Sentiment Analysis for Social Media Comments

Christoph Emunds, Benedikt Heinrichs, Dominik Nerger, Richard Polzin

June 27, 2017

Abstract

Analyzing customer experience offers valuable data for Business Intelligence. Especially in e-commerce, where users write reviews about products and services, the analysis of the customers' sentiment towards these entities yields significant insights into potential strengths and weaknesses of the product or service.

Our work focuses on evaluating customer experience through the application of aspect-based sentiment analysis on social media posts. The data set contains a year's worth of Facebook comments on the pages of the two supermarket chains Tesco and Sainsbury. The goal is to extract as many triplets (e, a, s) as possible, where e is an entity (product or service), a is an aspect of this entity (performance, battery, politeness, etc.), and s is the sentiment polarity label (negative, neutral, positive).

To accomplish this task, many different subtasks need to be solved. After a rudimentary preprocessing routine, the posts need to be identify the part of speech (POS) category for every word. Then Named Entity Recognition (NER) must be applied and words containing sentiment need to be detected and evaluated. As a significant number of social media texts do not follow the grammatical rules or contain a lot of misspellings the complexity of those tasks is increased significantly. With the completed analysis we identify potential products and services, their corresponding aspects and sentiment words, which are scored and aggregated into opinions.

The results obtained from this analysis are visualized to get a better understanding of the customers' feelings towards these products, services, and their aspects. The visualization supports businesses in a wide range of decisions, such as product positioning and pricing. This can lead to better customer satisfaction, faster reactions to trends and overall revenue growth.

Contents

1 Introduction

Sentiment analysis or opinion mining uses techniques from natural language processing (NLP), text analysis, and computational linguistics to extract subjective information from texts. In applications that involve analyzing customer experience, this subjective information typically represents the position of a person towards a certain entity. Quantifying feelings and opinions that customers have towards products and services offers valuable information for the company.

Such information can be used to monitor reputation or improve the understanding of the market's needs. In contrast to generic sentiment analysis this information can only be acquired by precisely identifying the entities and aspects that a sentiment is expressed towards. This specific tasked is referred to as Aspect Based Sentiment Analysis (ABSA).

The project presented in this document focuses on ABSA, finding and analyzing customers' opinions on products and their aspects, from social media posts of a supermarket chain. In today's world, social media is a big part of the social life. On social networks, like Facebook, many companies represent themselves through a corporate account. This opens up the possibility for many people to interact with those companies. Everyone can leave a comment on the page of a company, celebrity, or a sports teams. People can review or comment with a lot of context. General posts can be created, there can be reactions to news or a posting can concern a specific product. These comments or reviews are often filled with positive or negative sentiments. It should therefore be possible to collect and summarize the information in these posts and extract an overview of the sentiment towards products or aspects.

2 Related work

The related work presented in this section gives an overview over the main techniques that are used to identify entities and their aspects.

2.1 POS-Tagging

POS-Tagging (part-of-speech tagging), also called grammatical tagging or word-category disambiguation, refers to the process of identifying particular parts of speech like nouns or verbs. Identifying the role of a certain word within a sentence is important for the task of sentiment analysis, as identifying entities and their corresponding aspects can be handled much easier relying on certain assumptions about the part of speech of words.

While being important POS-Tagging is also a complex problem. Word-forms in natural language are often ambiguous. For example the word 'dogs' is usually thought of as a plural noun, but can be used as a verb as well:

> The sailor dogs the hatch.

Due to the complexity, machine learning techniques are often applied in POS-Taggers. Popular approaches such as the Viterbi algorithm (Viterbi, 2006), the Brill tagger (Brill, 1992) or the Baum-Welch algorithm (Rabiner, 2012) work with techniques such as dynamic programming, supervised learning or hidden Markov models.

2.2 NER-Tagging

Named-Entity recognition (NER) is a task that seeks to locate and classify specific information in text. This information is called a named entity and can refer to categories such as the names of persons, locations, times or many others.

An annotated sentence could look like this :

> [Tim Cook]$_{[Person]}$ has a Net worth of [785 million USD]$_{[MonetaryValue]}$ as of [March 16. 2017]$_{[Time]}$.

While state-of-the-art NER-Taggers perform very well and produce near-human performance they are also brittle and do not perform well in domains they were not designed for (Thierry Poibeau, 2001).

