Sentiment Analysis of Customer Feedback in the Business Domain

Omar El-Begawy*, **Robin Wikström**†, **Hannu Toivonen***

Åbo Akademi University
*Department of Information Technologies
†Laboratory of Industrial Management
20500 Turku, Finland
{omar.elbegawy,robin.wikstrom,hannu.toivonen}@abo.fi

Abstract

For companies to understand both their business and their customers, they need to work through the large amounts of feedback received. Analyzing feedback retrieved in a business-to-consumer context is less complex than analyzing comments from a business-to-business context. These comments tend to be lengthier, more grammatically advanced and use domain specific terminology. Though being more challenging, the benefits from automatizing the analysis of these comments include cost savings and the ability to analyse larger amounts of text at a time. In this article, we present the results of using recursive neural tensor networks to classify feedback retrieved from company representatives into different sentiment categories. The performance of the classifier is evaluated and discussed. Especially the problem of addressing grammatical errors is pinpointed.

1. Introduction

For companies to understand the way in which their customers make decisions, they need to understand text. If companies can "listen to the customer" there are a lot of benefits to gain. Traditionally, this means conducting interviews and surveys and reading what the customer has said about them. Although there is nothing wrong with this approach, however, sometimes the volume of this information can be large, and in such cases there is a need for an automated way of making sense of the data. Furthermore, the amount of available data and the number of different sources of data has increased rapidly, especially due to social media applications, organizations are more and more utilizing this potential knowledge source (Ittoo et al., 2015).

The field of sentiment analysis aims to provide tools which facilitate the analysis of large amounts of opinionated commentaries by people. Besides being a field which spans multiple research fields, such as data mining, pattern recognition, optimization, machine learning, and others, sentiment analysis is also interesting from a business perspective by giving more insights into customer behavior. The field of sentiment analysis also considers tasks such as opinion mining and subjectivity analysis, all of which denote similar and overlapping fields of study (Pang and Lee, 2006; Turney, 2002).

Sentiment analysis can generate useful information to industry analysts. The study of sentiment can reveal valuable patterns regarding the opinions of a product or service. This information can be used for public relations, marketing and product development. Rather than using much time to extract opinions from unstructured human-authored documents, a computerized approach can be used to facilitate and speed up the process. The result can be a condensed version of the opinions expressed and a saving in expert labor power. Furthermore, in a lot of cases, there is a need to quickly get an overview of the collected data. By using a sentiment analysis tool to quickly work through the data, one can receive a rough estimate on short notice (Liew et al., 2014).

2. Goal of the paper

We conducted sentiment analysis on customer feedback. This feedback has been collected as part of the analysis of industrial projects in the shipping industry, in order to collect information about company processes. While customer feedback is widely available, our study focuses on feedback given by company representatives to those of another company. This data often contains incomplete sentences with linguistic errors, and the performance of classifiers is not clear under such circumstances. This study will address an attempt to apply sentiment analysis on text with a high number of imperfections.

The rest of this paper is structured as follows: Section 3 details the materials and methods used, Section 4 presents the results of the study and Section 5 concludes the paper.

3. Sentiment Analysis of Customer Feedback

This section introduces the data utilized and the technology applied for the sentiment analysis.

3.1. The Data

For the analysis we used a corpus consisting of 349 questionnaire answers. While the answers were written in understandable English, they contained a lot of spelling mistakes and grammatical errors. These were not answers written by professional writers, nor edited afterwards. While a human reviewer can easily notice such errors, a computer program is less able to do so.

3.2. Sentiment analysis models

Pretrained models were used in this study since the small dataset was not large enough to conduct training on by itself. There was too much variation in the sentences for a classifier to establish reliable patterns. Limited data can lead either to bias or inability to make informed classifications.

3.2.1. *pattern*

The *pattern* library classifies sentiments on the basis of adjectives that the sentence contains[1]. These sentences have been graded according to their sentiment polarity. The *pattern* library approach is a classic example of a keyword based sentiment analysis.

3.2.2. VADER

VADER (Valence Aware Dictionary for sEntiment Reasoning) is a sentiment analysis library that uses a rule-based approach to classify the sentiments of sentences (Hutto and Gilbert, 2014). It uses lexical features mined from several corpora, and human curated heuristics. These are combined in a rule engine for sentence sentiment classification. The authors emphasize the suitability of VADER for microblog texts.

