Synalp-Empathic: A Valence Shifting Hybrid System for Sentiment Analysis

Alexandre Denis, Samuel Cruz-Lara, Nadia Bellalem and Lotfi Bellalem
LORIA/University of Lorraine
Nancy, France
{alexandre.denis, samuel.cruz-lara, nadia.bellalem, lotfi.bellalem}@loria.fr

Abstract

This paper describes the *Synalp-Empathic* system that competed in SemEval-2014 Task 9B Sentiment Analysis in Twitter. Our system combines syntactic-based valence shifting rules with a supervised learning algorithm (Sequential Minimal Optimization). We present the system, its features and evaluate their impact. We show that both the valence shifting mechanism and the supervised model enable to reach good results.

1 Introduction

Sentiment Analysis (SA) is the determination of the polarity of a piece of text (positive, negative, neutral). It is not an easy task, as proven by the moderate agreement between human annotators when facing this task. Their agreement varies whether considering document or sentence level sentiment analysis, and different domains may show different agreements as well (Bermingham and Smeaton, 2009).

As difficult the task is for human beings, it is even more difficult for machines which face syntactic, semantic or pragmatic difficulties. Consider for instance irrealis phenomena such as "*if this is good*" or "*it would be good if*" that are both neutral. Irrealis is also present in questions ("*is this good?*") but presupposition of existence does matter: "*can you fix this terrible printer?*" would be polarized while "*can you give me a good advice?*" would not. Negation and irrealis interact as well, compare for instance "*this could be good*" (neutral or slightly positive) and "*this could not be good*" (clearly negative). Other difficult phenomena include semantic or pragmatic effects, such as point of view ("*Israel failed to defeat Hezbollah*", negative for Israel, positive for Hezbollah), background knowledge ("*this car uses a lot of gas*"), semantic polysemy ("*this vacuum cleaner sucks*" vs "*this movie sucks*"), etc.

From the start, machine learning has been the widely dominant approach to sentiment analysis since it tries to capture these phenomena alltogether (Liu, 2012). Starting from simple n-grams (Pang et al., 2002), more recent approaches tend to include syntactic contexts (Socher et al., 2011). However these supervised approaches all require a training corpus. Unsupervised approaches such as the seminal paper of (Turney, 2002) require training corpus as well but do not require annotations. We propose in this paper to look first at approaches that *do not* require any corpus because annotating a corpus is in general costly, especially in sentiment analysis in which several annotators are required to maintain a high level of agreement[1]. Nevertheless supervised machine learning can be useful to *adapt* the system to a particular domain and we will consider it as well.

Hence, we propose in this paper to first consider a domain independent sentiment analysis tool that does not require any training corpus (section 2). Once the performance of this tool is assessed (section 2.4) we propose to consider how the system can be extended with machine learning in section 3. We show the results on the SemEval 2013 and 2014 corpora in section 4.

2 Sentiment Analysis without Corpus

We present here a system that does sentiment analysis without requiring a training corpus. We do so in three steps: we first present a raw lexical baseline that naively considers average valence taking the prior valence of words from polarity lexicons.

[1] as done in SemEval2013 SA task (Nakov et al., 2013)

Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014), pages 605–609,
Dublin, Ireland, August 23-24, 2014.

We then show how to adapt this baseline to the Twitter domain. Finally, we describe a method wich takes into account the syntactic context of polarized words. All methods and strategies are then evaluated.

2.1 Raw Lexical Baseline

The raw lexical baseline is a simple system that only relies on polarity lexicons and takes the average valence of all the words. The valence is modeled using a continuous value in $[0, 1]$, 0.5 being neutral. The algorithm is as follows:

1. perform part of speech tagging of the input text using the Stanford CoreNLP tool suite,
2. for all words in the input text, retrieve their polarity from the lexicons using lemma and part of speech information. If the word is found in several lexicons, return the average of the found polarities. Otherwise if the word is not found, return 0.5.
3. then for the tweet, simply compute the average valence among all words.

We tried several lexicons but ended with focusing on the Liu's lexicon (Hu and Liu, 2004) which proved to offer the best results. However Liu's lexicon is missing slang or bad words. We therefore extended the lexicon using the *onlineslangdictionary.com* website which provides a list of slang words expressing either positive or negative properties. We extracted around 100 words from this lexicon which we call *urban lexicon*.

