teragram:
Rule-based detection of sentiment phrases using SAS Sentiment Analysis

Hilke Reckman, Cheyanne Baird, Jean Crawford, Richard Crowell,
Linnea Micciulla, Saratendu Sethi, and Fruzsina Veress
SAS Institute
10 Fawcett Street
Cambridge, MA 02138, USA
hilke.reckman@sas.com

Abstract

For SemEval-2013 Task 2, A and B (Sentiment Analysis in Twitter), we use a rule-based pattern matching system that is based on an existing 'Domain Independent' sentiment taxonomy for English, essentially a highly phrasal sentiment lexicon. We have made some modifications to our set of rules, based on what we found in the annotated training data that was made available for the task. The resulting system scores competitively, especially on task B.

1 Introduction

SAS taxonomies for sentiment analysis are primarily topic-focused. They are designed to track sentiment around brands, entities, or other topics and subtopics in a domain (Lange and Sethi, 2011; Lakkaraju and Sethi, 2012; Albright and Lakkaraju, 2011). Domain-independent taxonomies have a second function. In addition to performing topic-focused tasks, they can be set up to perform sentiment analysis at the document level, classifying the whole document as positive, negative, or neutral. In this task all sentiment expressions are taken into account, rather than only those which are related to the tracked topic. This second function is becoming increasingly important. It allows for a broader perspective that is complementary to topic-focused opinion mining.

We participated in both subtask A and B of SemEval-2013 Task 2: Sentiment Analysis in Twitter (Wilson et al., 2013) with an adaptation of our existing system. For task B, identifying the overall sentiment of a tweet, our taxonomy mainly needed some fine-tuning to specifically accommodate Twitter data. (Normally tweets only make up a small part of the data we work with.) We also made a few adaptations to focus entirely on document level sentiment, whereas originally the main focus of our system was on tracking sentiment around products. For task A, identifying the sentiment of ambiguous phrases in a tweet, a few more modifications were needed.

Our system is entirely rule-based, and the rules are hand-written. In some cases, statistical text mining approaches are used for the discovery of topics and terms to facilitate rule writing. Our sentiment analysis software does offer a statistical component, but our experience is that purely rule-based models work better for our typical sentiment analysis tasks.

Advantages of rules are that problems observed in the output can be targeted directly, and the model can become more and more refined over time. Also, they allow for simple customization. In our brand-centered work, we customize our taxonomies for one or more brands that we want to track. When we build a taxonomy for a new domain, we build upon work we have done before in other domains. The assignment of sentiment to certain phrases can be sensitive to context where it needs to be. The canceled task C, identifying sentiment related to a topic, could have been approached successfully with a rule-based approach, as our rules are specifically designed to connect sentiment to targeted topics.

Section 2 describes the basic architecture of our system, followed by a section on related work. Then sections 4 and 5 describe the adaptations made for

*Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, pages 513–519, Atlanta, Georgia, June 14-15, 2013.

each subtask and present the results. This is followed by a more general discussion of our approach in the light of these results in section 6, and the conclusion in section 7.

2 The base system

The datasets we normally use for the development of our taxonomies include blogs, forums, news, and Twitter. When developing a domain-specific taxonomy, we collect data for that particular domain, e.g. Banking, Retail, Hospitality. We build the taxonomy with the terms we encounter in those documents, and test on a new set of documents. The Domain Independent taxonomy started out as the common base derived from several of these taxonomies, and was then built out and tested using a wider range of English-language documents. Since we used some other tweets in the development of the original system, our submission is considered unconstrained.

Our rules are patterns that match words or sequences of words, which makes our approach essentially lexicon-based. Matching occurs left-to-right and longer matches take precedence over shorter ones. The top level rules in our sentiment taxonomy are set up to recognize positive and negative word-sequences. There is also a set of 'neutral' rules at that level that block the assignment of positive or negative sentiment in certain cases.

A positive or negative sequence can consist of a single word from the positive or negative word-lists, or a spelled out phrase from the positive or negative phrase-lists. Alternatively, it can be built up out of multiple components, for example an emphatic modifier and a sentiment term, or a negation and a sentiment term. We call these sequences Positive and Negative 'Contexts', since they are contexts for the topic-terms that we normally track.

Documents are preprocessed by an in-house POS-tagger. Rules can require a word to have a particular part of speech.

The words in the word-list, or in any of the other rules, can be marked with an '@'-sign to enable morphological expansion, and in that case they will match any of the forms in their paradigm. For example 'love@' will match *love*, *loves*, *loved*, and *loving*. This functionality is supported by a morphological dictionary that links these forms to their stem.