3.2.3. Recursive Neural Tensor Networks

Recursive neural networks (RNNs) form a family of neural networks whose first use was in the classification of logical terms (Goller and Küchler, 1996). The network is called recursive because the information is transmitted between repeating patterns of one parent and two children (Socher et al., 2012). Recursive neural networks can be generalized further to give recursive neural tensor networks (RNTNs) (Socher et al., 2013b). The RNTNs have the same binary network structure as the RNN. Novel in the RNTN is the use of tensors to allow more interactions between the input vectors. This gives more parameters with which to calculate the composition of child node vectors (Dong et al., 2014).

Recursive neural tensor networks have been used for sentiment analysis (Socher et al., 2013b) and relationship reasoning (Socher et al., 2013a). In sentiment analysis RNTNs have been shown to outperform other methods, and are also able to classify correctly sentences with negations (e.g. not good) and negated negations (e.g. not bad).

The RNTN has been trained on the Rotten Tomatoes movie review corpus (Pang and Lee, 2006). However, rather than doing a sentence by sentence labeling, as has been done earlier in the field, the labeling has been done at a more fine-grained level. The 10,662 sentences of the corpus were parsed with the Stanford Parser (Klein and Manning, 2003) into 215 154 phrases, with lengths ranging from one word until the full sentence. Each of these phrases was reviewed for the intensity of negative or positive sentiment. The relation of the phrases to the sentence was described through a binary tree structure. By examining the way in which the phrases build up the original sentence, the model can evaluate how the overall sentiment of a sentence is built up on the basis of its constituent parts. The collection of these tree-like descriptions of the sentences is called the Stanford Tree Bank, and forms the training data for the neural network.

3.3. Technical and Methodological Foundation for the Sentiment Analysis

Three different libraries were used for the sentiment analysis. Both *pattern* and VADER are implemented as Python libraries, and were accessed through the APIs provided by TextBlob library[2], version 0.13.0, and NLTK[3], version 3.2.4. Recursive neural tensor networks have been applied in the the Stanford NLP Core sentiment classifier (Manning et al., 2014; Chen and Manning, 2014; de Marneffe et al., 2006), available as a Java library and this study used version 3.5.2. Only tools that were freely available and could be run on a local computer were used.

In order to make the classifiers' results comparable to those of the human evaluators, the outputs were truncated into three classes, positive, neutral and negative.

pattern gives a polarity score in the range $[-1,1]$, and VADER gives a similar compound score in the same range. In order to assign the sentences into to the given classes, a score in the range $[-1,-0.33[$ classified the answer as negative, the range $[-0.33,0.33]$ as neutral, and $]0.33,1]$ as positive.

The Stanford API's classifier was used to assign the same set of sentences into five classes: very negative, negative, neutral, positive, and very positive. The very negative and negative classification results were combined into one class of negative, and similarly the results in the positive and very positive classes were combined into one positive class. It was not seen to have a noticeable impact on the classifiers output, since there were no answers classified as very positive and only one answer classified as very negative.

4. Results

This section presents the results of the study, as well as some performance evaluations. The confusion matrices containing more detailed results for the classifications can be found in Appendix 1.

4.1. Classification Results

A corpus of 349 previously unlabeled answers was used. These were labeled by the authors into three classes: positive, neutral and negative. In cases where the sentiment seemed ambiguous, the answers were labeled as neutral. The corpus contained 24 positive answers, 189 neutral answer and 126 negative answers, making the distribution of the corpus very uneven.

The RNTN classifier had the highest recall rate for negative elements, 77.78%. Less than a quarter of the negative answers were falsely classified. But for the neutral class it also had the lowest recall rate of 45.50%, with the classifier classifying over half of the neutral answers as other than neutral. *pattern* was the best performer for neutral sentences with having a recall of 91.53%. *pattern* and VADER also categorised most sentences in the neutral category, which might also account for their high recall rate. For positive recall the results were more even, with 35.29%,

[1] Here `https://github.com/sloria/TextBlob/blob/dev/textblob/en/en-sentiment.xml` and here `https://www.clips.uantwerpen.be/pages/pattern-en#sentiment`

[2] `https://github.com/sloria/TextBlob`

[3] `http://www.nltk.org/_modules/nltk/sentiment/vader.html`

64.71% and 50.00% respectively for *pattern*, VADER and RNTN.

