since it requires the use of Natural Language Processing techniques in order to automatically identify the features in the opinions under analysis [4]. This task is even harder when dealing with opinions about stores, since the features in this domain are not clearly pre-determined. Besides the most obvious aspect -the price of available products-, other important features are: the site usability, delivery cost and time, the store's reliability, customer care service, and even the product packaging.

The work presented here focuses on the feature extraction task, proposing a domain-free process which was successfully applied to automatic feature extraction from opinions about services (a domain not yet covered by the available literature). This task is harder for this domain, since the use of ontology is not possible (as used in works with products [5]). Yet, there are no tagged corpora (i.e., benchmarks of opinions on services) for the use of machine learning.

The proposed process was used in the construction of *WhatMatter*, a prototype system for feature extraction from opinions on services. The system executes four main steps: Frequent nouns identification, Relevant nouns identification, Feature indicators mapping, and Unrelated nouns removal. Statistical methods were used here, avoiding the need for domain ontologies or training corpora.

WhatMatter, was used to extract features from a manually tagged corpus in Portuguese containing 200 opinions on shops services. The obtained results were very satisfactory, considering the difficulty of the task: precision of 77.24%, recall of 90.94% and F-measure of 83.54%.

The following section briefly presents the basic concepts of Sentiment Analysis, highlighting current works on Feature Extraction. Section 3 shows details of the proposed technique, and section 4 presents experiments results. Finally, section 5 brings conclusions and future work.

2 Sentiment Analysis and Feature Extraction

The research area of Sentiment Analysis, also known as Opinion Mining, covers the computational treatment of sentiments and emotions expressed in a text [2]. As said, SA counts on four tasks: opinions identification, features extraction, sentiment classification, or visualization and summarization of results.

The first task aims to determine whether the sentence under analysis contains or not an opinion. For that, it is necessary to distinguish opinions (e.g., "I really liked the store's website") from simple statements (e.g., "This was the first time that I bought a notebook"). Only sentences containing opinions are collected for further processing.

As it can be observed, opinions and sentiments always refer to an entity (an object, a service, a person), explicitly or not. The **Feature Extraction task** aims to identify the entities being referred to (e.g., "the shop's website" in the example above). The **Classification task** aims to determine the opinions' polarity (positive, negative or neutral) regarding the features being commented on. The above example brings a positive opinion about the store's website.

The final task aims to present a summary of the performed analysis to the user, in order to support the decision making process. This summary usually lists the amount of positive and negative evaluations of each feature. Some systems also present graphs in order to clarify the summary visualization.

It is important to highlight that these tasks are not necessarily performed in a pipeline fashion. In practice, some works perform two tasks in one go. For instance, some works classify a given piece of text as positive, negative or without opinion, performing the identification and classification tasks simultaneously [6]. Yet, the majority of the works in SA usually focuses on only one or two of these four tasks.

2.1 Feature Extraction

As said, Feature Extraction (FE) is perhaps the most difficult task in SA. In the related literature, we did not find many works focusing on FE. In fact, the majority of the available works focus on sentiment classification.

In general, the works on FE do not follow any "consensual" approach, frequently offering ad-hoc solutions. Nevertheless, it is possible to identify two mainstream directions. On one hand, some works deploy machine learning techniques to identify terms appearing in positive or negative opinions [7]. In this case, the extraction of features depends upon the availability of a tagged corpus. Yet, other works are based on ontologies, being thus domain-dependent [5].

These approaches do not seem to be adequate for FE in the services domain, since this domain cannot be organized into an ontology (because its features are not clearly pre-defined), and there are no available corpora for machine learning.

The remaining of this section briefly reviews two relevant works on FE which obtained good results in this task.

A work worth to mention is the Red Opal system [8], which aims to identify the best evaluated products with respect to a set of automatically extracted features. In this work, a word is considered as a feature when the probability of finding it in texts containing opinions is very different from the probability of finding it in any other kind of text. Clearly, the drawback of this work is the need for a huge corpus of texts (opinions and non-opinions) in the target language in order to infer the probability of a word being or not a feature (Red Opal used more than 100 millions of words in English).

