UFRGS: Identifying Categories and Targets in Customer Reviews

Anderson Kauer
Institute of Informatics – UFRGS
Porto Alegre – RS – Brazil
aukauer@inf.ufrgs.br

Viviane P. Moreira
Institute of Informatics – UFRGS
Porto Alegre – RS – Brazil
viviane@inf.ufrgs.br

Abstract

This paper reports on our participation in SemEval-2015 Task 12, which was devoted to Aspect-Based Sentiment Analysis. Participants were required to identify the category (entity and attribute), the opinion target, and the polarity of customer reviews. The system we built relies on classification algorithms to identify aspect categories and on a set of rules to identify the opinion target. We propose a two-phase classification approach for category identification and use a simple method for polarity detection. Our results outperform the baseline in many cases, which means our system could be used as an alternative for aspect classification.

1 Introduction

Aspect Based Sentiment Analysis aims at discovering the opinions or sentiments expressed by a user on the different aspects of a given entity (Hu and Liu, 2004; Liu, 2012). Recently, a number of methods and techniques have been developed to tackle this task and some of them rely on syntactic dependencies to locate the opinion target (Kim and Hovy, 2004; Qiu et al., 2011; Liu et al., 2013). A syntactic parser takes a natural language sentence as input and outputs the relationships between the words in the sentence. Figure 1 shows the dependency tree for the sentence "The phone has a good screen." and the grammatical relations of each token (det, subj, mod, obj). We explore using grammatical relations to help identify the opinion targets.

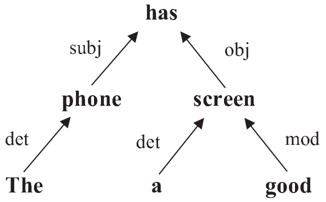

Figure 1: Example of a dependency tree (Liu et al., 2013).

In this paper, we describe a system which took part on SemEval-2015, and the way it was applied to category and polarity classification. Our system participated in all subtasks from Task 12 (Aspect Based Sentiment Analysis). For more details on this task, please refer to Pontiki et al. (2015). Our system combines classification algorithms, coreference resolution tools, and a syntactic parser. One of our goals was to minimize the use of external resources.

The remainder of this paper is organized as follows: Our system is described in Section 2. Section 3 reports on the evaluation results. Finally, section 4 concludes the paper.

2 Description of the System

In this section, we describe the different components of the system.

2.1 Pre-processing

A distinctive characteristic of Web content is the high prevalence of noise. This directly impacts the quality of the results generated by a syntactic parser. In our system, we used the StanfordNLP Core toolkit (Manning et al., 2014).

The training sentences provided by the organizers were sometimes composed by more than one sentence. Thus, before submitting them to the parser,

Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015), pages 725–729,
Denver, Colorado, June 4-5, 2015. ©2015 Association for Computational Linguistics

a cleaning step based on regular expressions was performed. In this step, we replaced all punctuation marks by commas and removed non-alphabetic characters.

Then, the standard pre-processing tools available from the StanfordNLP Core were applied (tokenization, sentence splitting, part-of-speech tagging, morphological analysis, syntactic parsing, coreference resolution, and sentiment analysis).

2.2 Aspect Category Identification

We treated the problem of identifying aspect categories as a classification task. Thus, we made use of the classifiers available from Weka (Hall et al., 2009) to build models based on the training data. In Task 12, categories are formed by a pair Entity#Attribute. The organizers have provided a list of possible entities and, for each entity, a list of attributes.

For each entity, we built a binary classifier where each instance contains the lemmas on the sentence and coreference lemmas to the previous sentences. The class indicates whether the instance belongs to the entity (*i.e.*, *positive* means that the instance belongs to the entity and *negative* means it does not belong to the entity). For each entity, the features were selected using the *InfoGainAttributeEval* with *Ranker* as a search method (available from Weka). The threshold set up to Ranker was 0, which means that the words selected by the method must contribute to identify the class.

