Synalp-Empathic: A Valence Shifting Hybrid System for Sentiment Analysis
Alexandre Denis, Samuel Cruz-Lara, Nadia Bellalem, Lotfi Bellalem

To cite this version:

HAL Id: hal-01099671
https://hal.science/hal-01099671
Submitted on 22 Jan 2015

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L’archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d’enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.
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 (Bermann 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 altogether (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 agreement1. 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)
We then show how to adapt this baseline to the Twitter domain. Finally, we describe a method which 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 onlinelangdictionary.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*ck”, twitter specific expressions (“mistweet” → ”regretted tweet”), missing apostrophe (“isn’t” → ”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
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.](image)

**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 “a 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.

<table>
<thead>
<tr>
<th>Rule</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>neg(P2; P_D) → neg(P2; P_D)</td>
<td>he’s not happy</td>
</tr>
<tr>
<td>det(P2, “no”) → det(P2, “no”)</td>
<td>there is no hate</td>
</tr>
<tr>
<td>amod(1,0) → amod(0,0)</td>
<td>a missed opportunity</td>
</tr>
<tr>
<td>nsubj(1,0) → nsubj(0,0)</td>
<td>my dreams are crushed</td>
</tr>
<tr>
<td>nsubj(1,0) → nsubj(1,1)</td>
<td>my problem is fixed</td>
</tr>
<tr>
<td>N(0.5, P_D) → N(P_D, P_D)</td>
<td>(propagation)</td>
</tr>
</tbody>
</table>

Table 1: Excerpt of valence shifting rules.

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

Table 2: Results of syntactic system.

<table>
<thead>
<tr>
<th>System</th>
<th>F-score</th>
<th>Gain</th>
</tr>
</thead>
<tbody>
<tr>
<td>Raw lexical baseline</td>
<td>54.75</td>
<td>0.25</td>
</tr>
<tr>
<td>+ Part of speech fix</td>
<td>55.00</td>
<td>+0.25</td>
</tr>
<tr>
<td>+ Colloquialism rewriting</td>
<td>57.66</td>
<td>+2.66</td>
</tr>
<tr>
<td>+ Hashtag splitting</td>
<td>57.80</td>
<td>+0.14</td>
</tr>
<tr>
<td>+ Lexicon fetch strategy</td>
<td>58.25</td>
<td>+0.45</td>
</tr>
<tr>
<td>+ Valence shifting</td>
<td>62.37</td>
<td>+4.12</td>
</tr>
<tr>
<td>+ Ignoring forms</td>
<td>62.97</td>
<td>+0.60</td>
</tr>
</tbody>
</table>

³Evaluated 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 Garcia-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

<table>
<thead>
<tr>
<th>Features</th>
<th>F-score</th>
<th>Gain</th>
</tr>
</thead>
<tbody>
<tr>
<td>Syntactic system</td>
<td>33.69</td>
<td></td>
</tr>
<tr>
<td>+ Words</td>
<td>63.03</td>
<td>+29.34</td>
</tr>
<tr>
<td>+ Polarity counts</td>
<td>65.02</td>
<td>+1.99</td>
</tr>
<tr>
<td>+ Punctuation</td>
<td>65.65</td>
<td>+0.63</td>
</tr>
<tr>
<td>+ Metrics weighting</td>
<td>67.83</td>
<td>+2.18</td>
</tr>
</tbody>
</table>

Table 3: Detailed results on SemEval 2013.

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%).

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