Finally, Hu e Liu [1, 4] propose a system for features extraction and classification based on free texts. The focus here is on electronic products, which have a reduced number of possible features. This way, the system was able to reach high precision and recall rates simply by using association mining and identifying infrequent features that are irrelevant to the given product.

3 The WhatMatter System

We present here a domain-free process for feature extraction based on natural language processing and statistical methods, dispensing with the use of a domain ontology or of training corpora. The proposed technique was used in the implementation of *WhatMatter*, a system for feature extraction from opinions on services (a domain not yet covered by the available literature).

Although our focus is on FE, the system also accounts for the classification and visualization tasks (this was important to help in the analysis of the experiments). However, the deployed techniques in these two tasks were very simple and are not worth to mention here. The initial SA phase (opinion identification) was done by hand.

The FE process receives as input a text containing an opinion, and returns the extracted features. The extraction process includes four main steps (sections 3.1 to 3.4): Frequent nouns identification, Relevant nouns identification, Feature indicators mapping, and Unrelated nouns removal.

Most decisions on the FE process were guided by a manual analysis of a collected corpus with 2 thousand opinions, used to help in the knowledge acquisition process. A distinct corpus, with 200 opinions, was created to conduct the experiments (section 4).

Before starting the FE process itself, a pre-processing phase is needed, in order to prepare the text for further processing. First of all, the input text is split into sentences, and each sentence is then analyzed by a Part-of-Speech (POS) Tagger [9]. These POS tags are used by the steps mentioned above.

3.1 Step 1: Frequent nouns identification

A detailed analysis of a large number of opinions on services revealed that the most frequent nouns in the text usually correspond to relevant features. This finding was also confirmed by the available literature in FE (e.g., Nakagawa and Mori [10] have reached a similar conclusion for opinions expressed on products).

This way, the system's first module receives the tagged sentences as input, and totalizes the frequency of occurrence of each noun. Here, a noun is considered frequent when it is within the 3% most frequent nouns (this threshold was empirically determined based on the knowledge acquisition corpus - see section 4). The selected nouns are collected into a list of candidate features for further processing.

3.2 Step 2: Relevant nouns identification

A further analysis of the corpus showed that the list of most frequent nouns does not always cover all important features being commented on. There are still some features mentioned in a smaller number of opinions that are also important to be extracted by the system.

Consider the following sentences: "This store has excellent prices" and "I've found excellent products in here". Both sentences use the same opinion word (the

adjective "excellent"), however this adjective is modifying two different nouns (prices and products), both referring to relevant domain features.

The corpus analysis still revealed that adjectives near frequent nouns are good indicators of relevance when found near not so frequent nouns. This way, our next step consists of collecting the adjectives adjacent to the candidate features identified in the Step 1, and use them to identify new nouns to be included in the candidate features list. To improve the accuracy of this step, we disregard prepositions and irrelevant words (known as stopwords) when determining adjacency between a noun and adjectives. This Step is divided into two, as follows.

Step 2.1 - Adjectives identification Here, for each sentence in the corpus and each noun in the candidate features list, the system identifies whether there is an adjective immediately before or after the noun, or separated only by prepositions or stopwords. The identified adjectives are stored in the relevant adjectives list.

Step 2.2 - New candidate features (nouns) identification In this module, the system processes again each sentence in the corpus looking for nouns adjacent to the adjectives collected in the previous module. After this, the candidate features list created in Step 1 is replaced by a new list with all nouns identified by this module (which includes the nouns in the initial candidate features list plus the ones selected by this module).

3.3 Step 3: Feature indicators mapping

When analyzing users' opinions, we note that some relevant features may not be explicitly mentioned in the text. For example, in the sentence "The hotel was expensive", the word "price" was omitted, even thought the user is clearly referring to this feature. This practice makes more difficult the automatic identification of features. In this context, the adjectives and adverbs used to implicitly refer to a feature are also very important, being called feature indicators [11].