The rules are organized into lists that represent useful concepts, which can be referred to in other rules as a means of abstraction. For example the rule:

_def{Negation} _def{PositiveAdjectives}

matches phrases that are composed of a negation (as defined in the list named *Negation*) and a positive adjective (as defined in the list named *PositiveAdjectives*). *Negation* includes rules like 'hasn't been', 'doesnt'[sic], 'not exactly the most', etc., and *PositiveAdjectives* contains a rule that matches words in *PositiveWords* if they are also tagged as adjectives. For efficiency reasons the dependencies cannot be circular, hence not allowing for recursion.

Distance rules can be used to capture a longer span, matching a specified pattern at the beginning and at the end, including arbitrary intervening words up to a specified number. They can also be used to make matching a term dependent on specified terms in the context. For example,

(SENT, (DIST_4, "_a{_def{HigherIsBetter}}", "_a{_def{Lowering}}"))

will capture phrases that say a company's profit (*HigherIsBetter*) went down (*Lowering*). The SENT-operator prevents matching across sentence boundaries.

(ORDDIST_7, "_def{PositiveContext}", "_a{_def{PositiveAmbig}}")

will capture ambiguous positive expressions when they follow an unambiguously positive sequence within a distance of 7 words.

This ensemble of lists and rules has grown relatively organically, and is motivated by the data we encounter. We introduce new distinctions when we feel it will make a difference in terms of results, or sometimes for ease of development and maintenance.

Usually each sentiment expression has the same weight, and one positive and one negative expression cancel each other out. However at the top level we can introduce weights, and we have done so in this model. We have created lists of weak positive and negative expressions, and we gave those very

Positive:

- (ORDDIST_2, "_a{exceed@}", "_a{expectation@}")
- :Pro could not be happier
- blown away by
- _def{Negation} want@ it to end
- above and beyond
- break@ down barriers
- can't go wrong with
- dying to _def{Consume}
- save@ me _def{Money}
- (ALIGNED, "_c{treat@}", ":N")

Negative:

- _def{Negation} find _def{NounPhrases} _def{PositivePhrases}
- (SENT, (ORDDIST_7, "_a{disappointed that}", "_a{_def{PositivePhrases}}"))
- I would have loved
- _def{Negation} accept@
- breach of _def{PositiveWords}
- _def{Money} magically disappears
- lack of training
- make@ no sense
- subject@ me to
- fun dealing with

Figure 1: Examples of rules for positive and negative phrases and patterns.

low weights, so that they would only matter if there were no regular-strength expressions present. We limited some of those weak sentiment rules to subtask A only, but they clearly helped with recall there.

Negations in the default case turn positives into negatives and negatives into neutrals. In addition to negations we also have sentiment reversers, which turn negatives into positives. Simple negations normally scope over a right-adjacent word or phrase, for example a noun phrase or a verb. A special class of clausal negations (*I don't think that*) by approximation take scope over a clause.

This system contains roughly 2500 positive words and 2000 positive phrases, and roughly 7500 negative words and 3000 negative phrases. Some examples are given in Figure 1. The neutral list also contains about 2000 rules. Other helper lists such as *Negation*, *EmphaticModifiers*, and *Money* typically contain about a hundred rules each.

A system like this takes about six to eight weeks to build for a new language. This requires a developer who is already familiar with the methodology, and assumes existing support for the language, including a morphological dictionary and a part-of-speech tagger.

3 Related work

In tasks that are not topic-related, purely rule-based models are rare, although the winning system of SemEval-2010 Task 18 (Wu and Jin, 2010), somewhat similar to task A, was rule-based (Yang and Liu, 2010). Liu (2010) suggests that more rule-based work may be called for. However, there are many other systems with a substantial rule-based component (Nasukawa and Yi, 2003; Choi and Cardie, 2008; Prabowo and Thelwall, 2009; Wilson et al., 2005). Systems commonly have some rules in place that account for the effect of negation (Wiegand et al., 2010) and modifiers. Sentiment lexicons are widely used, but mainly contain single words (Baccianella et al., 2010; Taboada et al., 2011). For topic-related tasks, rule-based systems are a bit more common (Ding et al., 2008).

4 Task A

Task A was to assign sentiment to a target in context. The target in isolation would often be ambiguous. It was a novel challenge to adapt our model for this subtask.

Since we normally track sentiment around specific topics, we can usually afford to ignore highly ambiguous phrases. Typical examples of this are ambiguous emoticons and comments like *no joke* at the end a sentence, or directly following it. When these are used and could be disambiguated, usually there is a less ambiguous term available that occurs closer to the topic-term that we are interested in. (In some cases we do use the topic as disambiguating context.)

