Dictionary-Based Sentiment Analysis Applied to a Specific Domain

Laura Cruz[1], José Ochoa[1,2], Mathieu Roche[3], and Pascal Poncelet[4]

[1] Universidad Nacional de San Agustín, Perú
lcruzq@unsa.edu.pe
[2] Universidad Católica San Pablo, Perú
jeochoa@ucsp.edu.pe
[3] TETIS (AgroParisTech, Cirad, Cnrs, Irstea), France
mathieu.roche@cirad.fr
[4] LIRMM (Cnrs, Univ. Montpellier), France
pascal.poncelet@lirmm.fr

Abstract. The web and social media have been growing exponentially in recent years. We now have access to documents bearing opinions expressed on a broad range of topics. This constitutes a rich resource for natural language processing tasks, particularly for sentiment analysis. Nevertheless, sentiment analysis is usually difficult because expressed sentiments are usually topic-oriented. In this paper, we propose to automatically construct a sentiment dictionary using relevant terms obtained from web pages for a specific domain. This dictionary is initially built by querying the web with a combination of opinion terms, as well as terms of the domain. In order to select only relevant terms we apply two measures $AcroDef_{MI3}$ and *TrueSkill*. Experiments conducted on different domains highlight that our automatic approach performs better for specific cases.

Keywords: Text Mining, Web Mining, Sentiment analysis

1 Introduction

The web and social media have been growing exponentially in recent years, which constitutes a rich resource for sentiment analysis tasks. For instance, social networking sites enable users to express their thoughts and opinions about products [1] and companies are increasingly taking these opinions into account to make better decisions [11]. Sentiment analysis currently involves a process to identify the sentiment orientation of opinions. The latter are highly unstructured by nature, thus requiring the application of Natural Language Processing (NLP) techniques [17].

Obviously, documents may include opinions about several topics, but terms[5] used to express opinions are usually specific and highly correlated to a particular domain [6]. For instance, the sentence *"The fruit is organic"* would be very

[5] In this paper, we use *term* in order to characterize linguistic features.

unusual in movie domain and then irrelevant in this case. However, it is obviously useful in agricultural domain. Both machine learning and dictionary-based approaches have been proposed in the literature to tackle these issues. For instance, a machine learning method applying text categorization techniques was proposed in [12]. By this method, graphs, minimum cut formulation, context, and domain were considered to extract subjective portions of documents.

Some dictionary-based approaches are currently available for general applications (e.g. SentiWordNet[6]). They are not really appropriate for specific domains and new approaches have been developed to automatically learn the dictionary. These methods generally assume that positive (resp. negative) adjectives or verbs appear more frequently near a positive (resp. negative) seed term [9]. For instance, in [16, 19], the authors propose an unsupervised learning algorithm for getting a dictionary in order to classify reviews considering seed terms to calculate the semantic orientation of phrases.

In this paper, we propose a new approach to automatically learn expressed opinions. We first focus on a new method for selecting relevant candidate terms from a set of documents. As many candidate terms may be extracted we propose to use two different but complementary measures to select the most representative ones: $AcroDef_{MI3}$ and *TrueSkill*. Furthermore, in order to highlight the fact that of our approach is well useful for extracting terms for a specific domain we compare our proposal to the well-known SentiWordNet.

The paper is organized as follows. Our approach is presented in Section 2. The experimental setup is described in Section 3. In Section 4, we present and discuss the obtained results. Concluding remarks are presented in Section 5.

2 Our approach

The main process of our proposal is depicted in Figure 1 which involves the following steps:

1. First, a huge corpus for a specific domain is created by querying the web in order to get positive and negative documents relative to this domain.
2. Some pre-processing methods are performed over the documents in order to get the language of the document, remove tags, scripts, images, etc.
3. This step forms the core of the process. It focuses on the selection of terms that could be classified as positive (resp. negative) for the domain. To do, so first, a part-of-speech tagging is performed on documents in order to focus on nouns and adjectives since they are well relevant for extracting opinions[7].

[6] http://sentiwordnet.isti.cnr.it

[7] For simplicity, in this paper, we only report experiments that have been conducted on nouns and adjectives. Other experiments have been done by using adverbs and verbs.