3 Approach

3.1 Preprocessing

We pre-processed the data by extracting the actual text of the posts into separate files. We decided to exclude posts that are less than 20 characters long or include images. This is due to the fact that posts which are too short do not yield much information most of the time. Furthermore, posts that include images often refer to objects in the image, which makes it hard to understand the author's sentiment without analyzing the image content.

3.2 Extracting entities and aspects

For each post, the POS tags as well as different entities are obtained, with the entities being created through different Named Entity Recognition models. Both POS-tagging and Named Entity Extraction are accomplished with OpenNLP. They use maximum entropy, which takes the data and classifies it, while not assuming anything about the probability distribution, besides the things that have been observed.

The Named Entity Recognition in the English language is available for different entities, each with their own model. We utilized the following models: *person*, *location*, *organization*, *date* and *money*.

Because no pre-trained model exists for the Named Entity Recognition of products or services, we decided to use the OpenNLP model for organization to appropriately tag these. Most nouns in the English language are written in

lower case, but organizations as well as products contain capital letters. The OpenNLP model for organizations is a fitting alternative to bypass the non-existent models for products and services because of this. Therefore, we consider everything that is tagged as *Organization* as a product or service.

Searching for aspects is done by iterating over the complete set of posts for each entity. If a post contains the product, we identify the sentences it appears in. We then look for other nouns in these sentences and count their frequency. From those potential aspects, we consider the ones to be real aspects that occur at least in 10% of the posts that mentioned the product.

After identifying a set of aspects for each entity, we merge similar entities and their aspect lists by tokenizing the entities' names and lemmatizing them. This results in a reduction of entities, e.g. *Customer Service* and *Customer Services* are considered as the same entity. The entities' aspect lists are merged accordingly. All aspects that are also part of the entity's name are removed from the aspect list. Amongst other things, this removes the aspects *tesco* and *express* from the list of aspects for the entity *Tesco Express*.

If no aspects can be found for a specific entity, it is removed from the final list of entities. For every entity that is part of the final list, the aspect *general* is added as well.

3.3 Determining sentiment polarities

nltk's Vader module Specifically developed for social media

4 Results

We were given a data set containing a year's worth of Facebook comments on the pages of two supermarket chains Tesco and Sainsbury. However, we decided to perform our approach only on the Tesco data set.

Through performing our pre-processing tasks of removing posts with less than 20 characters or an image, we were able to reduce the amount of posts by 40.6%, from 37701 original posts down to 22395 afterwards.

By executing the POS- and NER-tagger, we came up with 1598 products that were mentioned in the posts. Because there were a lot of products that did not make sense, we removed some of them by hand, resulting in 1379 products. Furthermore, products that did not possess an aspect were removed as well. Therefore, 1256 products that at least contain one aspect were identified, decreasing the product list by 21.5%.

The original goal was to extract as many quintuples $(e_i, a_{ij}, s_{ijkl}, h_k, t_l)$ as possible, where e_i is the i'th entity and a_{ij} is the j'th aspect of entity i the opinion is expressed on. s_{ijkl} is the sentiment polarity, which can take on the values *positive* or *negative*. h_k describes the opinion holder and t_l the time at which the opinion was expressed.

However, from the data set that was given to us, it was not possible to determine the person that wrote a post. Moreover, we did not focus on extracting the time a certain sentiment was expressed, as we did not aim to provide a temporal overview over the shifting of opinions towards the products and services.

We labeled a small subset of posts concerning the entity *Customer Service* by hand. This includes 200 posts with 254 extracted sentiment triples with unknown sentiment.

After labeling these triples with the Vader module, 139 of the 254 triples (i.e. 54.72%) result in the same sentiment.

4.1 Visualization

We will use the sentiment lexicon by (Hu & Liu, 2004), which includes misspellings, morphological variants, slang and social-media mark-up of 2006 positive and 4783 negative words.

On the server side, we use Python with the library Flask. On the client side, we use the JavaScript libraries d3.js, vue.js and jQuery.

For the visualization, we use the *JSON* file containing the aggregated opinions that are the result of executing the pipeline as well as a *JSON* containing the positive and negative sentiment words that are provided by the lexicon-based sentiment analysis.

The data is visualized with each product having its own page that is loaded from a dropdown menu and each aspect of the product being visualized with a bar chart as well as the specific posts being shown below the bar chart. The bar chart can be seen in Figure 1. It shows the amount of negative, neutral and positive sentiments regarding the combination of product and aspect.