We used two approaches to classify the sentences. In the first approach, *one-phase classification*, for each entity dataset we trained six classifiers using *all* the sentences. These six classifiers (namely IBk, ThresholdSelector, BayesianLogisticRegression, Logistic, MultiClassClassifier, and SMO) were the top performers on our experiments on the training data. We will refer to those as *Category classifiers*, as they will be used to actually determine the class. Since the classifiers for each category are independent, it is possible that a sentence is predicted as belonging to more than one category.

Classifiers were also built for each attribute belonging to that entity using *only* the sentences containing the entity. We call these *Attribute classifiers*, as they will be used to generate features for the *Category classifiers*.

In the two-phase approach (Figure 2), first we train n *Attribute classifiers* using all sentences but the current. In the experiments reported in Section 3, we used twenty *Attribute classifiers* (n=20). Then, the outputs from each of the n *Attribute classifiers* were used as features for the *Category classifiers* (second phase). This phase requires significant processing time since a new dataset is created for each instance and the models have to be updated. This method assumes that the features in each instance contain "what the others tell about it" using different prediction models.

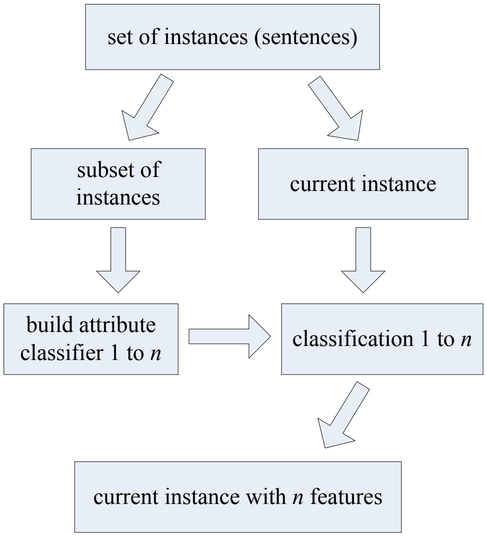
set of instances (sentences)
subset of instances
current instance
build attribute classifier 1 to *n*
classification 1 to *n*
current instance with *n* features

Figure 2: Two-phase classification pipeline.

To classify a new unseen instance, first it needs to be processed so that its lemmas and coreferences are identified. Then, word frequencies are selected and the n *Attribute classifiers* generate the values of the features for the second phase.

The final predicted class is the top scoring (*i.e.,* with the highest sum of scores) obtained from the results of the six *Category classifiers*. Although this has not happened in our experiments, a tie between the scores of the positive and negative classes is possible. In such a case, the sentence will be assigned to the positive class (*i.e.*, as belonging to the category).

2.3 Opinion Target Identification

The opinion target is detected after the category has been identified. For each pair Entity#Attribute discovered in the sentence, the candidate words are selected in order of information gain for that category. The words from attribute classification are concatenated with the words for entity classification. The assumption is that the words from attribute classification are more significant than the words from entity classification (which are more generic).

We select the word pairs which are directly associated (on the dependency tree) by a grammatical relation such as *adjectival modifier, noun compound modifier*, and *nominal subject*. We consider the opinion targets to be nouns/noun phrases as this has been widely adopted in the related literature (Hu and Liu, 2004; Qiu et al., 2011; Liu et al., 2013). Thus, the potential POS tags for targets are NN (singular nouns) and NNS (plural nouns). In order to identify incorrect targets, we rely on a list of 5k words assembled by Qian (2013). This exceptions list contains words with little or no meaning and that normally are not an aspect. The main target is the first candidate noun which is not in such a list.

If no nouns are found among the candidates, we find the nouns in the same sentence that are indirectly related to the candidate words (*i.e.* by transitivity), then we select the first noun. When still no nouns are found, then the opinion is set to *NULL* (it does not exist in the sentence). Target expressions are obtained using *noun compound modifier* (nn) associations.

A current limitation is that we do not identify multiple target expressions for the same category. We assume that for each category found, there is only one target in the sentence. However, since a sentence may be assigned to several categories, in these cases, more than one target may be identified and returned.