In order to find such terms (adjectives or nouns) as in [9], we follow the following hypothesis: *the closer a positive (resp. negative) adjective/noun to another positive (resp. negative) adjective/noun, the more positive (resp. negative) it is*. Accordingly, to this, we apply a window size algorithm for selecting the relevant terms closest to given seed terms. Finally, as many candidate terms may be generated, an efficient filtering approach is applied by using $AcroDef_{MI3}$ and *TrueSkill* to select the most relevant positive and negative terms. Finally based on the results, two lexicons of positive and negative terms are generated.

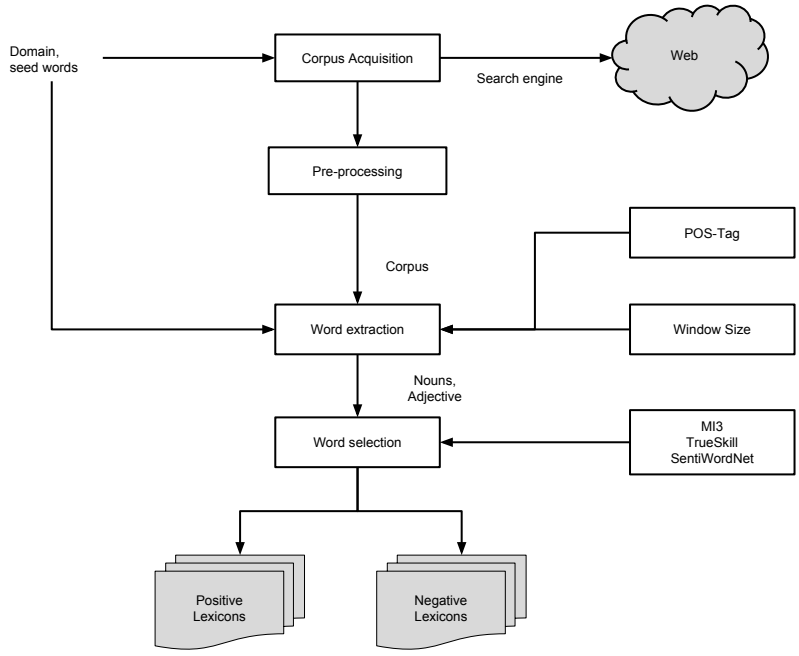


Fig. 1: Lexicons are automatically obtained from web pages for a specific domain filtered by seed terms. Extraction of relevant terms are then obtained by evaluating the relevance of candidate terms that are obtained after the analysis of the documents.

Basically, our approach could be used in many different domains. So in order to highlight its generality, experiments have been conducted on four different domains. They will be described in the experimental section. In the next sections, we describe more in detail the different steps.

2.1 Corpus Acquisition

It is now well admitted that some terms can be positive, neutral, or negative depending of the domain. Nevertheless, some terms are positive or negative irrespective of the domain (e.g. *good*). The main idea of our approach is thus to start the process by considering adjectives which are positive or negative in all domains. These terms will be considered as *seed terms*. We thus select the two following seed sets: P = {*good*, *nice*, *excellent*, *positive*, *fortunate*, *correct*, *superior*}, Q = {*bad*, *nasty*, *poor*, *negative*, *unfortunate*, *wrong*, *inferior*}. From these sets, we ensure a positive (resp. negative) web page retrieved from the other web pages related to a given domain. The following query illustrates an example of what is generated to get only positive documents about Genetic Modified Organism (GMO):

+GMO +good -bad -nasty -poor -negative -unfortunate -wrong -inferior

where + and - mean that the document must have (+) or not (-) a given term. At the end, we are thus provided with positive and negative web pages denoted by corpus$^+$, corpus$^-$. Each corpus is splitted by the term used in the query. For instance, by considering the previous example we have in corpus$^+$ a set of document focusing mainly on *good*, i.e. no other positive terms are within documents, and more importantly not having a negative term (e.g *-bad -nasty*, and so on).

In the next section, we focus on the terms that are close to the seed terms by considering POS-Tagging as well as a window size algorithm.

2.2 Term extraction

First of all HTML tags, scripts, blank spaces and stop words are removed from web pages. We apply a part-of-speech tagger (in our case we have experimented *TreeTagger*[8]) to keep only adjectives and nouns. To be relevant with the previous hypothesis that an opinion candidate term is close to a seed term, a window size algorithm has been used. It aims at finding terms in both left and right sides of a seed term given a distance k. This distance is then the number of left (resp. right) terms of a given seed term. By varying k we are this able to better extract the most correlated opinion terms. For instance, by applying *TreeTagger* to retrieve adjectives (i.e. `JJ`) and nouns (i.e. `NNS`) as illustrated in Figure 2.

