V3: Unsupervised Generation of Domain Aspect Terms for Aspect Based Sentiment Analysis

Aitor García-Pablos,
Montse Cuadros, Seán Gaines
Vicomtech-IK4 research centre
Mikeletegi 57, San Sebastian, Spain
{agarciap,mcuadros}@vicomtech.org

German Rigau
IXA Group
Euskal Herriko Unibertsitatea,
San Sebastian, Spain
german.rigau@ehu.es

Abstract

This paper presents V3, an unsupervised system for aspect-based Sentiment Analysis when evaluated on the SemEval 2014 Task 4. V3 focuses on generating a list of aspect terms for a new domain using a collection of raw texts from the domain. We also implement a very basic approach to classify the aspect terms into categories and assign polarities to them.

1 Introduction

The automatic analysis of opinions, within the framework of opinion mining or sentiment analysis, has gained a huge importance during the last decade due to the amount of review web sites, blogs and social networks producing everyday a massive amount of new content (Pang and Lee, 2008; Liu, 2012; Zhang and Liu, 2014). This content usually contains opinions about different entities, products or services. Trying to cope with this large amounts of textual data is unfeasible without the help of automatic Opinion Mining tools which try to detect, identify, classify, aggregate and summarize the opinions expressed about different topics (Hu and Liu, 2004) (Popescu and Etzioni, 2005) (Wu et al., 2009) (Zhang et al., 2010). In this framework, aspect based opinion mining systems aim to detect the sentiment at "aspect" level (i.e. the precise feature being opinionated in a clause or sentence).

In this paper we describe our system presented in the SemEval 2014 task 4[1] *Aspect Based Sentiment Analysis* (Pontiki et al., 2014), which focuses on detecting opinionated aspect terms (e.g. *wine list* and *menu* in restaurant domain, and *hard disk* and *battery life* in laptop domain), their categories and polarities in customer review sentences.

The task provides two training datasets, one of restaurant reviews and other of laptop reviews. The restaurant review dataset consists of over 3,000 English sentences from restaurant reviews borrowed from (Ganu et al., 2009). The laptop review dataset consists of over 3,000 English sentences extracted from customer reviews. The task is divided in four different subtasks: subtask 1 aspect term extraction, subtask 2 aspect term polarity detection, subtask 3 aspect category detection, subtask 4 aspect category polarity detection. Our system mainly focused on subtask 1, but we have also participated in the other subtasks.

The paper is organized as follows: section 2 presents our approach, section 3 details the improvement methods used for the aspects term selection and section 4 focus on category and polarity tagging. Finally section 5 presents the results obtained and section 6 draws the conclusions and future work.

2 Our approach

We have adapted the double-propagation technique described in (Qiu et al., 2009; Qiu et al., 2011). This method consists of using a minimal seed list of aspect terms and opinion words and propagate them through an unlabelled domain-related corpus using a set of propagation rules. The goal is to obtain an extended aspect term and opinion word lists. (Qiu et al., 2009) define opinion words as words that convey some positive or negative sentiment polarities. They only extract nouns as aspect terms and adjectives as opinion words, and we assume the same restriction.

The propagation rules have the form of dependency relations and some part-of-speech restrictions. Some rules extract new aspect terms, and others extract new opinion words. Table 1 shows

[1] http://alt.qcri.org/semeval2014/task4/

Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014), pages 833–837,
Dublin, Ireland, August 23-24, 2014.

the rules used in our approach, similar to those detailed in (Qiu et al., 2011) with some modifications. In this table, T stands for *aspect term* (i.e. a word already in the aspect terms set) and O for *opinion word* (i.e. a word already in the opinion words set). W means *any word*. The dependency types used are *amod*, *dobj*, *subj* and *conj*, which stand for *adjectival modifier*, *direct object*, *subject* and *conjunction* respectively. Additional restrictions on the Part-Of-Speech (POS) of the words present in the rule are shown in the third column of the table. The last column indicates to which set (aspect terms or opinion words) the new word is added.

To obtain the dependency trees and word lemmas and POS tags, we use the Stanford NLP tools[2] (De Marneffe et al., 2006). Our initial seed words are just the adjectives *good* and *bad*, which are added to the initial opinion words set. The initial aspect terms set starts empty. Each sentence in the dataset is analysed to obtain its dependency tree and the rules are checked sequentially. If rule is triggered, then the word indicated by that rule is added to the corresponding set (aspect terms or opinion words, depending on the rule). These new words can be then used to trigger the propagation rules later. After the last sentence the process starts from the beginning to check the rules with the newly added words. The process stops when no more words have been added during a full dataset iteration.