Class	*pattern*	VADER	RNTN
negative	3.17%	29.36%	77.78%
neutral	91.53%	70.37%	45.50%
positive	35.29%	64.71%	50.00%

Table 1: Classifier recall for each class.

The precision for the negative classification was highest for VADER, with 80.43%. For the neutral and positive classifications, the highest precision was given by RNTN, with 79.63% and 34.69%.

Class	*pattern*	VADER	RNTN
negative	66.67%	80.43%	51.04%
neutral	56.17%	62.44%	79.63%
positive	34.29%	24.44%	34.69%

Table 2: Classifier precision for each class.

The overall accuracy for the *pattern*, VADER and RNTN classifiers was 54.15%, 55.01% and 57.59%, respectively, while the accuracy for a random guess was 37.99%, se Table 3. The proximity of the accuracies is probably due to the uneven class distribution in the data. *pattern* and VADER predict most often neutral classes, which is also the most frequent class in the data.

The RNTN had the highest calculated Cohens κ 0.316. (Viera and Garrett, 2005). Compared to the other classifiers, the RNTN's performance can be said to be fair, even though the uneven distribution of the data does not say anything conclusive about the classifiers performance.

	pattern	VADER	RNTN
Accuracy	54.15%	55.01%	57.59%
κ	0.094	0.246	0.316

Table 3: Accuracy and Cohen's κ for different classifiers.

4.2. Effects of Grammatical Errors on Classification

Many of the comments in the corpus were ungrammatical, fragmentary, idiosyncratic and written in a telegraphic style. This differs from standard corpora used for training sentiment classifiers, in that they usually consist of complete and grammatically correct sentences. We, nevertheless, applied it on this type of textual data and in order to see whether language mistakes had any effect on the classification results, a sample of the sentences were selected and corrected for a second analysis. The results of the new analysis did not show much change, thus indicating that the sentiment analyzer was quite tolerant of language errors.

In Figure 1 the RNTN's sentiment buildup for three variations of the same sentence are shown. The original sentence *Customer not able to deliver payment security according to contract*, is erroneous in two ways: it lacks the verb *is* which defines the customers activity, and it also misses a period at the end.

In Figure 1, section A the original sentence is shown. The second version in section B has the grammar corrected, and in section C the negation has been turned into a positive. For section A which has a negative sentence incorrectly labeled as neutral, the addition of a period to the end corrects the classification to negative. This has the same result as correcting the grammar in section B. The grammatically correct sentence is classified correctly as positive irrespective of the ending period.

5. Discussion and Summary

As the RNTN classified nearly a third of all positive answers as negative, the RNTN classifier was, at least according to the available data, not very good at classifying positive answers correctly. The low number of positive answers in the corpus is likely to be the main reason why the classifier had a poor performance for the positive class. Had there been more positive answers, the proportions might have looked different. The classifier had the highest rate of correct classifiers for the negative answers. One could speculate that the classifier is better suited for finding negative answers.

One reason why the RNTN classification results were not always able to classify correctly is that the sentences in the corpus differed too much from those in the training set. Sentiment analysis is often domain specific, and therefore a sentiment analyzer trained on movie reviews can be unable to provide a good classification of customer feedback. A further reason why RNTN did not predict well is that the answers were structurally different in that they were often short phrases or consisted of multiple sentences, while the training set consisted of single sentences that were complete.

This might not have been such an issue for *pattern* and VADER because they use keywords and heuristics to assess sentiment. These might be more suitable approaches for short text fragments with uncomplicated structures.

Short, ungrammatical sentences have been studied in the example of tweets on Twitter (Agarwal et al., 2011; Go et al., 2009). However, besides the sentence itself these have the advantage of high number of training data, non-linguistic sentiment cues (e.g. emoticons) or metadata (e.g. likes, retweets). Therefore using tweet-based sentiment analysis tools might not be suitable outside that domain.

Traditional neural networks tend to have the problem of not being clear about the ways in which they arrive at their conclusions. They rarely produce results that can be used for usable insights, as they might assign high importance to nodes that may make the model work properly with respect to analysing the data, but do not provide a generalization of data easily understandable for human thinking. The RNTN neural network trained on the Stanford Sentiment Treebank, however, differs because the data used in its training was so fine-grained that the neural network resembles a meaningful decision-tree. These trees provide more understandable models of sentence structures.

