LSIS at SemEval-2016 Task 7: Using Web Search Engines for English and Arabic Unsupervised Sentiment Intensity Prediction

Amal Htait *,+
amal.htait@lsis.org
*Aix-Marseille University, CNRS
LSIS UMR 7296
13397, Marseille
France

Sebastien Fournier *,+
sebatien.fournier@lsis.org

Patrice Bellot *,+
patrice.bellot@lsis.org
+Aix-Marseille University, CNRS
CLEO OpenEdition UMS 3287
13451, Marseille
France

Abstract

In this paper, we present our contribution in SemEval2016 task7[1]: Determining Sentiment Intensity of English and Arabic Phrases, where we use web search engines for English and Arabic unsupervised sentiment intensity prediction. Our work is based, first, on a group of classic sentiment lexicons (e.g. Sentiment140 Lexicon, SentiWordNet). Second, on web search engines' ability to find the co-occurrence of sentences with predefined negative and positive words. The use of web search engines (e.g. Google Search API) enhance the results on phrases built from opposite polarity terms.

1 Introduction

A sentiment lexicon is a list of words and phrases, such as ”excellent”, ”awful” and ”not bad”, each is being assigned with a positive or negative score reflecting its sentiment polarity and strength. Sentiment lexicon is crucial for sentiment analysis (or opining mining) as it provides rich sentiment information and forms the foundation of many sentiment analysis systems (Pang and Lee, 2008; Liu, 2012).

Sentence intensity is essential when we need to compare sentences having the same polarity orientation. It is expressed by words or phrases with different strengths. For example, the word ”excellent” is stronger than ”good”. The sentiment words, like ”good” and ”bad”, are used to express positive and negative sentiments. But also intensifier and diminisher words can change the degree of the expressed sentiment, an intensifier increases the intensity of a positive or negative word like ”very” and a diminisher decrease its intensity like ”barely”.

However, sentence intensity prediction of short sentences faces several challenges:

1. Due to the nature of the short sentence itself; the limited size of the sentences, the informal language of the content that may contain slang words and non-standard expressions (e.g. LOL instead of laughing out loud, greaaaaaat etc.), and the high level of noise due to the absence of spell checker tools.

2. Due to the sentiment lexicons not including all of the vocabulary needed, or may not be totally balanced between positive and negative sentences.

Our proposal is to:

1. Calculate the probability of positivity for the phrase in the sentiment lexicons, using the point-wise mutual information (PMI) (Cover et al., 1994).

2. When the phrase is not included in the sentiment lexicons, we use the web search engine to find probability of positivity for the phrase based on its co-occurrence near the word ”excellent” and near the word ”poor”.

2 Related work

The sentiment words are the main factor for sentiment classification, by consequence sentiment words and phrases can be used for sentiment classification in an unsupervised method, a method that

[1] http://alt.qcri.org/semeval2016/task7/

Proceedings of SemEval-2016, pages 469–473,
San Diego, California, June 16-17, 2016. ©2016 Association for Computational Linguistics

can solve the problem of domain dependency and reduce the need for annotated training data. The method of Turney (2002) is such a technique. It performs classification based on fixed syntactic patterns that are usually used to express opinions. The syntactic patterns are formed based on part-of-speech (POS) tags, then the sentiment orientation (SO) of the patterns is calculated using the pointwise mutual information (PMI) measure. We renounced the use of syntactic patterns due to the majority of one word phrases in the competition files. Turney and Littman (2003) created two sets of prototypical polar words, one containing positive and another containing negative example words. To compute a new term's polarity, they used the point-wise mutual information (PMI) between that word and each of the prototypical sets (Lin, 1998). The same method was used by Kiritchenko et al. (2014), for the purpose of creating a large scale Twitter sentiment lexicons.

The work of Turney and Littman (2003) is the base of our approach, we use several available sentiment lexicons, and also we use web search engine for each phrase not included in those sentiment lexicons.

3 Difficulties Comparison in this task with languages: English and Arabic

Although the same method is applied for both languages: English and Arabic, the level of difficulty is different when treating both of them. On the resources level, for the English language, we can find many ”free” and ”available on-line” datasets of sentiment lexicons and labeled tweets (e.g. Sentiment140 with 1600k records). For the Arabic language, the resources are limited and our data-set of sentiment lexicons and labeled tweet is of 16K records only. On the language characters level, the Arabic language needed special treatment for the sentiment lexicons files, and for using the web search engines (e.g. using coding: UTF-8). On the language use and diversity level, there are 22 Arabic-speaking countries but the people of these countries speak their own ”mutant-Arabic” languages (dialects), mostly influenced by other languages (e.g. French, English). Also most of the Arabic sentences are pronounced differently (due to accents) in these countries, and then written differently on the social media.