3 Selecting aspect term candidates

The double-propagation process populates both sets of domain aspect terms and domain opinion words, but we focus our attention in the aspect terms set. Due to the nature of the process it tends to generate hundreds of different potential aspect terms, many of them being incorrect. We apply some additional processes to improve the list.

3.1 Ranking the aspect terms

One way to reduce the undesired terms is to rank them, pushing the incorrect aspect terms to the bottom of the list and using only a certain subset of top ranked terms. In order to rank this list we have modelled the double-propagation process as a undirected graph population process. Each new aspect term or opinion word discovered by applying a propagation rule is added as a vertex to the graph. The rule used to extract the new word is added as an edge to the graph, connecting the original word and the newly discovered word.

We have applied the well-known PageRank algorithm (Brin and Page, 1998) to score the vertices of the graph. To calculate the PageRank scores we have used the JUNG framework[3] (OMadadhain et al., 2005), a set of Java libraries to work with graphs. The value of the alpha parameter that represents the probability of a random jump to any node of the graph has been left at 0.15 (in the literature it is recommended an alpha value between 0.1 and 0.2). The aspect terms are then ordered using their associated score, being the most relevant aspect term, the one with the highest score. Then the list can be trimmed to a certain amount of top ranked terms, trying to balance the precision and recall of the resulting list.

3.2 Filtering undesired words

The double-propagation method always introduces many undesired words. Some of these undesired words appear very frequently and are combined with a large number of words. So, they tend to also appear in high positions in the ranking. Many of these words are easy to identify, and they are not likely to be useful aspect terms in any domain. Examples of these words are: *nothing*, *everything*, *thing*, *anyone*, *someone*, *somebody*, etc. In this work we use a domain agnostic stop word list to deal with this kind of words. The authors of the original double-propagation approach use some clause and frequency based heuristics that we do not employ here.

3.3 Detecting multiword terms

Many aspect terms are not just single words, but compounds and multiword terms (e.g. *wine list*, *hard disk drive*, *battery life*, etc.). In the original double-propagation paper, the authors consider adjacent nouns to a given aspect term as multiword terms and perform an *a posteriori* pruning based on the frequency of the combination. We have tried to add multiword terms without increasing the amount of noise in the resulting list. One of the approaches included in the system exploits WordNet [4] (Fellbaum, 1999), and the following simple rules:

[2] http://nlp.stanford.edu/software/lex-parser.shtml

[3] http://jung.sourceforge.net

[4] http://wordnet.princeton.edu/

Rule	Observations	Constraints	Action
R11	O → amod→W	W is a noun	W→T
R12	O→dobj→W1 ←subj←W2	W2 is a noun	W2→T
R21	T ← amod ← W	W is an adjective	W→O
R22	T → subj → W1 ← dobj ← W2	W2 is an adjective	W2→ O
R31	T → conj → W	W is a noun	W → T
R32	T → subj → W1 ← dobj ← W2	W2 is a noun	W → T
R41	O → conj → W	W is an adjective	W→ O
R42	O → Dep1 → W1 ← Dep2 ← W2	Dep1==Dep2, W2 is an adjective	W2→ O

Table 1: Propagation rules.

- If word N and word N+1 are nouns, and the combination is an entry in WordNet (or in Wikipedia, see below). E.g.: *battery life*
- If word N is an adjective and word N+1 is a noun, and the combination is an entry in WordNet. E.g.: *hot dog*, *happy hour*
- If word N is an adjective, word N+1 is a noun, and word n is a relational adjective in WordNet (lexical file 01). E.g.: *Thai food*

In order to improve the coverage of the WordNet approach, we also check if a combination of two consecutive nouns appears as a Wikipedia article title. Wikipedia articles refer to real word concepts and entities, so if a combination of words is a title of a Wikipedia article it is very likely that this word combination is also meaningful (e.g. *DVD player*, *USB port*, *goat cheese*, *pepperoni pizza*). We limit the lookup in Wikipedia titles just to combination of nouns to avoid the inclusion of incorrect aspect terms.

4 Assigning categories and polarities

Despite we have focused our attention on acquiring aspect terms from a domain, we have also participated in the rest of subtasks: grouping aspect terms into a fixed set of categories, and assigning polarities to both aspect terms and categories.