This Step performs a mapping from feature indicators found in the text to the actual features being referred to. However, this mapping requires extra care, since several adjectives can be quite versatile, and their meaning is usually domain/context dependent. For example, in the sentence "The traffic is heavy", "heavy" does not describe the weight of the traffic.

The usual way to perform this task is via the use of a manually compiled list of such mappings. It is not clear yet whether there are more effective approaches, as little research has been done in the search for alternatives[2]. Our system uses a manually compiled list of 20 feature indicators for the chosen domain to identify features that are implicitly mentioned (e.g., "price") through the use of any of its indicators (e.g., "cheap" or "expensive").

3.4 Step 4: Unrelated nouns removal

This Step is a refinement of Step 2, which selects infrequent nouns as candidate features based on opinion words. Note that among relevant nouns, this practice

may also select nouns/terms that are irrelevant to the domain under analysis. This is due to the fact that opinion writers may use common adjectives to qualify several different entities (nouns), including both relevant and irrelevant features.

In this Step, we filter irrelevant nouns using the PMI-IR measure [12], a variant of the Pointwise Mutual Information (PMI) measure. PMI is used in Information Theory to calculate the association between two or more words. It compares the probability of finding each word in isolation with the probability of finding them together, helping to determine their correlation [13].

Turney [12] used this concept in sentiment classification, creating the PMI-IR measure, which estimates PMI by querying a Web search engine with the words under analysis, and computing the number of returned hits (matching documents). This measure is given by the formula bellow:

$$\mathrm{PMI}_{IR}(t1, t2) = log_2 \left(\frac{Hits(t1 \wedge t2)}{Hits(t1) * Hits(t2)} \right) \quad (1)$$

where *Hits(t1)* is the number of pages containing *t1*, *Hits(t2)* is the number of pages containing *t2* and *Hits(t1* $\wedge$ *t2)* is the number of pages with both terms.

In our system, we built a module that queries Google in 3 steps: the first step computes the number of hits for each candidate feature in isolation; the second step computes the number of hits for a word representing the opinion's domain (for example "stores"); and the third step computes the number of hits with both the candidate feature and the word representing the domain.

Based on these results, the system computes the PMI-IR of the each candidate feature in the domain, and eliminates the nouns whose measure is bellow an empirically determined threshold.

After the execution of Step 4, all the remaining features and indicators are saved in a text file and the feature extraction process is finished.

4 Experiments Results

In order to validate the proposed process, we built a prototype system called *WhatMatter*. As said, the system also accounts for the classification and visualization tasks, in order to help in the analysis of the conducted experiments.

The corpus used in the system's construction and validation was collected by hand, counting on 200 opinions on services (see section 4.1). Section 4.2 brings details on the performed experiments.

The system was implemented in Java 1.6, counting on a modular architecture, to safeguard extensibility. The current prototype uses the TreeTagger library [9] configured with the Portuguese language pack to perform the POS tagging of the input sentences (Pre-processing phase).

Finally, we highlight that, although POS Taggers are language-dependent, this work is domain and language free. Due to the system's modular architecture, the POS Tagger can be changed as desired.

4.1 The Corpora

In order to help in the system's implementation and validation, it was necessary to build two corpora with a representative number of opinions on services (totalizing 2,200 opinions). The first corpus, with 2 thousand opinions, was used to acquire knowledge about the domain, in order to build our solution - Section 3. A distinct corpus, with 200 opinions, was used as a validation corpus for the experiments.

Both corpora were downloaded from E-bit[1], a Brazilian site dedicated to collect reviews regarding the quality of services provided by online stores. Initially, the HTML tags were automatically removed from the downloaded Web pages, and the opinions were stored in a text file. For each opinion, we also stored the number of stars (from 0 to 5) associated to the user's comment. These stars were used by our system in the classification phase as being the opinion's polarity. The opinion usually conveys (implicitly or explicitly) the features that were considered in his/her evaluation of the service or store being commented on. In fact, we used the stars to compute the polarity of the features in the opinion being processed.