One way of improving the RNTN classification results would be the use of tools for grammatical normalisation,

Classified as	Real class	Example sentence
Negative	Positive	*Small project with small organization are normally easy to handle* *We are now finishing of punch list and very close co-operation and strong nervs are needed from our side, however the worst is over*
Positive	Negative	*Many quality problems with our products, still several that are pending* *Budget of the project is very tight*

Table 4: Examples of incorrectly classified sentences.

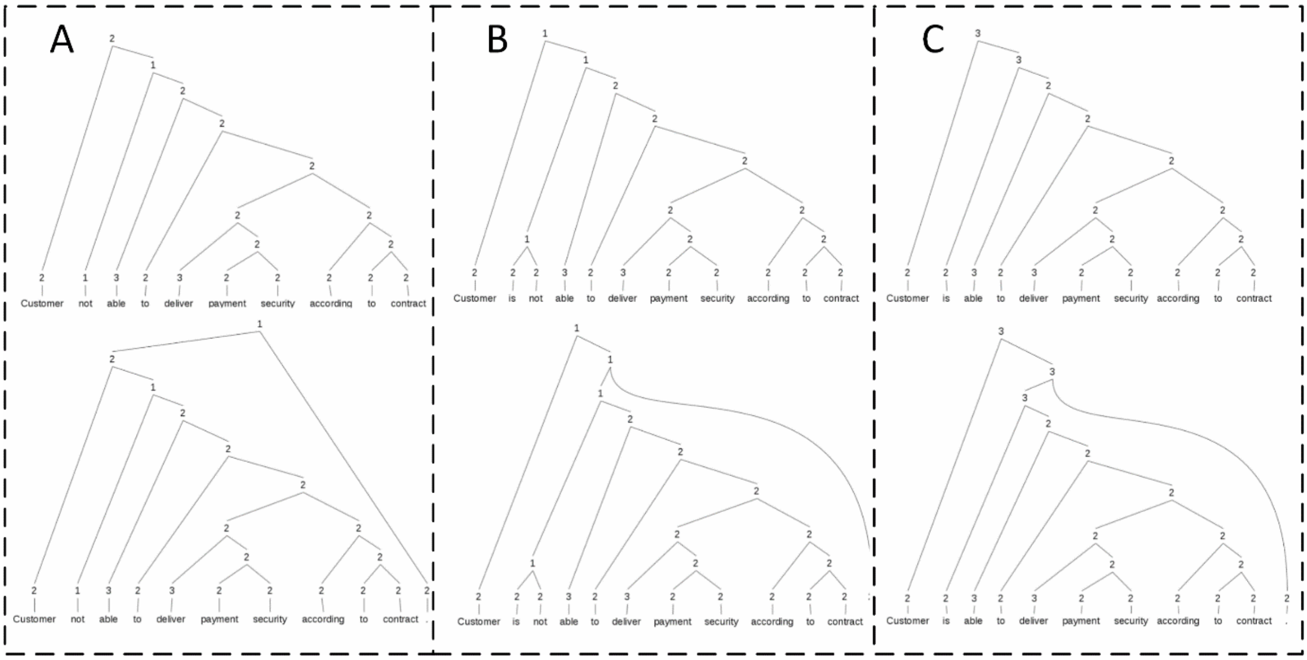
A
B
C
Customer not able to deliver payment security according to contract
Customer is not able to deliver payment security according to contract
Customer is able to deliver payment security according to contract
Customer not able to deliver payment security according to contract .
Customer is not able to deliver payment security according to contract .
Customer is able to deliver payment security according to contract .

Figure 1: Result obtained with different sentences: A (left panels): *Customer not able to deliver payment security according to contract.* B (middle panels): *Customer is not able to deliver payment security according to contract.* C (right panels): *Customer is able to deliver payment security according to contract.* The top trees do not have a period at the end, while the bottom ones end with a period. The numbers in the nodes correspond as follows: 1=negative, 2=neutral and 3=positive.

such as grammar correctors and spell checkers. This would make the questionnaire answers syntactically more similar to those in the training set. Another way would be the creation of a sentiment treebank specific for the business domain. This would make it possible to use the RNTNs to train a classifier which would be better at classifying the questionnaire answers. The main problem here is still the use of domain specific jargon, which makes it difficult even for human readers to interpret some answers.