To group the aspect terms into categories, we have employed WordNet similarities. The idea is to compare the detected aspect terms against a term or group of terms representative of the target categories. In this case the categories (only for restaurants) were *food*, *service*, *price*, *ambience* and *anecdotes/miscellaneous*.

Initially, the representative word for each category (except for the *anecdotes/miscellaneous*) was the name of the category itself. We use the similarity measure described by (Wu and Palmer, 1994). Detected aspect terms are compared to the set of representative words on each category, and they are assigned to the category with a higher similarity result. For example using this approach, the similarity between *food* and *cheese* is 0.8, while similarity between *service* and *cheese* is 0.25, and between *price* and *cheese* is 0.266. Thus, in this case *cheese* is assigned to the category *food*.

If the similarity does not surpass a given minimum threshold (manually set to 0.7), the current aspect term is not assigned to the category to avoid assigning a wrong category just because the other were even less similar. After classifying the aspect terms of a given sentence into categories, we assign those categories to the sentence. If no category has been assigned, then we use the *anecdotes/miscellaneous* category as the default one.

This approach is quite naive and it has many limitations. It works quite well for the category *food*, classifying ingredients and meals, but it fails when the category or the aspect terms are more vague or abstract. In addition, we do not perform any kind of word sense disambiguation or sense pruning, which probably would discard unrelated senses.

For detecting the polarity we have used the SentiWords (Guerini et al., 2013; Warriner et al., 2013) as a polarity lexicon. Using direct dependency relations between aspect terms and polarity bearing words we assign the polarity value from the lexicon to the aspect term. We make a simple count of the polarities of the aspect terms classified under a certain category to assign the polarity of that category in a particular sentence.

5 Evaluation

The run submitted to the SemEval task 4 competition was based on 25k unlabelled sentences extracted from domain related reviews (for restaurants and laptops) obtained by scraping different websites. We used these unlabelled sentences to execute our unsupervised system to generate and

Restaur. aspect terms	Precision	Recall	F-score
SemEval Baseline	0.525	0.427	0.471
V3 (S)	**0.656**	0.562	0.605
V3 (W)	0.571	0.641	0.604
V3 (W+S)	0.575	**0.645**	**0.608**

Table 2: Results on the restaurant review test set.

Laptops aspect terms	Precision	Recall	F-score
SemEval Baseline	**0.443**	0.298	0.356
V3 (S)	0.265	0.276	0.271
V3 (W)	0.321	0.425	**0.366**
V3 (W+S)	0.279	**0.444**	0.343

Table 3: Results on the laptop review test set.

rank the aspect term lists. Then we used those aspect term lists to annotate the sentences using a simple lemma matching approach between the words. The generated aspect term lists were limited to the first ranked 550 items after some initial experiments with the SemEval training sets.

The SemEval test datasets (restaurants and laptops) contain about 800 sentences each. The restaurant dataset contains 1,134 labelled gold aspect term spans, and the laptop dataset contains 634 labelled gold aspect term spans. We compare the results against the SemEval baseline which is calculated using the scripts provided by the SemEval organizers. This baseline splits the dataset into *train* and *test* subsets, and uses all the labelled aspect terms in the *train* subset to build a dictionary of aspect terms. Then it simply uses that dictionary to label the test subset for evaluation.

Tables 2 and 3 show the performance of our system with respect to the baselines in both datasets. "V3 (S)" stands for our system only using the SemEval test data (as our approach is unsupervised it learns from the available texts for the task). (W) refers to the results using our own dataset scraped from the Web. Finally (W+S) refers to the results using both SemEval and our Web dataset mixed together. The best results are highlighted in bold. For subtask 1, although our system outperforms the baseline in terms of F-score in both datasets, in the competition our system obtained quite modest results ranking 24th and 26th out of 29 participants

Restaur. categories	Precision	Recall	F-score
SemEval Baseline	**0.671**	**0.602**	**0.638**
V3	0.638	0.569	0.602

Table 4: Results on restaurant category detection using the test set.

Polarity detection accuracy	Baseline	V3
Restaur. aspect terms	**0.642**	0.597
Restaur. categories	**0.656**	0.472
Laptop aspect terms	0.510	**0.538**

Table 5: Results for the polarity classification subtasks (subtasks 2 and 4).

for restaurants and laptops respectively.