Also, we generally place slightly more emphasis on precision than on recall, assuming that with enough data the important trends will emerge, even if we ignore some of the unclear cases and outliers. This makes the output cleaner and more pleasant to

work with for follow-up analysis.

4.1 Model adaptations and processing

We adapted our model to task A by introducing lists of ambiguous positive and negative terms that were then disambiguated in context, e.g. if there was another sentiment term of a specified polarity nearby. We also added some larger patterns that included an ambiguous term, but as a whole had a much clearer polarity. Below are some examples of rules for the word *like*, which is highly ambiguous in English.

1. (ALIGNED, "_c{like@}", ":V") (pos)
2. likes (pos)
3. I like (pos)
4. like magic (pos)
5. give it a "like" (pos)
6. kinda like it (weakpos)
7. doesn't seem like (hypothetical)
8. How can you like (neg)
9. don't like (neg)
10. like to pretend (neg)
11. treated like a number (neg)
12. Is it like (neutral)
13. a bit like (neutral)
14. the likes of (neutral)

A seemingly obvious rule for *like* is (1), restricting it to usage as a verb. However, disambiguating *like* is a difficult task for the tagger too, and the result is not always correct. Therefore this rule is a fall-back case, when none of the longer rules apply. Inflected forms such as (2) are pretty safe, with a few exceptions, which can be caught by neutralizing rules, such as (14). The hypothetical case, (7), is not used in task A, but it is in task B.

A potential issue for our results on this task is that our system only returns the longest match. So in a sentence such as '*I didn't like it*', if you ask people to annotate *like*, they may say it is positive, whereas the longer phrase *didn't like* is negative. In the output of our system, *like* will only be part of a negative sequence. The information that it was originally recognized as a positive word cannot be retrieved at the output level.

We found that the annotators for task A were in general much more liberal in assigning sentiment than we normally are. We made major gains by removing some of our neutralizing rules, for example those that neutralize sentiment in hypothetical contexts, and by classifying negations that were not part of a larger recognized phrase as weak negatives.

The annotations in the development data were sometimes confusing (see also section 6). We had some difficulty in figuring out when certain terms such as *hope* or *miss you* should be considered positive and when negative. The verb *apologize* turned out to be annotated sometimes positive and sometimes negative in near identical tweets.

The test items were processed as follows:

1. run the sentiment model on the text (tweet/SMS)
2. identify the target phrase as a character span
3. collect detected sentiment that overlaps with the target phrase
   (a) if there is no overlapping sentiment expression, the sentiment is neutral
   (b) if there is exactly one overlapping sentiment expression, that expression determines the sentiment
   (c) if there is more than one sentiment expression that overlaps with the target, compute which sentiment has more weight (and in case of a draw, assign neutral)

4.2 Results

We get a higher precision for positive and negative sentiment on task A than any of the other teams, but we generally under-predict sentiment. Precision on neutral sentiment is very low. Detecting neutral phrases did not seem to be a very important goal in the final version of this task, though. The results of our predictions on the Twitter portion of the data are shown in Figure 2.

These results are slightly different from what we submitted, as we did not realize at the time of submission that the encoding of the text was different in the test data than it had been in the previously released data. The submitted results are included in the summarizing Table 1 at the end of the discussion section.

Some targets are easily missed. We do not have a good coverage of hashtags yet, for example. We incorporate frequent misspellings that are common in Twitter and SMS. However, we have no general strategy in place to systematically recognize unconventionally spelled words (Eisenstein, 2013). For

gs \ pred	positive	negative	neutral	
positive	1821	77	888	2734
negative	47	1091	403	1541
neutral	11	6	143	160
	1879	990	1382	4435

class	precision	recall	f-score
positive	0.9691	0.6661	0.7895
negative	0.9293	0.7080	0.8037
neutral	0.1035	0.8938	0.1855
average(pos and neg)			**0.7966**

Figure 2: Confusion table and scores on task **A**, tweets

a project that processes Twitter data it would also make sense to periodically scan for new hashtags and add them to the rules if they carry sentiment. However, a sentiment lexicon is never quite complete.