4.2 Evaluation of the Feature Extraction Process

The FE process was evaluated based on two traditional measures used in Sentiment Analysis and Text Classification: precision and recall[1]. We also computed the F-measure, a combined metric that takes both precision and recall into consideration, as follows [14]:

$$\text{F-measure} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{2}$$

In order to calculate these metrics, it was necessary to manually extract the relevant features appearing in the opinions on the validation corpus. This task was performed as follows: for each sentence containing opinions, all the implicit and explicit features evaluated by the user were identified and stored on a separate file. This list was then compared to the list of automatically extracted features, and the precision, recall and F-measure rates were calculated.

We did not consider the computational cost of our solution as relevant because, in most SA applications, the feature extraction process is an off-line activity that is not frequently repeated [15].

As the prototype counts on a modular architecture, it was possible to evaluate the contribution of each Step in the FE process in isolation (see Table 1). The evaluation process started by executing Step 1 for each opinion in the corpus. The evaluation measures were calculated by comparing the automatically extracted features to the expected result (the list of manually extracted features). After all opinions in the corpus had been processed, we computed the average result over the corpus of 200 opinions. Following, the other 3 Steps were also

[1] http://www.ebit.com.br/

individually evaluated for every opinion in the corpus. Table 1 summarizes the average precision, recall and f-measure for each Step in the FE process.

Table 1. Average precision, recall and f-measure of the FE process.

Steps	Precision	Coverage	F-Measure
Step 1	28.09%	24.91%	26.40%
Steps 1+2	29.12%	53.96%	37.83%
Steps 1+2+3	50.62%	92.08%	65.33%
Steps 1+2+3+4	77.24%	90.94%	83.54%

The first module (Step 1) extracts only the 3% most frequent features in the text. Clearly, this module did not reach acceptable levels of Precision and Recall, obtaining an F-measure of 26.4%. Following, we executed Step 2, which uses the opinion words (adjectives) to extract more relevant nouns from the text. This Step significantly increased the recall measure, with a slight increase in precision, obtaining an F-measure of 37.83%.

The next experiment included Step 3, the feature indicator mapping module, in order to identify features that were implicitly mentioned in the opinion. This module uses the manually compiled list of indicators, which counts on 20 adjectives and adverbs that are related to 6 relevant features in the service domain. This addition significantly increased the obtained measures, showing an F-measure of 65.33%.

Finally, we noticed that even though the prototype was able to extract most of the relevant features, a good deal of irrelevant nouns was still appearing in the relevant features list. To minor this problem, we used the PMI-IR metric, aiming to remove nouns that were not related to the chosen domain. The word "store" was used to query the Web search engine, representing the chosen domain. This Step significantly increased our precision, with a slight decrease in recall. However, the system obtained its best performance, with F-measure of 83.54%, precision rate above three quarters and recall above 90%. These results were considered as very good, taking into account the difficulties to work in the services domain.

Figure 1 presents the WhatMatter interface showing the extracted features for the collected corpus. As said, the system also implements the classification and visualization tasks, to help in the analysis of the experiments on the FE task. The disk at the upper left corner shows the occurrence frequency of each feature in the corpus (each slice corresponds to a feature). The disk at the upper right corner shows the total amount of positive and negative opinions in the corpus. Below, the interface shows one bar graph for each feature extracted. Each column represents the amount of opinions receiving a particular evaluation (from 0 to 5 stars), which represents the opinions degree (sentiment evaluation).

Fig. 1. WhatMatter System - Extracted features with sentiment evaluation

5 Conclusions and Future Work

Sentiment Analysis is a field that has received constant attention in recent years. The present work has as its main contributions: (1) the proposal of a domain-free process for feature extraction based on statistics and natural language processing; (2) the implementation of the WhatMatter system, used for FE from opinions on services (a domain not yet covered by the literature, and which is more complex than the usually investigated products domain).

In experiments with 200 opinions on Brazilian online stores, our system achieved very satisfactory performance rates (precision of 77.24% and recall of 90.94%) in the extraction of implicit and explicit features.