In Figure 2, the '*good*' term is a positive seed term and its nearest adjective is '*safe*' given a $k = 1$ distance. Likewise, '*scientific*' and '*studies*' terms are retrieved with distance of $k = 2$. In addition, '*safe*' is a positive candidate term because it occurs close to the positive seed term (i.e. *good*). In this sense, we can have a set of opinion terms that can be candidates to be included into the resulting dictionaries.

[8] http://www.cis.uni-muenchen.de/~schmid/tools/TreeTagger/

Fig. 2: An example of applying a window size algorithm on the *good* seed term.

To get the correlation score and the usefulness on our specific domain (here GMO in the example) of each extracted term two approaches have been used: $AcroDef_{MI3}$ and *TrueSkill* and this is described in the next section.

2.3 Candidate Term Selection

From the set of candidate terms, we thus have to filter the most relevant ones: the positive (resp. negative) terms that are very specific to a domain. In order to select the relevant candidate opinion terms, we propose to adapt the statistical measure $AcroDef_{MI3}$ [14, 15] (see Algorithm 1 where we illustrate only for positive terms, the process for the negative terms is similar) as well as a probabilistic measure based on *TrueSkill* [10, 8] (see Algorithm 2).

The $\boldsymbol{AcroDef_{MI3}}$ measure: To filter associations extracted at the previous step, we use a ranking function in order to delete the irrelevant adjectives associations placed at the end of a list. Several quality measures in the literature are based on ranking functions. They are brought out of various fields: Association rules extraction [7], terminology extraction [4], and so forth. One of the most commonly used measures to compute a sort of relationship between the terms, called co-occurrence, is Church's Mutual Information (MI). The formula is the following [3]:

$$MI(x, y) = \log \frac{nb(x, y)}{nb(x)nb(y)} \tag{1}$$

This measure tends to extract rare and specific co-occurrences according to [4]. The Cubic Mutual Information ($MI3$) is an empirical measure based on MI that enhances the impact of frequent co-occurrences. This measure defined by formula (2) gives interesting results [5, 18].

$$MI3(x, y) = \log \frac{nb(x, y)^3}{nb(x)nb(y)} \tag{2}$$

Like many other studies based on web resources, the nb function used by the MI and $MI3$ measures represents the number of web pages provided by a search

engine.

Our approach relies on the dependence calculation of two terms, i.e. seed terms (st), and candidate term (ct). This is based on the number of pages given by a search engine with the queries 'st and ct' and 'ct and st'. This dependence is computed in a given domain D (for instance $D = \{GMO\}$). Then we apply $AcroDef_{MI3}$ (formula (3)) described in [14]:

$$AcroDef_{MI3}(st, ct) = \log \frac{(nb(st \text{ and } ct \text{ and } D) + nb(ct \text{ and } st \text{ and } D))^3}{nb(st \text{ and } D) \times nb(ct \text{ and } D)} \quad (3)$$

The selection of terms is based on the application of Algorithm 1.

Algorithm 1 Term selection algorithm using $AcroDef_{MI3}$

Require: *corpus*, *seed terms* = P, *terms of the domain*
Ensure: correlation score values for each *term*
1: **for** each *corpus* **do**
2: $terms^+ = window\_size(corpus^+, P)$
3: **for** term in $terms^+$ **do**
4: given each seed term and terms of the domain compute the correlation score:
5: $score \leftarrow max(AcroDef_{MI3})$

The *TrueSkill* measure: Unlike $AcroDef_{MI3}$, in *TrueSkill*, terms are extracted for each positive (resp. negative) page against k random negative (resp. positive) pages, and then the scores for each term are computed. Therefore, after having this outcome *TrueSkill* can give a score for each term of the positive page. This score depends on how many times it appears in the positive page so that it increases or decreases if it is also found on a negative page. The principle of *TrueSkill* is illustrated in Figure 3.