Therefore we experimented with a guessing component. If we do not detect any sentiment in the target sequence, we let our model make a guess, based on the overall sentiment it assigns to the document, assuming that an ambiguous target overall is more likely to be positive in a positive context and negative in a negative context. (Note that this is different from our disambiguation rules, which only apply to explicitly listed items.) This gives us substantial gains on this subtask (Figure 3). However, this may not hold up in a similar task where there are more neutral instances than there were here, as we see a decrease in precision on positive and negative.

gs \ pred	positive	negative	neutral	
positive	2147	230	357	2734
negative	137	1249	155	1541
neutral	50	33	77	160
	2334	1512	589	4435

class	precision	recall	f-score
positive	0.9199	0.7853	0.8473
negative	0.8261	0.8105	0.8182
neutral	0.1307	0.4813	0.2056
average(pos and neg)			**0.8327**

Figure 3: Confusion table and scores on task **A**, tweets, with guessing

5 Task B

Task B was to predict the overall sentiment of a tweet. This was much closer to the task our taxonomy is designed for, and yet it turned out to be different in subtle ways.

5.1 Model adaptations and processing

We quickly found that running the model as we had adapted it for subtask A over-predicted sentiment on subtask B. We therefore put most of our neutralizing rules back in place for this subtask, and restricted a subset of the weak sentiment terms to subtask A only. We disabled the mechanism that helped us catch ambiguous terms in subtask A (see section 4.1).

For processing we used our standard method, comparing the added weights of the positive and of the negative sequences found. The highest score wins. In case of a draw, the document is classified as neutral. 'Unclassified' (no sentiment terms found) also maps to neutral for this task. A confidence score is computed, but not used here.

5.2 Results

Our system compares positively to those of the other teams. Originally we were in 3rd place as a team on the Twitter data. After correcting for the encoding problem we rise to second (assuming the other teams did not have the same problem). Among unconstrained systems only, we are first on tweets and second on SMS. The results, after the correction, are shown in Figure 4. As for task A, the original results are included in the final summarizing Table 1.

gs \ pred	positive	negative	neutral	
positive	1188	88	296	1572
negative	66	373	162	601
neutral	408	202	1030	1640
	1662	663	1488	3813

class	precision	recall	f-score
positive	0.7148	0.7557	0.7347
negative	0.5626	0.6206	0.5902
neutral	0.6922	0.6280	0.6586
average(pos and neg)			**0.6624**

Figure 4: Confusion table and scores on task **B**, tweets

6 Discussion

We modified an existing rule-based system for SemEval Task 2. While the development of this existing system was a considerable time investment, the modifications for the two SemEval subtasks took no more than about 2 person-weeks in total. The models used in task A and B have a large common base, and our rule-based approach measures up well against other systems. This shows that if the work is done once, it can be re-used, modified, and refined.

As mentioned in section 4.1, the annotations did not always seem consistent. The guidelines did not ask the annotators to keep in mind a particular task or purpose for their annotations. However, the correct annotation of a tweet or fragment can vary depending on the purpose of the annotation. Non-arbitrary choices have to be made as to what counts as sentiment: *Do you try to identify cases of implicit sentiment? Do you count cases of quoted or reported '3rd-party'-sentiment? ...* Ultimately it depends on what you are interested in: *Do you want to: -track sentiment around certain topics? -know how authors are feeling? -assess the general mood? -track distressing versus optimistic messages in the news? ...* While manual rule writing allows us to choose a consistent strategy, it was not obvious what the optimal strategy was in this SemEval task.

There were considerable differences in annotation strategy between task A and task B, which shared the same tweets. The threshold for detecting sentiment appeared to be considerably lower in task A than in task B. This suggests that different choices had been made. These choices probably reflect how the annotators perceived the tasks.

In our core business, we primarily track sentiment around brands. One of the choices we made was to also include good and bad news about the brand (such as that the company's stock went up or down) where no explicit sentiment is expressed, because the circulation of such messages reflects on the reputation of the brand. (Liu (2010) points out that a lot of sentiment is implicit.) In task B, we noticed that 'newsy' tweets had a tendency to be annotated as neutral. We did not have the time to thoroughly adapt our model for that interpretation.

Both manually annotating training data for supervised machine learning and using training data for manual rule writing require a lot of work. Both can be crowd-sourced to a large extent if the process is made simple enough, and the instructions are clear enough. All methods that use lists of sentiment terms benefit from automatically extracting such terms from a corpus (Qiu et al., 2009; Wiebe and Riloff, 2005). As those methods become more sophisticated, the work of rule writers becomes easier. Since the correct annotation depends on the task at hand, and there are many different choices that can be made, annotated data can be hard to reuse for a slightly different task than the one for which it was created. In rule-based models it is easier to leverage earlier work and to slightly modify the model for a new task. Both the rules and the model's decision-making process are human-interpretable.

Table 1 (next page) summarizes our results on the various portions of the task, and under different conditions. The results on SMS-data are consistently lower than their counterparts on tweets, but they follow the same pattern. We conclude that the model generalizes to SMS, but not perfectly. This is not surprising, since we have never looked at SMS-data before, and the genre does appear to have some idiosyncrasies.