In future work we aim to: (1) investigate/propose a domain-free technique for the automatic identification of feature indicators (adjectives, verbs and adverbs); (2) apply the techniques described by Stoyanov and Cardie [16] for co-reference resolution; and (3) use the same process described here for FE in different corpora, covering, for instance, other kinds of services (hotels, physician's offices) and other languages (e.g., English and Spanish).

References

1. Hu, M., Liu, B.: Mining and summarizing customer reviews. In: KDD '04: Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery

and data mining, New York, NY, USA, ACM (2004) 168–177
2. Liu, B.: Handbook of natural language processing. http://www.cs.uic.edu/ liub/FBS/NLP-handbook-sentiment-analysis.pdf (2009) Rascunho da segunda edio. Acesso em 27 de Agosto de 2009.
3. Dave, K., Lawrence, S., Pennock, D.M.: Mining the peanut gallery: opinion extraction and semantic classification of product reviews. In: WWW '03: Proceedings of the 12th international conference on World Wide Web, New York, NY, USA, ACM (2003) 519–528
4. Hu, M., Liu, B.: Mining opinion features in customer reviews. In: AAAI'04: Proceedings of the 19th national conference on Artifical intelligence, AAAI Press (2004) 755–760
5. Wang, B.B., Mckay, R.I.B., Abbass, H.A., Barlow, M.: A comparative study for domain ontology guided feature extraction. In: ACSC '03: Proceedings of the 26th Australasian computer science conference, Darlinghurst, Australia, Australia, Australian Computer Society, Inc. (2003) 69–78
6. Chesley, P.: Using verbs and adjectives to automatically classify blog sentiment. In: In Proceedings of AAAI-CAAW-06, the Spring Symposia on Computational Approaches. (2006) 27–29
7. Jin, W., Ho, H.H.: A novel lexicalized hmm-based learning framework for web opinion mining. In: Proceedings of the 26th Annual International Conference on Machine Learning, New York, NY, USA, ACM (2009) 465–472
8. Scaffidi, C., Bierhoff, K., Chang, E., Felker, M., Ng, H., Jin, C.: Red opal: product-feature scoring from reviews. In: EC '07: Proceedings of the 8th ACM conference on Electronic commerce, New York, NY, USA, ACM (2007) 182–191
9. Schmid, H.: TreeTagger - a language independent part-of-speech tagger, Available at: http://www.ims.uni-stuttgart.de/projekte/corplex/TreeTagger/. Access date: 30 June 2007 (2007)
10. Nakagawa, H., Mori, T.: A simple but powerful automatic term extraction method. In: COLING-02 on COMPUTERM 2002, Morristown, NJ, USA, Association for Computational Linguistics (2002) 1–7
11. Liu, B., Hu, M., Cheng, J.: Opinion observer: analyzing and comparing opinions on the web. In: WWW '05: Proceedings of the 14th international conference on World Wide Web, New York, NY, USA, ACM (2005) 342–351
12. Turney, P.D.: Thumbs up or thumbs down?: semantic orientation applied to unsupervised classification of reviews. In: ACL '02: Proceedings of the 40th Annual Meeting on Association for Computational Linguistics, Morristown, NJ, USA, Association for Computational Linguistics (2002) 417–424
13. Cover, T.M., Thomas, J.A.: Elements of information theory. Wiley-Interscience, New York, NY, USA (1991)
14. Jansche, M.: Maximum expected f-measure training of logistic regression models. In: HLT '05: Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing, Morristown, NJ, USA, Association for Computational Linguistics (2005) 692–699
15. Jain, A., Zongker, D.: Feature selection - evaluation, application, and small sample performance. IEEE Transactions on Pattern Analysis and Machine Intelligence **19**(2) (1997) 153–158
16. Stoyanov, V., Cardie, C.: Partially supervised coreference resolution for opinion summarization through structured rule learning. In: EMNLP '06: Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing, Morristown, NJ, USA, Association for Computational Linguistics (2006) 336–344