2.2 Twitter Adaptations

From this lexical base we considered several small improvements to adapt to the Twitter material. We first observed that the Stanford part of speech tagger had a tendency to mistag the first position in the sentence as proper noun. Since in tweets this position is often in fact a common noun, we systematically retagged these words as common nouns. Second, we used a set of 150 hand written rules designed to handle chat colloquialism i.e., abbreviations ("*wtf*" → "*what the f****", twitter specific expressions ("*mistweet*" → "*regretted tweet*"), missing apostrophe ("*isnt*" → "*isn't*"), and smileys. Third, we applied hashtag splitting (e.g. "*#ihatemondays*" → "*i hate mondays*"). Finally we refined the lexicon lookup strategy to handle discrepancies between lexicon and part of speech tagger. For instance, while the part of speech tagger may tag *stabbed* as an adjective with lemma *stabbed*, the lexicon might list it as a verb with lemma *stab*. To improve robustness we therefore look first for the inflected form then for the lemma.

2.3 Syntactic Enhancements

Valence Shifting Valence shifting refers to the differential between the *prior polarity* of a word (polarity from lexicons) and its contextual polarity (Polanyi and Zaenen, 2006). Following (Moilanen and Pulman, 2007), we apply polarity rewriting rules over the parsing structure. However we differ from them in that we consider dependency rather than phrase structure trees.

The algorithm is as follows:

1. perform dependency parsing of the text (with Stanford CoreNLP)
2. annotate each word with its prior polarity as found in polarity lexicons
3. rewrite prior polarities using dependency matching, hand-crafted rules
4. return the root polarity

Table 1 shows example rules. Each rule is composed of a matching part and a rewriting part. Both parts have the form (N, L_G, P_G, L_D, P_D) where N is the dependency name, L_G and L_D are respectively the lemmas of the governor and dependent words, P_G and P_D are the polarity of the governor and dependent words. We write the rules in short form by prefixing them with the name of the dependency and either the lemma or the polarity for the arguments, e.g. $N(P_G, P_D)$. For instance, the *inversion* rule "neg(P_G, P_D) → neg(!P_G, P_D)" inverts the polarity of the governor P_G for dependencies named *neg*. One important rule is the *propagation* rule "N(0.5, P_D) → N(P_D,P_D)" which propagates the polarity of the dependent word P_D to the governor only if it is neutral. Another useful rule is the *overwrite* rule "amod(1,0) → amod(0,0)" which erases for *amod* dependencies, the positive polarity of the governor given a negative modifier.

The main algorithm for rule application consists in testing all rules (in a fixed order) on all dependencies iteratively. Whenever a rule fires, the whole set of rules is tested again. Potential looping

Rule	Example
neg(P_G, P_D) → neg(!P_G, P_D)	he's not happy
det(P_G, "no") → det(!P_G,"no")	there is no hate
amod(1,0) → amod(0,0)	a missed opportunity
nsubj(0,1) → nsubj(0,0)	my dreams are crushed
nsubj(1,0) → nsubj(1,1)	my problem is fixed
N(0.5, P_D) → N(P_D,P_D)	(propagation)

Table 1: Excerpt of valence shifting rules.

is prevented because (i) the dependency graph returned by the Stanford Parser is a directed acyclic graph (de Marneffe and Manning, 2008) and (ii) the same rule cannot apply twice to the same dependency.

For instance, in the sentence "*I do not think it is a missed opportunity*", the verb "*missed*" has negative polarity and the noun "*opportunity*" has positive polarity. The graph in Figure 1 shows different rules application: first the *overwrite* rule (1.) changes the positive polarity of "*opportunity*" to a negative polarity which is then transferred to the main verb "*think*" thanks to the *propagation* rule (2.). Finally, the *inversion* rule (3.) inverts the negative polarity of *think*. As a result, the polarity of the sentence is positive.

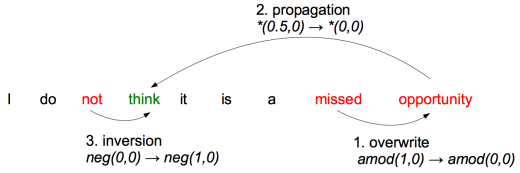

Figure 1: Rules application example.

Various Phenomena Several other phenomena need to be taken into account when considering the co-text. Because of irrealis phenomena mentioned in the introduction, we completely ignored questions. We also ignored proper nouns (such as in "*u need 2 c the documentary* The Devil Inside") which were a frequent source of errors. These two phenomena are labeled *Ignoring forms* in Table 2. Finally since our approach is sentence-based we need to consider valence of tweets with several sentences and we simply considered the average.

2.4 Results on SemEval2013

We measure the performance of the different strategies on the 3270 tweets that we downloaded from the SemEval 2013 Task 2 (Nakov et al., 2013) test corpus[2]. The used metrics is the same than SemEval 2013 one, an unweighted average between positive and negative F-score.

System	F-score	Gain
Raw lexical baseline	54.75	
+ Part of speech fix	55.00	+0.25
+ Colloquialism rewriting	57.66	**+2.66**
+ Hashtag splitting	57.80	+0.14
+ Lexicon fetch strategy	58.25	+0.45
+ Valence shifting	62.37	**+4.12**
+ Ignoring forms	62.97	+0.60

Table 2: Results of syntactic system.