In Figure 3, we have $S = \{s_{1,1}, s_{1,2}, ..., s_{1,n}\}$ and $S = \{s_{2,1}, s_{2,2}, ..., s_{2,n}\}$ where s are the learned score value for each term in positive and negative web pages. p is the performance value for each term, which depends mainly of previously score s of the term; t is the sum of total performance for each term in the corpus. As *TrueSkill* learns s according its outcome of matches, we set a high punctuation for $corpus^+$, and less punctuation for $corpus^-$. This process is detailed in Algorithm 2.

Finally, *TrueSkill* gives a score to each term of the corpus in a match, and those values are updated for each match. On one hand, if a term is often found in a $corpus^-$ its value tends to decrease. On the other hand, if it is in a $corpus^+$

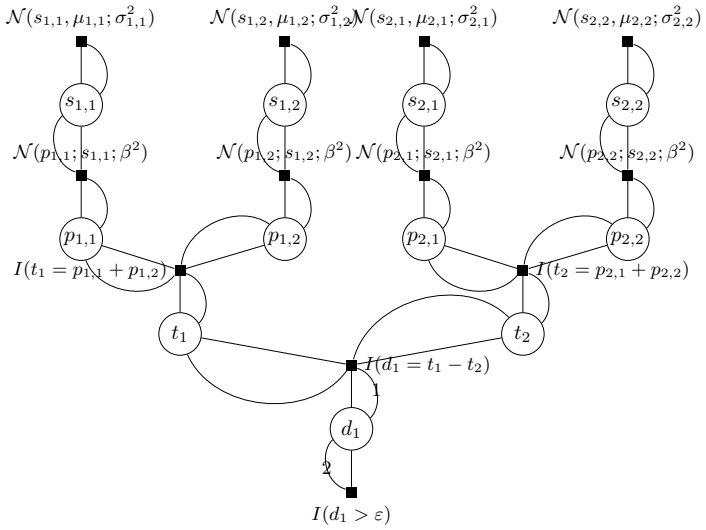


Fig. 3: *TrueSkill* measure provides a score for each term selected given the positive and negative corpora.

its value will increase. If the term is found in both corpora it tends to be constant. The velocity of the score increases or decreases depending on the term combination in each corpus and the number of matches.

3 Experiments

In order to evaluate our approach, experiments over four datasets were conducted. First we focused on both measures $AcroDef_{MI3}$ and *TrueSkill* to evaluate their efficiency for pruning candidate terms. Second we evaluated the opinion classification task with both measures. Finally in order to really evaluate our automatic obtained dictionaries, classification are also compared with a general

Algorithm 2 Term selection algorithm using *TrueSkill*

Require: *corpus*, *seed_terms(P, Q)*
Ensure: correlation score values for each *term*
1: $k = 10$ number of matches for each *corpus*.
2: **for** each *corpus* **do**
3: $terms^+ = window\_size(corpus^+, P)$
4: **for** k random $corpus^-$ **do**
5: $terms^- =$ window_size$(corpus^-, Q)$
6: given each term compute the correlation score:
7: $score \leftarrow TrueSkill(terms^+, terms^-, t = [1, 2])$

dictionary. *SentiWordnet* is a lexical resource for opinion mining, mainly it comprises 21479 adjectives and 117798 nouns, and assigns three sentiment scores to each word, i.e. positive, negative, and neutral.

3.1 Datasets

In order to show that our approach is generic, we use four datasets on very different domains: agriculture, movie, kitchen, and book:

- On the agricultural domain, tweets have been collected, and have been manually labeled. We obtained a corpus of 183 tweets, i.e. 72 positive and 111 negative tweets.
- Available resources[9] of the movie domain were introduced in [13] with 1000 positive and 1000 negative opinion documents.
- Finally, the kitchen and book domains[10] introduced in [2] have both 1000 positive and 1000 negative opinions.

Table 1 shows the number of candidate terms related to each domain after applying the window size algorithm with $k = 1$, and for each seed term we get the first 20 web pages.

Table 1: Total of inferred lexicon terms by domain.