One of the most important source of errors are the multiword aspect term detection. In the SemEval datasets, about the 25% of the gold aspect terms are multiword terms. In both datasets we find a large number of names of recipes and meals, composed by two, three or even more words, which cannot appear in our aspect term lists because we limit the multiword length up to two words.

As mentioned in the introduction our approach focuses mainly in the aspects so the approach for detecting categories and polarities needs more attention. Table 4 presents our results on category detection and table 5 our results on polarities. The results are quite poor so we do not comment on them here. We will address these subtasks in future work.

6 Conclusions and future work

In this paper we propose a simple and unsupervised system able to bootstrap and rank a list of domain aspect terms from a set of unlabelled domain texts. We use a double-propagation approach, and we model the obtained terms and their relations as a graph. Then, we apply the PageRank algorithm to score the obtained terms. Despite the modest results, our unsupervised system for detecting aspect terms performs better than the supervised baseline. In our future work we will try to improve the way we deal with multiword terms to reduce the amount of incorrect aspect terms and generate a better ranking. We also plan to try different methods for the category grouping, and explore knowledge-based word sense disambiguation methods for improving the current system.

Acknowledgements

This work has been partially funded by SKaTer (TIN2012-38584-C06-02) and OpeNER (FP7-ICT-2011-SME- DCL-296451).

References

Sergey Brin and Lawrence Page. 1998. The anatomy of a large-scale hypertextual web search engine. *Computer networks and ISDN systems*, 30(1):107–117.

Marie-Catherine De Marneffe, Bill MacCartney, Christopher D Manning, et al. 2006. Generating typed dependency parses from phrase structure parses. In *Proceedings of LREC*, volume 6, pages 449–454.

Christiane Fellbaum. 1999. *WordNet*. Wiley Online Library.

Gayatree Ganu, N Elhadad, and A Marian. 2009. Beyond the Stars: Improving Rating Predictions using Review Text Content. *WebDB*, (WebDB):1–6.

Marco Guerini, Lorenzo Gatti, and Marco Turchi. 2013. Sentiment analysis: How to derive prior polarities from sentiwordnet. *arXiv preprint arXiv:1309.5843*.

Minqing Hu and Bing Liu. 2004. Mining opinion features in customer reviews. *AAAI*.

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

Joshua OMadadhain, Danyel Fisher, Padhraic Smyth, Scott White, and Yan-Biao Boey. 2005. Analysis and visualization of network data using jung. *Journal of Statistical Software*, 10(2):1–35.

Bo Pang and Lillian Lee. 2008. Opinion mining and sentiment analysis. *Foundations and trends in information retrieval*, 2(1-2):1–135.

Maria Pontiki, Dimitrios Galanis, John Pavlopoulos, Harris Papageorgiou, Ion Androutsopoulos, and Suresh Manandhar. 2014. Semeval-2014 task 4: Aspect based sentiment analysis. *Proceedings of the International Workshop on Semantic Evaluation (SemEval)*.

AM Popescu and Oren Etzioni. 2005. Extracting product features and opinions from reviews. *Natural language processing and text mining*, (October):339–346.

Guang Qiu, Bing Liu, Jiajun Bu, and Chun Chen. 2009. Expanding Domain Sentiment Lexicon through Double Propagation. *IJCAI*.

Guang Qiu, Bing Liu, Jiajun Bu, and Chun Chen. 2011. Opinion word expansion and target extraction through double propagation. *Computational linguistics*, (July 2010).

Amy Beth Warriner, Victor Kuperman, and Marc Brysbaert. 2013. Norms of valence, arousal, and dominance for 13,915 english lemmas. *Behavior research methods*, 45(4):1191–1207.

Zhibiao Wu and Martha Palmer. 1994. Verbs semantics and lexical selection. In *Proceedings of the 32nd annual meeting on Association for Computational Linguistics*, pages 133–138. Association for Computational Linguistics.

Yuanbin Wu, Qi Zhang, Xuanjing Huang, and Lide Wu. 2009. Phrase dependency parsing for opinion mining. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 3-Volume 3*, pages 1533–1541. Association for Computational Linguistics.

Lei Zhang and Bing Liu. 2014. Aspect and Entity Extraction for Opinion Mining. *Data Mining and Knowledge Discovery for Big Data*.

L Zhang, Bing Liu, SH Lim, and E O'Brien-Strain. 2010. Extracting and ranking product features in opinion documents. *Proceedings of the 23rd International Conference on Computational Linguistics*, (August):1462–1470.