4 Method

Our contribution is an unsupervised method with the use of web search engine as a way to maximize the chances of finding all the slang words, abbreviations, non-standard expressions that a classic corpora will not include.

The method is to calculate the sentiment score for a term w from the sentiment lexicons as shown in the Equation 1 (Kiritchenko et al., 2014):

$$SentSc(w) = PMI(w, pos) - PMI(w, neg) \quad (1)$$

PMI stands for pointwise mutual information, it measures the degree of statistical dependence between two terms. It is used in our work to calculate the degree of statistical dependence between a term and a class (negative or positive).

$$PMI(w, pos) = \log_2 \frac{freq(w, pos) \cdot N}{freq(w) \cdot freq(pos)} \quad (2)$$

Where $freq(w, pos)$ is the number of times a term w occurs as positive or in a positive tweet, $freq(w)$ is the total frequency of term w in sentiment lexicons and labeled tweets, $freq(pos)$ is the total number of positive terms in sentiment lexicons and labeled tweets, and N is the total number of terms in the data-set (Kiritchenko et al., 2014). $PMI(w, negative)$ is calculated similarly.

For the English language, we have done our testing using the below manual constructed sentiment lexicons:

1. Bing Liu Lexicon of Negative and postive words (Hu and Liu, 2004).
2. MPQA Subjectivity Lexicon, it is a Multi-Perspective Question Answering Subjectivity Lexicon (Wilson et al., 2005).

And the below automatic constructed sentiment lexicons:

3. Sentiment140 corpora containing tweets with positive or negative emoticons (Go et al., 2009).
4. NRC Hashtag Sentiment Lexicon (Mohammad and Turney, 2013).

5. SentiWordNet [2] (Baccianella et al., 2010), it is the result of automatically annotating all WordNet synsets according to their degrees of positivity, negativity, and neutrality.
6. Sentiment words from the MPQA word list (Riloff et al., 2003; Wilson et al., 2005). We used the positive, negative words only.
7. And we also use the test file's data of Semeval-2013 Task2 (subtaskA) [3] with positive and negative annotated tweets.

And based on the test results, we used the sentiment lexicons 1,3,4,5 and 7 of the ones previously mentioned, which gave the best results.

For the Arabic language, we are using the below manual constructed sentiment lexicons:

1. Arabic Sentiment Tweets Dataset[4], a set of Arabic tweets containing over 10,000 entries.
2. Twitter data-set for Arabic Sentiment Analysis[5], 1000 positive tweets and 1000 negative ones on various topics such as: politics and arts.
3. LABR Lexicons[6].
4. NRC Hashtag Sentiment Lexicon in many languages (Mohammad and Turney, 2013).

The sentiment lexicons and labeled tweets we are taking as a base for our method do not include all the needed phrases and words. For example: the hashtag phrases (e.g. #live_love_laugh), the phrases with no space between the words (e.g. goodvibes), and in Arabic language the English words written in Arabic characters (e.g. cute written as كيوت).

To solve that issue, we use the web search engines to calculate the probability of using the phrase in a positive context, since the orientation of a phrase is negative if that phrase is more associated with the word "poor" and positive if it is more associated with the word "excellent". For that purpose we apply the Equation 3 for the sentiment orientation (SO) (Turney, 2002):

$$SO(p) = \log_2 \frac{hits(pNEAR\text{"}excellent\text{"}) \cdot hits(\text{"}poor\text{"})}{hits(pNEAR\text{"}poor\text{"}) \cdot hits(\text{"}excellent\text{"})} \tag{3}$$

Where $hits(x)$ is the number of pages returned from a search engine for a query based on the phrase used. For example, $hits('poor')$ represents the number of pages returned for the query 'poor'. When there are a phrase p and 'excellent' (or 'poor') connected by $NEAR$ operator, it is the co-occurrences of the phrase and 'excellent' (or 'poor') in same pages on a specified range of words (we choose the range of 10 words (Turney, 2002)). The SO values of the extracted phrase is considered as its sentiment intensity.