7 Conclusion

Our model is essentially a highly phrasal sentiment lexicon. Ways of defining slightly more abstract patterns keep the amount of work and the number of rules manageable. The model is applied through pattern matching on text, and returns a sentiment prediction based on the number of positive and negative expressions found, based on the sum of their weights. This is not mediated by any machine learning.

Slightly different versions of this system were employed in subtasks A and B. It turned out to be a strong competitor in Task 2 of SemEval-2013, especially on subtask B, where it scored in the top three.

References

Russell Albright and Praveen Lakkaraju. 2011. Combining knowledge and data mining to understand sentiment: A practical assessment of approaches. Technical report, SAS White Paper, January.

	Task A Twitter		Task A SMS		Task B Twitter		Task B SMS	
	F-score	rank	F-score	rank	F-score	rank	F-score	rank
Submitted	0.7489	*3of7* *13of23*	0.7283	*4of7* *11of19*	0.6486	***1of15*** *3of34*	0.5910	***2of15*** *5of29*
After fixing encoding	0.7966	*3of7* *11of23*	0.7454	*3of7* *8of19*	0.6624	***1of15*** ***2of34***	0.6014	***1of15*** *4of29*
With guessing	0.8327	(*2of7*) (*8of23*)	0.7840	(*2of7*) (*7of19*)	NA		NA	

Table 1: Summary of results. The first rank indication is relative to the other systems in the unconstrained category. The second is relative to the total number of participating teams (by highest scoring system).

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 7th conference on International Language Resources and Evaluation (LREC10), Valletta, Malta, May*.

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*, pages 793–801. Association for Computational Linguistics.

Xiaowen Ding, Bing Liu, and Philip S Yu. 2008. A holistic lexicon-based approach to opinion mining. In *Proceedings of the international conference on Web search and web data mining*, pages 231–240. ACM.

Jacob Eisenstein. 2013. What to do about bad language on the internet. In *Proc. of NAACL*.

Praveen Lakkaraju and Saratendu Sethi. 2012. Correlating the analysis of opinionated texts using sas text analytics with application of sabermetrics to cricket statistics. In *Proceedings of SAS Global Forum 2012*, number 136.

Kathy Lange and Saratendu Sethi. 2011. What are people saying about your company, your products, or your brand? In *Proceedings of SAS Global Forum 2011*, number 158.

Bing Liu. 2010. Sentiment analysis and subjectivity. *Handbook of natural language processing*, 2:568.

Tetsuya Nasukawa and Jeonghee Yi. 2003. Sentiment analysis: Capturing favorability using natural language processing. In *Proceedings of the 2nd international conference on Knowledge capture*, pages 70–77. ACM.

Rudy Prabowo and Mike Thelwall. 2009. Sentiment analysis: A combined approach. *Journal of Informetrics*, 3(2):143–157.

Guang Qiu, Bing Liu, Jiajun Bu, and Chun Chen. 2009. Expanding domain sentiment lexicon through double propagation. In *Proceedings of the 21st international joint conference on Artifical intelligence*, pages 1199–1204.

Maite Taboada, Julian Brooke, Milan Tofiloski, Kimberly Voll, and Manfred Stede. 2011. Lexicon-based methods for sentiment analysis. *Computational linguistics*, 37(2):267–307.

Janyce Wiebe and Ellen Riloff. 2005. Creating subjective and objective sentence classifiers from unannotated texts. In *Computational Linguistics and Intelligent Text Processing*, pages 486–497. Springer.

Michael Wiegand, Alexandra Balahur, Benjamin Roth, Dietrich Klakow, and Andrés Montoyo. 2010. A survey on the role of negation in sentiment analysis. In *Proceedings of the workshop on negation and speculation in natural language processing*, pages 60–68. Association for Computational Linguistics.

Theresa Wilson, Paul Hoffmann, Swapna Somasundaran, Jason Kessler, Janyce Wiebe, Yejin Choi, Claire Cardie, Ellen Riloff, and Siddharth Patwardhan. 2005. Opinionfinder: A system for subjectivity analysis. In *Proceedings of HLT/EMNLP on Interactive Demonstrations*, pages 34–35. Association for Computational Linguistics.

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

Yunfang Wu and Peng Jin. 2010. Semeval-2010 task 18: Disambiguating sentiment ambiguous adjectives. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, pages 81–85. Association for Computational Linguistics.

Shi-Cai Yang and Mei-Juan Liu. 2010. Ysc-dsaa: An approach to disambiguate sentiment ambiguous adjectives based on saaol. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, pages 440–443. Association for Computational Linguistics.