As shown in Table 2, the raw lexical baseline starts at 54.75% F-score. The two best improvements are *Colloquialism rewriting* ($+2.66$) that seems to capture useful polarized elements and *Valence shifting* ($+4.12$) which provides an accurate account for shifting phenomena. Overall other strategies taken separately do not contribute much but enable to have an accumulated $+1.44$ gain of F-score. The final result is 62.97%, and we will refer to this first system as the *Syntactic system*.

3 Machine Learning Optimization

The best F-score attained with the syntactic system (62.97%) is still below the best system that participated in SemEval2013 (69.02%)[3]. To improve performance, we input the valence computed by the syntactic system as a feature in a supervised machine learning (ML) algorithm. While there exists other methods such as (Choi and Cardie, 2008) which incorporates syntax at the heart in the machine algorithm, this approach has the advantage to be very simple and independent of any specific ML algorithm. We chose the Sequential Minimal Optimization (SMO) which is an optimization of Support Vector Machine (Platt, 1999) since it was shown (Balahur and Turchi, 2012) to have good results that we observed ourselves.

In addition to the valence output by our syntactic system, we considered the following additional low level features:

- *1-grams words*: we observed lower results with n-grams ($n > 1$) and decided to keep 1-grams only. The words were lemmatized and no tf-idf weighting was applied since it showed lower results.

- *polarity counts*: it is interesting to include low level polarity counts in case the

[2]Because of Twitter policy the test corpus is not distributed by organizers but tweets must be downloaded using their identifiers, resulting in discrepancies from the official campaign (3814 tweets).

[3]Evaluated on full 3814 tweets corpus

syntactic system does not correctly capture valence shifts. We thus included independent features counting the number of positive/negative/neutral words according to several lexicons: Liu's lexicon (Hu and Liu, 2004), our *urban* lexicon, SentiWordnet (Baccianella et al., 2010), QWordnet (Agerri and Garca-Serrano, 2010) and MPQA lexicon (Wilson et al., 2005).

- *punctuation count*: exclamation and interrogation marks are important, so we have an independent feature counting occurrences of "?", "!", "?!", "!?".

Thanks to the ML approach, we can obtain for a given tweet the different probabilities for each class. We were then able to adapt each probabilities to favor the SemEval metrics by weighting the probabilities thanks to the SemEval 2013 training and development corpus using 10-fold cross validation (the weights were trained on 90% and evaluated on 10%). The resulting weights reduce the probability to assign the neutral class to a given tweet while raising the positive/negative probabilities. This optimization is called *metrics weighting* in Table 3.

4 Optimization Results

We describe here the results of integrating the syntactic system as a feature of the SMO along with other low level features. The SemEval 2014 gold test corpus was not available at the time of this writing hence we detail the features only on the SemEval 2013 gold test corpus.

4.1 On SemEval 2013

The results displayed in Table 3 are obtained with the SMO classifier trained using the WEKA library (Hall et al., 2009) on our downloaded SemEval 2013 development and training corpora (7595 tweets). As before, the given score is the average F-score computed on the SemEval 2013 test corpus. Note that the gain of each feature must be interpreted in the context of other features (e.g. *Polarity counts* needs to be understood as *Words+Polarity Counts*).

The syntactic system feature, that is considering only one training feature which is the valence annotated by the syntactic system, starts very low (33.69%) since it appears to systematically favor positive and neutral classes. However adding the 1-gram lemmatized words raises the result to 63.03%, slightly above the syntactic system alone (62.97%). Considering polarity counts raises the F-score to 65.02% showing that the syntactic system does not capture correctly all valence shifts (or valence neutralizations). Considering an independent feature for punctuation slightly raises the result. Metrics weighting, while not being a training feature per se, provides an important boost for the final F-score (67.83%).

Features	F-score	Gain
Syntactic system	33.69	
+ Words	63.03	+29.34
+ Polarity counts	65.02	+1.99
+ Punctuation	65.65	+0.63
+ Metrics weighting	67.83	+2.18

Table 3: Detailed results on SemEval 2013.

4.2 On SemEval 2014

We participated to SemEval 2014 task B as the *Synalp-Empathic* team (Rosenthal et al., 2014). The results are 67.43% on the Twitter 2014 dataset, 3.53 points below the best system. Interestingly the score obtained on Twitter 2014 is very close to the score we computed ourselves on Twitter 2013 (67.83%) suggesting no overfitting to our training corpus. However, we observed a big drop in the Twitter 2013 evaluation as carried out by organizers (63.65%), we assume that the difference in results could be explained by difference in datasets coverage caused by Twitter policy.