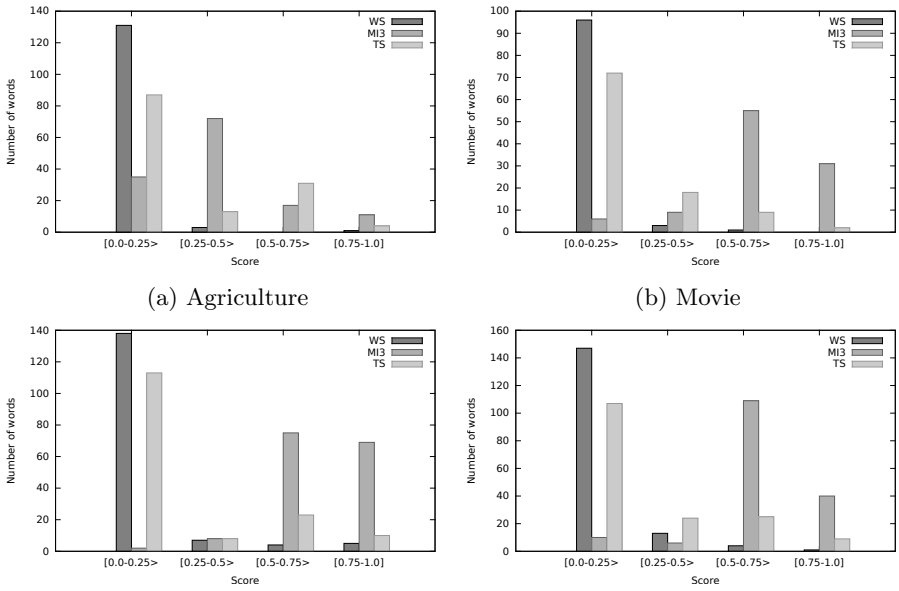
Lexicon	**Agriculture**		**Movie**		**Kitchen**		**Book**	
	P	N	P	N	P	N	P	N
Adjective	146	83	104	72	157	26	168	87
Noun	334	207	247	169	335	81	330	197

3.2 Results of $AcroDef_{MI3}$ and *TrueSkill*

In the following WS stands for the terms extracted after the window size algorithm. We thus compared $MI3$: seed terms with WS followed by $AcroDef_{MI3}$ and TS: seed terms with WS followed by *TrueSkill*.

Figure 4 shows the normalized scores of all measures over each term by using the min-max scale algorithm. The window score is based on the frequency of a given term after applying the window size algorithm. As expected we thus have a high number of terms with low score. We can notice that terms have the more distributed score after applying $AcroDef_{MI3}$ and *TrueSkill*.

[9] http://www.cs.cornell.edu/People/pabo/movie-review-data/
[10] https://www.cs.jhu.edu/~mdredze/datasets/sentiment/



(a) Agriculture (b) Movie

(c) Kitchen (d) Books

Fig. 4: Lexicons for each domain with their normalized score.

3.3 Classification

As the context of $SentiWordNet$ is not exactly the same as ours, the neutral class is considered as follows. For a term, we compute the difference between its positive and negative score and if the result is greater than zero we assign the term as positive otherwise as negative.

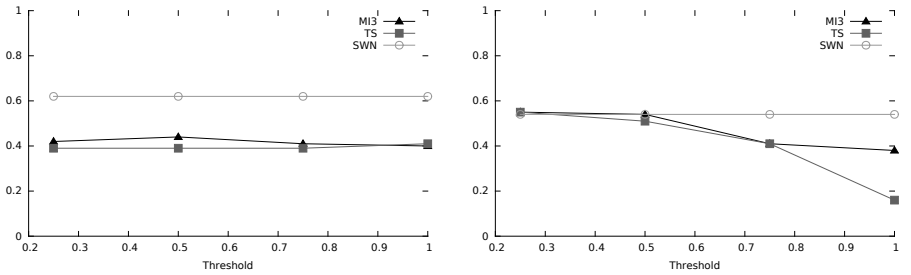
We have positive and negative lexicons (dictionaries) for each dataset (i.e. agriculture, movie), as shown in Table 1. In order to validate the algorithms we calculate *F-Score* values for each domain. Figures 5, 6, 7, and 8 show the *F-Score* values using the built lexicons with our approach and SentiWordNet. On our experiments, the *F-Score* is evaluated using the lexicons with a score greater than a given threshold.

4 Discussion of the results

In order to evaluate *F-Score* results, Table 2 shows the high values obtained for each dataset when the inferred lexicons for each domain are considered. For predicting negative elements, the *F-Score* values of *TrueSkill* are 0.66 and 0.62 for movie and book domains respectively, and 0.55 for agriculture domain based on $AcroDef_{MI3}$.