Appendix

This appendix contains the confusion matrices for the sentiment analysis. The confusion matrix for the *pattern*, VADER and RNTN classifiers are given in Table A1, Table A2 and Table A3. These provide the basis for the calculated recall, precision and accuracy results in the main text.

6. References

Agarwal, Apoorv, Boyi Xie, Ilia Vovsha, Owen Rambow, and Rebecca Passonneau, 2011. Sentiment analysis of twitter data. In *Proceedings of the Workshop on Languages in Social Media*, LSM '11. Stroudsburg, PA, USA: Association for Computational Linguistics.

Chen, Danqi and Christopher Manning, 2014. A fast and accurate dependency parser using neural networks. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*.

de Marneffe, Marie-Catherine, Bill MacCartney, and Christopher D Manning, 2006. Generating typed dependency parses from phrase structure parses. In *Proceedings of the 5th International Conference on Language Resources and Evaluation (LREC 2006)*.

Dong, Li, Furu Wei, Zhou Ming, and Ke Xu, 2014. Adaptive multi-compositionality for recursive neural models with applications to sentiment analysis. In *Proceedings of the Twenty-Eighth AAAI Conference on Artificial In-*

		Truth data			
		negative	neutral	positive	classification overall
Classifier results	negative	4	2	0	6
	neutral	113	173	22	308
	positive	9	14	12	35
	reference overall	126	189	34	349

Table A1: Confusion matrix for the *pattern* classifier.

		Truth data			
		negative	neutral	positive	classification overall
Classifier results	negative	37	8	1	46
	neutral	69	133	11	213
	positive	20	48	22	90
	reference overall	126	189	34	349

Table A2: Confusion matrix for the VADER classifier.

		Truth data			
		negative	neutral	positive	classification overall
Classifier results	negative	98	84	10	192
	neutral	15	86	7	108
	positive	13	19	17	49
	reference overall	126	189	34	349

Table A3: Confusion matrix for the RNTN classifier.

telligence, volume 24. AAAI Press.

Go, Alec, Richa Bhayani, and Lei Huang, 2009. Twitter sentiment classification using distant supervision. 150:Entropy.

Goller, Christoph and Andreas Küchler, 1996. Learning task-dependent distributed representations by backpropagation through structure. In *Proceedings of the International Conference on Neural Networks*, volume 4.

Hutto, Clayton J. and Eric Gilbert, 2014. Vader: A parsimonious rule-based model for sentiment analysis of social media text. In Eytan Adar, Paul Resnick, Munmun De Choudhury, Bernie Hogan, and Alice H. Oh (eds.), *Eighth International Conference on Weblogs and Social Media (ICWSM-14)*. The AAAI Press.

Ittoo, Ashwin, Le Minh Nguyen, and Antal van den Bosch, 2015. Text analytics in industry: Challenges, desiderata and trends. *Computers in Industry*:1–12.

Klein, Dan and Chistopher D Manning, 2003. Accurate unlexicalized parsing. In *ACL '03 Proceedings of the 41st Annual Meeting on Association for Computational Linguistics*, volume 1.

Liew, Wan Te, Arief Adhitya, and Rajagopalan Srinivasan, 2014. Sustainability trends in the process industries: A text mining-based analysis. *Computers in Industry*, 65:393–400.

Manning, Christopher D, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J Bethard, and David McClosky, 2014. The stanford corenlp natural language processing toolkit. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*.

Pang, Bo and Lillian Lee, 2006. Opinion mining and sentiment analysis. *Foundations and Trends in Information Retrieval*, 1:91–231.

Socher, Richard, Danqi Chen, Christopher D Manning, and Andrew Ng, 2013a. Reasoning with neural tensor networks for knowledge base completion. In *Proceedings of the Advances in Neural Information Processing Systems 26 (NIPS 2013)*.

Socher, Richard, Brody Huval, Christoper D Manning, and Andrew Y Ng, 2012. Semantic compositionality through recursive matrix-vector spaces. In *Proceedings of the 2012 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Socher, Richard, Alex Perelygin, Jean Y Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, and Christopher Potts, 2013b. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*.

Turney, Peter D, 2002. 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*. Association for Computational Linguistics.

Viera, Anthony J and Joanne M Garrett, 2005. Understanding interobserver agreement: The kappa statistic. *Family Medicine*, 37:360–363.