Since the main goal of SemEval2016 Task7 evaluation is the ranking of phrases provided according to their sentiment intensity, we are able to simplify the Equation 3 by removing the part shown in Equation 4, which is constant and will not effect the ranking. The final equation we use is Equation 5.

$$hits(\text{"}poor\text{"})/hits(\text{"}excellent\text{"}) = 0.637 \tag{4}$$

$$SO(p) = \log_2 \frac{hits(pNEAR\text{"}excellent\text{"})}{hits(pNEAR\text{"}poor\text{"})} \tag{5}$$

For the Arabic language, we apply the same concept and equation, but we have to specify the words which once associated to a phrase, they make it more negative or more positive. We have done some tests, on a sample of 40 phrases from development data provided by SemEval-2016 Task7, using the translation of "poor" and "excellent" in Arabic (فقير, ممتاز). We had for many phrases the value of hits(p NEAR ممتاز) and hits(p NEAR فقير) equals to zero. Thus, we decided to choose a group of words that we find most sentimentally expressive:

1. Arabic Positive words:
   رائع جميل أحسن أفضل فرح جيد ذكي
2. Arabic Negative words:
   مخيف قبيح أسوأ غلط حزين سيئ غبي

We first tested our method using Bing Search Engine API[7][8]. But the use of the "near:" operator, to restrict the distance between search phrases, did not work as expected. For example when we search

[2] File:subjclueslen1-HLTEMNLP05.tff (http://www.cs.pitt.edu/mpqa/)

[3] https://www.cs.york.ac.uk/semeval-2013/task2/index.html

[4] http://www.mohamedaly.info/datasets/astd

[5] https://archive.ics.uci.edu

[6] http://www.mohamedaly.info/datasets/labr

[7] http://www.bing.com/toolbox/bingsearchapi

[8] Bing gives till 5,000 Transactions/month, set at 50 results per query, for free

Method	Kendall	Spearman
Google_Search_API	0.287	0.412
PMI_Sentiment_Lexicons	0.207	0.305

Table 1: Sample Eng. Test: Google Search API and Unsupervised PMI Sentiment Lexicons.

for the word ”awesome” near the word ”poor”, at ”near:5” we get 21360 results, and the same search with ”near:10”, although it should give larger number than the previous since we search in a wider range, it returns 21332 results. And we assume it is caused by the use limitation of Bing Search API, which as consequences gives bad results for the sentiment intensity prediction. Once applied on the test data provided by SemEval-2015 Task10 SubtaskE[9], it gives the below results reflecting the lack of correlation:
Kendall rank correlation coefficient: 0.029
Spearman rank correlation coefficient: 0.039

Then we applied our method using Google Search API[10]. We use it to return the number of documents containing the phrase of the query, within ten words of ’excellent’ (or ’poor’) in either order.

And as text prepossessing, we removed the hashtags (#) from the phrases, and we replaced the underscores (_) by spaces.

5 Experiments and Evaluations

We test our ”English language system” using English test data provided by SemEval-2015 Task10 SubtaskE[11] (1315 of general English phrases). And since the use of Google Search API is limited by a number of queries by day, we tested Google Search API by a sample of 40 sentences from the test data file. As shown in Table 1, the Google Search API gives better results than the unsupervised PMI Sentiment Lexicons method alone.

In the Table 2 we have the results of the methods: PMI_Sentiment_Lexicons, PMI_Sentiment_Lexicons + Google_Search_API and Bing_Search_API, using English test data provided by SemEval-2015 Task10 SubtaskE. The best results are for ”PMI_Sentiment_Lexicons + Google_Search_API” although the use of Google Search API is applied on 5% only of the file’s phrases (since those 5% were the only phrases not found in our data-set). And in case of no results returned from Google Search API, the phrase is classified Neutral and the value 0.5 is given as its sentiment intensity.

Method	Kendall	Spr.
PMI_Sentiment_Lexicons	0.443	0.620
PMI_Sent_Lex + Google_Search_API	0.452	0.631
Bing_Search_API	0.029	0.039

Table 2: Eng. Test: Bing Search API, Google Search API and PMI Sentiment Lexicons (Spr. as Spearman).

Method	Kendall	Spr.
PMI_Sentiment_Lexicons	0.417	0.584
PMI_Sent_Lex + Google_Search_API	0.402	0.561

Table 3: Arabic Test: Google Search API and PMI Sentiment Lexicons (Spr. as Spearman).

We test our ”Arabic language system” using Arabic development data provided by SemEval-2016 Task7 (200 of Arabic phrases), where we used Google Search API on 20% of the phrases (since those 20% were not found in our data-set). The results are in Table 3. We can notice that the use of Google Search API did not increase the values and that would be due to our choice in Arabic positive and negative words included in the SO equation.