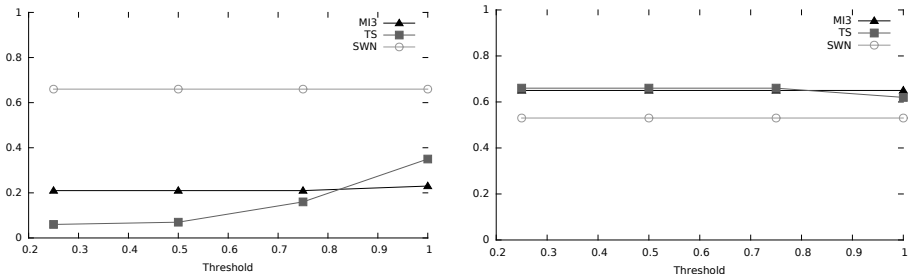
Table 2: Top F-score result of each domain classification, TrueSkill (TS) improves the results of *SentiWordNet* (SWN) in some cases.

	Agriculture		**Movie**		**Kitchen**		**Book**	
	P	N	P	N	P	N	P	N
Approaches	SWN	MI3	SWN	TS	SWN	SWN	SWN	TS
F-Score	0.62	**0.55**	0.66	**0.66**	0.70	0.54	0.67	**0.62**



(a) Positive (b) Negative

Fig. 5: *F-Score* results for agriculture tweet classification.

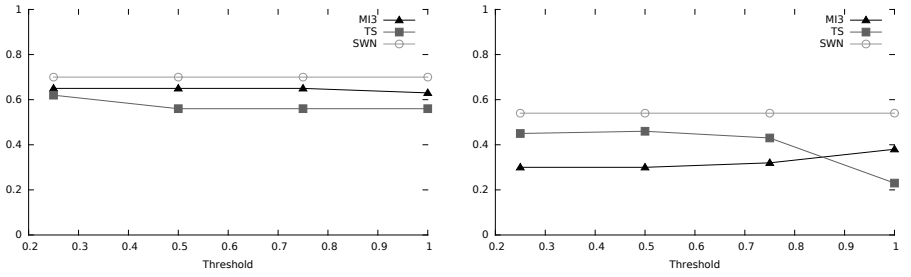


(a) Positive (b) Negative

Fig. 6: *F-Score* results for movie review classification.

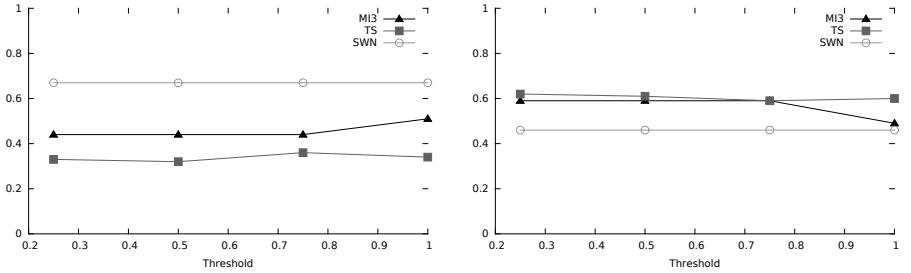
To sum up, our approach performs better with *F-Score* results than SentiWordNet for negative reviews. However, when positive reviews are considered, SentiWordNet performs better.

Figure 4 shows that kitchen domain is more generic than other domains due to the high number of terms ($\approx$ 70) with a high score ($\approx$ 0.9) obtained with $AcroDef_{MI3}$. This could explain that SentiWordNet performs better than *TrueSkill* and $AcroDef_{MI3}$ for positive and negative reviews for this domain.



(a) Positive (b) Negative

Fig. 7: *F-Score* results for kitchen review classification.



(a) Positive (b) Negative

Fig. 8: *F-Score* results for book review classification.

5 Conclusion

In this paper, we proposed a dictionary-based algorithm for sentiment analysis. Our approaches used $AcroDef_{MI3}$ and *TrueSkill* to compute an association score between each term and its sentiment orientation (i.e. positive, negative). The extraction of these new terms related to each domain is obtained using the window size algorithm. This enables us to automatically create dictionaries that have been proved useful to identify positive and negative documents of specific domains.

In future work, we plan to extend our approach to other languages (e.g. French and Spanish), and we would like to study the behavior of our methods with other domains by using multi-word terms in our lexicons.

Acknowledgement

This work has been supported and funded by FONDECYT and SONGES project[11] (FEDER and Occitanie).