2.4 Sentiment Polarity Attribution

For this subtask, we used a simple approach that assigns the polarity of the target as the general polarity of the sentence. Stanford NLP Core provides sentiment analysis based on a compositional model over trees using deep learning (Socher et al., 2013). The nodes of a binarized tree of each sentence are assigned a sentiment score.

We opted for this approach to minimize the external resources in the our system, such as sentiment lexicons or reviews collected from other sources. The underlying model for Stanford NLP Core Sentiment Analysis was built on a corpus consisting of 11,855 sentences extracted from movie reviews. We have made no attempt to change the model to adapt to our reviews and used it as is to determine the polarity of the sentences. Our contribution in this phase was just the benchmarking of an existing tool.

3 Evaluation

We experimented with all three datasets from Task 12, namely Restaurants (Res), Laptops (Lap), and Hidden (Hid) for which the domain was unknown. Details on the datasets are in Pontiki et al. (2015).

The evaluation occurs in two phases. In the first phase, participating systems were evaluated on category detection for Restaurants and Laptops. Additionally, identifying opinion target and the pair $(category, target)$ was requested for the Restaurants domain. In the second phase, the systems were evaluated on polarity detection on all three domains.

3.1 Opinion Category and Target Detection

When evaluating opinion category and target detection (first phase), three measures were taken into account: precision, recall, and F1. For both category and target detection, the baseline methodologies are presented in Pontiki et al. (2015). Table 1 shows the results obtained using our approach compared to the baseline for aspect category detection, whereas Table 2 outlines the results regarding aspect target detection. The results for the pair $(category, target)$ are presented in Table 3.

Table 1: Opinion Category detection.

Domain	Method	P	R	F1
Res	2Phase	0.6556	0.4323	0.5210
Res	1Phase	0.6835	0.4181	0.5188
Res	Baseline			0.5133
Res	1Phase-coref	0.6821	0.4180	0.5184
Res	2Phase-coref	0.6509	0.4090	0.5023
Lap	Baseline			0.4631
Lap	1Phase	0.5066	0.4040	0.4495
Lap	2Phase	0.4773	0.4209	0.4473
Lap	1Phase-coref	0.4834	0.4462	0.4640
Lap	2Phase-coref	0.4689	0.4388	0.4534

The system outperforms the baseline on both approaches for the Restaurants domain. In this domain, the two-phase approach was superior to the one-phase approach. For the laptop domain, however, we scored lower than the baseline. We attribute that to the increased difficulty the coreference resolution step had when processing the review texts in this domain because of the large number of out of vocabulary words (CPU, HD, RAM, etc). Table 1 shows that the results improve when the coreference resolution step is not performed. Nevertheless, for the Restaurant domain, it brought improvements.

Table 2: Opinion Target detection.

Domain	Method	P	R	F1
Res	2Phase	0.5656	0.4373	0.4932
Res	1Phase	0.5764	0.4244	0.4888
Res	Baseline			0.4807
Res	2Phase-exc.	0.5632	0.4354	0.4911
Res	1Phase-exc.	0.5739	0.4225	0.4867

Considering the results for opinion target detection, both versions of our system outperformed the baseline. The two-phase classification achieved better recall in both category and target detection, but worse precision compared to one-phase classification.

We ran some additional experiments to evaluate the use of the exceptions list during target identification. These runs in which the exceptions list were not used are labelled 1Phase-exc and 2Phase-exc in Table 2. The results show that using such a list did not impact the results.

Table 3: Opinion Category and Target pair detection.

Domain	Method	P	R	F1
Res	2Phase	0.4852	0.2722	0.3487
Res	Baseline			0.3444
Res	1Phase	0.4521	0.2734	0.3407
Res	1Phase-coref	0.4694	0.2639	0.3378
Res	2Phase-coref	0.4496	0.2591	0.3288

As for the results for the pair $(category, target)$ the two-phase classification outperforms both the baseline and the one-phase classification. The gain in terms of precision is three percentage points, while recall was slightly reduced. The best configuration was using coreference resolution and the exceptions list.