We apply the method with the higher score on the testing data provided by SemEval2016 task7[12]. A data-set was provided for each sub-task: the first sub-task’s data-set contains 2799 single words and phrases of general English. The second sub-task’s data-set contains 1069 English phrases of mixed polarity words (e.g. lazy Weekend). And the third data-set contains 1166 of single words and phrases commonly found in Arabic tweets. The results of the SemEval2016 Task7 are in Tables 4, 5 and 6. Compared to others teams’ systems, which are based on supervised methods with extremely large training corporas (1.6M) or Deep-learning approaches, we can say that our system give good results for the mixed polarity English sub-task (since it has the second best result). Also in the Arabic phrases sub-task, we have interesting results since we applied the unsupervised PMI Sentiment Lexicons method only.

[9] http://alt.qcri.org/semeval2015/task10/

[10] https://developers.google.com/web-search/docs/

[11] http://alt.qcri.org/semeval2015/task10/

[12] http://alt.qcri.org/semeval2016/task7/

Team	**Kendall**	**Spearman**	**Supervision**
ECNU	0.704	0.863	Supervised
UWB	0.659	0.854	Supervised
LSIS	0.345	0.508	**Unsupervised**

Table 4: SemEval2016 Task7: General English Results.

Team	**Kendall**	**Spearman**	**Supervision**
ECNU	0.523	0.674	Supervised
LSIS	0.422	0.590	**Unsupervised**
UWB	0.414	0.578	Supervised

Table 5: SemEval2016 Task7: Mixed Polarity English Results.

Team	**Kendall**	**Spearman**	**Supervision**
iLab-Edinb.	0.536	0.680	Supervised
NileTMRG	0.475	0.658	Supervised
LSIS	0.424	0.583	**Unsupervised**

Table 6: SemEval2016 Task7: Arabic phrases Results.

6 Conclusion

In this paper, we present our contribution in SemEval2016 task7: Determining Sentiment Intensity of English and Arabic Phrases, where we use web search engines for English and Arabic unsupervised sentiment intensity prediction. Applying our system to the 3 sub-tasks, faced to other teams' systems based on supervised approaches with much higher costs than ours:
- For the General English sub-task, our system have modest but interesting results.
- For the Mixed Polarity English sub-task, our system results achieve the second place.
- For the Arabic phrases sub-task, our system have very interesting results since we applied the unsupervised method only.
Although the results are encouraging, further investigation is required, in both languages, concerning the choice of positive and negative words which once associated to a phrase, they make it more negative or more positive.

References

Stefano Baccianella, Andrea Esuli, and Fabrizio Sebastiani. 2010. SentiWordNet 3.0: An Enhanced Lexical Resource for Sentiment Analysis and Opinion Mining. *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*, 0(January):2200–2204.

T M Cover, J A Thomas, and J Kieffer. 1994. *Elements of information theory*.

Alec Go, Richa Bhayani, and Lei Huang. 2009. Twitter Sentiment Classification using Distant Supervision. *Processing*, 150(12):1–6.

Minqing Hu and Bing Liu. 2004. Mining and summarizing customer reviews. *Proceedings of the 2004 ACM SIGKDD international conference on Knowledge discovery and data mining KDD 04*, 04:168.

Svetlana Kiritchenko, Xiaodan Zhu, and Saif M. Mohammad. 2014. Sentiment analysis of short informal texts. *Journal of Artificial Intelligence Research*, 50:723–762.

Dekang Lin. 1998. Automatic retrieval and clustering of similar words. *ACL '98 Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics*, pages 768–774.

Bing Liu. 2012. Sentiment Analysis and Opinion Mining. *Synthesis Lectures on Human Language Technologies*, 5(1):1–167.

Saif M. Mohammad and Peter D. Turney. 2013. Crowdsourcing a word-emotion association lexicon. *Computational Intelligence*, 29(3):436–465.

Bo Pang and Lillian Lee. 2008. Opinion Mining and Sentiment Analysis. *Foundations and Trends in Information Retrieval*, 2(12):1–135.

Ellen Riloff, Ellen Riloff, Janyce Wiebe, and Janyce Wiebe. 2003. Learning extraction patterns for subjective expressions. *Proceedings of the 2003 conference on Empirical methods in natural language processing*, pages 105–112.

Peter D Turney. 2002. Thumbs up or thumbs down? Semantic Orientation applied to Unsupervised Classification of Reviews. *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL)*, (July):417–424.

Theresa Wilson, Janyce Wiebe, and Paul Hoffmann. 2005. Recognizing contextual polarity in phrase-level sentiment analysis. In *HLT '05: Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, pages 347–354, Morristown, NJ, USA. Association for Computational Linguistics.