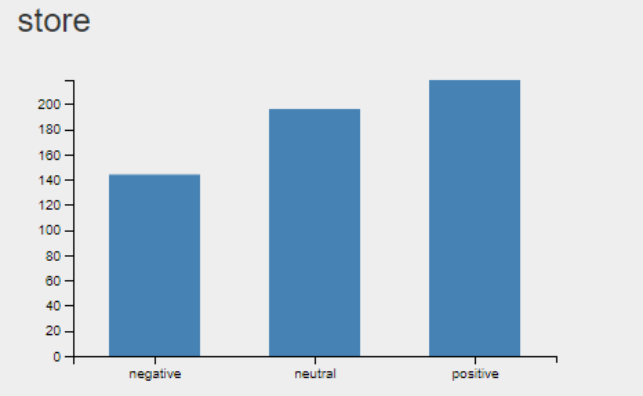

Figure 1: Bar chart displaying the sentiments regarding the customer service with aspect 'store'

An excerpt of posts regarding the combination of product and aspect is shown below each bar chart, with a maximum of 10 posts per aspect. In Figure 2, the highlighted posts can be seen. Positive and negative sentiments are highlighted

in colors green and red, respectively. Mentions of the product are highlighted in orange and the aspect is shown with a cyan highlighting.

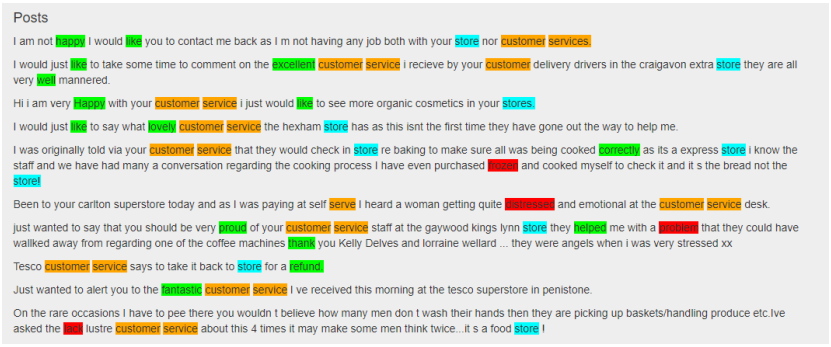

Figure 2: Highlighted posts regarding the customer service with aspect 'store'

5 Discussion and Conclusion

The accuracy of 54.72% as mentioned in Section 4 leaves a lot of potential for improvement. Several factors contribute to this rather bad accuracy score. First of all, the 200 posts that were labeled by hand are by no means indicative of the actual data set. Furthermore, we did not test different target functions for the polarity determination. Due to the lack of time, each of the four contributors tagged 50 posts. There is no guarantee that two people agree with what the other person labeled (would need Kappa measure).

We do not account for multi token aspects, since we only identify common nouns, which are single words.

Like mentioned in Section 3.2, we consider every organization found as product or service. This has many caveats. For one, it is required that services like *Customer Service* actually appear capitalized in our data set at least once to be found. Moreover, a lot of things that are not actually organizations are tagged as such. This could be improved by incorporating more knowledge regarding common nouns and noun phrases. Moreover, the results could be improved by finding a model that has been trained on products, which would result in more accurate entities and aspects.

6 Appendix

6.1 Using the code

To reproduce our work, the provided code has to be executed in a certain sequence. The following will talk about the needed steps.

References

Brill, E. (1992). A simple rule-based part of speech tagger. In *Proceedings of the third conference on applied natural language processing* (pp. 152–155). Stroudsburg, PA, USA: Association for Computational Linguistics. Retrieved from http://dx.doi.org/10.3115/974499.974526 doi: 10.3115/974499.974526

Hu, M., & Liu, B. (2004). Mining and summarizing customer reviews. In *Proceedings of the tenth acm sigkdd international conference on knowledge discovery and data mining* (pp. 168–177). New York, NY, USA: ACM. Retrieved from http://doi.acm.org/10.1145/1014052.1014073 doi: 10.1145/1014052.1014073

Rabiner, L. (2012). *First-hand:the hidden markov model.* Retrieved from http://ethw.org/First-Hand:The_Hidden_Markov_Model

Thierry Poibeau, L. K. (2001). Proper name extraction from non-journalistic texts. *Language and computers*, *37*, 144–157.

Viterbi, A. (2006, September). Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *IEEE Trans. Inf. Theor.*, *13*(2), 260–269. Retrieved from http://dx.doi.org/10.1109/TIT.1967.1054010 doi: 10.1109/TIT.1967.1054010