3.2 Opinion Polarity Detection

Table 4 shows the results in terms of accuracy on opinion polarity. Here, the methodology for the baseline is similar to the ones used for aspect category detection (also described in Pontiki et al. (2015)). In this subtask, we submitted only the results for the one-phase classification.

Table 4: Opinion Polarity detection.

Domain	Method	Accuracy
Res	1Phase	0.7172
Res	Baseline	0.5373
Lap	1Phase	0.6733
Lap	Baseline	0.5701
Hid	Baseline	0.7168
Hid	1Phase	0.6578

The Stanford Core Toolkit uses a model trained on movie reviews, and this was not the same domain of the datasets in the task. Still, the classification results outperformed the baseline on Restaurants and Laptops. However, for the Hidden domain, we scored lower than the baseline.

3.3 Error Analysis

The results obtained with our system are ranked between the 5th (out of 15) and the 14th (out of 22) places. A case by case analysis was performed to identify the most frequent causes of errors. In the task of aspect category classification, the choice of the threshold used during feature selection by the Ranker (0) may have negatively impacted the results. Nevertheless, some feature selection method is necessary since the use of all the words as features greatly increases the processing time.

We used words selected by their Information Gain as seeds to identify the target expression. In our experiments, in many cases, the target was next to the words selected by this strategy. This happens because the $positive$ class had fewer instances than the $negative$ class, and the Information Gain tends to select words that characterize the least frequent class. However, most classification errors happened because this strategy failed to identify infrequent words that corresponded to the expected categories. One possible alternative to mitigate this problem could be the use of synonyms.

The method we used for polarity detection considered the entire sentence. The limitation here is that many sentences contain more than one opinion, which may not convey the same polarity. This could be solved by identifying the context (*i.e.*, a region around the target) and limit the polarity attribution to that region.

4 Conclusion

This paper reports on the experiments that we conducted while taking part on SemEval-2015 Task 12. We showed that classification algorithms, coreference resolution tools, and a syntactic parser may be combined in a category/target detection system.We employed a two-phase approach to classify instances. Our results show that this approach can be an alternative to classify sentences without using lexicons, improving recall with a small decay in precision. As future work, we plan to improve the coreference resolution of review texts so as to further improve recall.

Acknowledgments

This work has been partially funded by CNPq-Brazil project 478979/2012-6. Anderson Kauer receives a scholarship from CNPq-Brazil. We would like to thank the anonymous reviewers for their helpful suggestions and comments.

References

Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H. Witten. 2009. The WEKA data mining software: an update. *SIGKDD Explor. Newsl.*, 11(1):10–18, November.

Minqing Hu and Bing Liu. 2004. Mining and summarizing customer reviews. In *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 168–177, New York, NY, USA. ACM.

Soo-Min Kim and Eduard Hovy. 2004. Determining the sentiment of opinions. In *Proceedings of the 20th International Conference on Computational Linguistics*, Stroudsburg, PA, USA. Association for Computational Linguistics.

Qian Liu, Zhiqiang Gao, Bing Liu, and Yuanlin Zhang. 2013. A logic programming approach to aspect extraction in opinion mining. In *Web Intelligence (WI) and Intelligent Agent Technologies (IAT), 2013 IEEE/WIC/ACM International Joint Conferences on*, volume 1, pages 276–283, Nov.

Bing Liu. 2012. Sentiment analysis and opinion mining. *Synthesis Lectures on Human Language Technologies*, 5(1):1–167.

Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. 2014. The Stanford CoreNLP natural language processing toolkit. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 55–60, Baltimore, Maryland. Association for Computational Linguistics.

Maria Pontiki, Dimitrios Galanis, Haris Papageogiou, Suresh Manandhar, and Ion Androutsopoulos. 2015. Semeval-2015 task 12: Aspect based sentiment analysis. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, Denver, Colorado.

Guang Qiu, Bing Liu, Jiajun Bu, and Chun Chen. 2011. Opinion word expansion and target extraction through double propagation. *Computational Linguistics*, 37(1):9–27.

Richard Socher, Alex Perelygin, Jean Y Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1631–1642.