[11] http://textmining.biz/Projects/Songes

References

1. Abdelmalek Amine, Reda Mohamed Hamou, and Michel Simonet. Detecting opinions in tweets. *International Journal Of Data Mining And Emerging Technologies*, 3(1):23–32, 2013.
2. John Blitzer, Mark Dredze, and Fernando Pereira. Biographies, bollywood, boomboxes and blenders: Domain adaptation for sentiment classification. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics (ACL'07)*, pages 187–205, 2007.
3. Kenneth W. Church and Patrick Hanks. Word association norms, mutual information, and lexicography. *Computational Linguistics*, 16:22–29, 1990.
4. Béatrice Daille. Study and implementation of combined techniques for automatic extraction of terminology. In *The Balancing Act: Combining Symbolic and Statistical Approaches to Language, MIT Press*, pages 49–66, 1996.
5. Doug Downey, Matthew Broadhead, and Oren Etzioni. Locating complex named entities in web text. In *Proceedings of the 20th international joint conference on Artifical intelligence (IJCAI'07)*, pages 2733–2739, 2007.
6. Benjamin Duthil, Francois Trousset, Mathieu Roche, Grard Dray, Michel Planti, Jacky Montmain, and Pascal Poncelet. Locating complex named entities in web text. In *Proceedings of the 22nd International Conference on Database and Expert Systems Applications (DEXA'11)*, pages 457–465, 2007.
7. Fabrice Guillet and Howard J. Hamilton. *Quality Measures in Data Mining*. Springer Verlag, 2007.
8. Shengbo Guo, Scott Sanner, Thore Graepel, and Wray L. Buntine. Score-based bayesian skill learning. In *Machine Learning and Knowledge Discovery in Databases - European Conference, ECML PKDD 2012, Bristol, UK, September 24-28, 2012. Proceedings, Part I*, pages 106–121, 2012.
9. Ali Harb, Michel Plantie, Gerard Dray, Mathieu Roche, Francois Trousset, and Pascal Poncelet. Web opinion mining: How to extract opinions from blogs? In *Proceedings of the 5th International Conference on Soft Computing As Transdisciplinary Science and Technology (CSTST'08)*, pages 211–217, 2008.
10. Ralf Herbrich, Tom Minka, and Thore Graepel. Trueskill(tm): A bayesian skill rating system. In *Advances in Neural Information Processing Systems 20*, pages 569–576. MIT Press, January 2007.
11. Edison Marrese-Taylor, Juan D. Velsquez, Felipe Bravo-Marquez, and Yutaka Matsuo. Identifying customer preferences about tourism products using an aspect-based opinion mining approach. *Procedia Computer Science*, 22(0):182 – 191, 2013.
12. Bo Pang and Lillian Lee. A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. In *Proceedings of the ACL*, pages 271–278, 2004.
13. Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. Thumbs up?: Sentiment classification using machine learning techniques. In *Proceedings of the ACL-02 Conference on Empirical Methods in Natural Language Processing (EMNLP'02)*, pages 79–86, 2002.
14. Mathieu Roche and Violaine Prince. Acrodef: A quality measure for discriminating expansions of ambiguous acronyms. In *Proceedings of the 6th International and Interdisciplinary Conference: Modeling and Using Context (CONTEXT'07), LNCS, Springer*, pages 411–424, 2007.
15. Mathieu Roche and Violaine Prince. A web-mining approach to disambiguate biomedical acronym expansions. *Informatica (Slovenia)*, 34(2):243–253, 2010.

16. Peter D. Turney. Thumbs up or thumbs down?: Semantic orientation applied to unsupervised classification of reviews. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics (ACL'02)*, pages 417–424, 2002.
17. Raisa Varghese and M. Jayasree. Aspect based sentiment analysis using support vector machine classifier. In *Proceedings of the International Conference on Advances in Computing, Communications and Informatics (ICACCI'13)*, pages 1581–1586, Aug 2013.
18. Jordi Vivaldi, Lluis Marquez, and Horacio Rodriguez. Improving term extraction by system combination using boosting. In *Proceedings of the 12th European Conference on Machine Learning (ECML'01)*, pages 515–526, 2001.
19. Guangwei Wang and Kenji Araki. Modifying so-pmi for japanese weblog opinion mining by using a balancing factor and detecting neutral expressions. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics (NAACL-Short'07)*, pages 189–192, 2007.