5 Discussion and Conclusion

We presented a two-steps approach for sentiment analysis on Twitter. We first developed a lexico-syntactic approach that does not require any training corpus and enables to reach 62.97% on SemEval 2013. We then showed how to adapt the approach given a training corpus which enables reaching 67.43% on SemEval 2014, 3.53 points below the best system. We further showed that the approach is not sensitive to overfitting since it proved to be as efficient on the SemEval 2013 and the SemEval 2014 test corpus. In order to improve the performance, it could be possible adapt the lexicons to the specific Twitter domain (Demiroz et al., 2012). It may also be possible to investigate how to learn automatically the valence shifting rules, for instance with Monte Carlo methods.

Acknowledgements

This work was conducted in the context of the ITEA2 1105 Empathic Products *project, and is supported by funding from the French Services, Industry and Competitivity General Direction. We would like to thank Christophe Cerisara for the insights regarding the machine learning system and Claire Gardent for her advices regarding the readability of the paper.*

References

Rodrigo Agerri and Ana Garca-Serrano. 2010. Q-wordnet: Extracting polarity from wordnet senses. In *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*, Valletta, Malta, may. European Language Resources Association (ELRA).

Stefano Baccianella, Andrea Esuli, and Fabrizio Sebastiani. 2010. Sentiwordnet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining. In *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*, Valletta, Malta, may. European Language Resources Association (ELRA).

Alexandra Balahur and Marco Turchi. 2012. Multilingual sentiment analysis using machine translation? In *Proceedings of the 3rd Workshop in Computational Approaches to Subjectivity and Sentiment Analysis*, WASSA '12, pages 52–60, Stroudsburg, PA, USA.

Adam Bermingham and Alan F. Smeaton. 2009. A study of inter-annotator agreement for opinion retrieval. In James Allan, Javed A. Aslam, Mark Sanderson, ChengXiang Zhai, and Justin Zobel, editors, *SIGIR*, pages 784–785. ACM.

Yejin Choi and Claire Cardie. 2008. Learning with compositional semantics as structural inference for subsentential sentiment analysis. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '08, pages 793–801, Stroudsburg, PA, USA.

Marie-Catherine de Marneffe and Christopher D. Manning. 2008. Stanford typed dependencies manual. Technical report.

Gulsen Demiroz, Berrin Yanikoglu, Dilek Tapucu, and Yücel Saygin. 2012. Learning domain-specific polarity lexicons. In *Proceedings of the 12th International Conference of Data Mining Workshops (ICDMW)*, pages 674–679, Dec.

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 10th International Conference on Knowledge Discovery and Data Mining*, pages 168–177.

Bing Liu, 2012. *Sentiment Analysis and Opinion Mining*. Morgan & Claypool Publishers, May.

Karo Moilanen and Stephen Pulman. 2007. Sentiment composition. In *Proceedings of Recent Advances in Natural Language Processing (RANLP 2007)*, pages 378–382, September 27-29.

Preslav Nakov, Sara Rosenthal, Zornitsa Kozareva, Veselin Stoyanov, Alan Ritter, and Theresa Wilson. 2013. SemEval-2013 task 2: sentiment analysis in twitter. In *Proceedings of the 7th International Workshop on Semantic Evaluation*.

Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. 2002. Thumbs up? Sentiment classication using machine learning techniques. In *Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing*, pages 79–86, Philadelphia, PA.

John C. Platt. 1999. Fast training of support vector machines using sequential minimal optimization. In *Advances in Kernel Methods*, pages 185–208. MIT Press, Cambridge, MA, USA.

Livia Polanyi and Annie Zaenen. 2006. Contextual valence shifters. In JamesG. Shanahan, Yan Qu, and Janyce Wiebe, editors, *Computing Attitude and Affect in Text: Theory and Applications*, volume 20 of *The Information Retrieval Series*, pages 1–10. Springer Netherlands.

Sara Rosenthal, Preslav Nakov, Alan Ritter, and Veselin Stoyanov. 2014. SemEval-2014 task 9: Sentiment analysis in twitter. In *Proceedings of the 8th International Workshop on Semantic Evaluation*.

Richard Socher, Jeffrey Pennington, Eric H. Huang, Andrew Y. Ng, and Christopher D. Manning. 2011. Semi-supervised recursive autoencoders for predicting sentiment distributions. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 151–161, Edinburgh.

Peter Turney. 2002. Thumbs up or thumbs down? semantic orientation applied to unsupervised classication of reviews. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 417–424, Philadelphia, PA.

Theresa Wilson, Janyce Wiebe, and Paul Hoffmann. 2005. Recognizing Contextual Polarity in Phrase-Level Sentiment Analysis. In *Proceedings of Human Language Technologies Conference/Conference on Empirical Methods in Natural Language Processing (HLT/EMNLP 2005)*, Vancouver, CA.