Learning to Analyze Relevancy and Polarity of Tweets

Rianne Kaptein

Oxyme
Amsterdam, The Netherlands
rianne@oxyme.com

Abstract. This paper describes the participation of Oxyme in the profiling task of the RepLab workshop. We use a machine learning approach to predict the relevancy and polarity for reputation. The same classifier is used for both tasks. Features used include query dependent features, relevancy features, tweet features and sentiment features. An important component of the relevancy features are manually provided positive and negative feedback terms. Our best run uses a Naive Bayes classifier and reaches an accuracy of 41.2% on the profiling task. Relevancy of tweets is predicted with an accuracy of 80.9%. Predicting polarity for reputation turns out to be more difficult, the best polarity run achieves an accuracy of 38.1%.

1 Introduction

The constantly growing volume of online opinions cannot longer be ignored by any company. The unsolicited opinions of consumers in the public domain are both a threat and an opportunity. Negative messages can harm your company's reputation, while positive messages or adequate responses to negative messages can lift your reputation. The goal of the RepLab Profiling task is to assign two important annotations to tweets:

1. Relevancy: Is the tweet related to the company?
2. Polarity for Reputation: Does the tweet have positive or negative implications for the company's reputation?

Concerning relevancy there are some differences compared to standard information retrieval tasks such as web search :

– Standing queries
  In most search scenarios the users create queries on the fly according to their information need at that moment. Queries might be adjusted according to the search results retrieved by their initial queries. For this task however the information need is clear, i.e. all tweets about the company in question. The same query is used to retrieve results over a longer period of time.

– Binary relevancy decisions
  Although relevancy of a search result is usually a binary decision, it is either relevant or non-relevant, the output of search systems is often a ranking of documents with the most relevant documents on top. For this task no ranking is generated, only the binary annotation relevant or non-relevant is assigned to each tweet.

Determining the polarity for reputation has some differences in comparison with standard sentiment analysis, but in our approach which learns from the training data this should not be an issue. The biggest challenges for the polarity for reputation analysis are:

– Dealing with two different languages: English and Spanish.
– The companies in the training data and the test data are different.

The main goals of our participation in this task are to:

– Explore explicit relevance feedback
  In web search it has always been difficult to extract more information from the user than the keyword query. In this type of search however queries are used for an extended period of time to retrieve many results, so the pay-off of explicit relevance feedback is higher.
– Devise a transparent method to analyse polarity for reputation
  Most Twitter sentiment analysis tools are far from perfect, and reach an accuracy anywhere between 40% and 70%. We do not expect our approach to work perfect either, so what is important in the interaction with users is to be able to explain why a tweet is tagged with a certain sentiment.

In the next section we describe our approach to determining relevancy and polarity of tweets. Section 3 describes the experimental set-up and results. Finally, in Section 4 the conclusions are presented.

2 Approach

To determine the relevancy and the polarity for reputation we use a machine learning approach which is the same for both tasks. We use standard machine learning algorithms, including Naive Bayes, Support Vector Machines and Decision Trees. The features used in the machine learning algorithms are a combination of features found in related work.

2.1 Query Dependent Features

The first group of features we use to determine the relevancy of tweet are features that depend on the query only, and not on specific tweets as suggested by [5]. In our approach we use the following features:

– Wikipedia Disambiguation This feature has 4 possible values, the higher the value, the more ambiguous the query:

    - 0: There is no disambiguation page for the entity name
    - 1: There is a disambiguation page, but the page with the entity name leads directly to the entity
    - 2: There is a disambiguation page, and the page with the entity name leads to this disambiguation page.
    - 3: The page with the entity name leads to another entity
- Is the query also a common first or last name?
- Is the query also a dictionary entry?
- Is the query identical to the entity name? Here we disregard corporation types such as 'S.A'. Abbreviations such as 'VW', and partial queries such as 'Wilkinson' for the entity 'Wilkinson Sword' are examples where the query is not identical to the query name.
- The amount of negative feedback terms. This feature has 3 possible values:
    - 0: No negative feedback terms
    - 1: 1 to 3 negative feedback terms
    - 2: 4 to 10 negative feedback terms
    - 3: More than 10 negative feedback terms
- Query difficulty, this feature is a combination of all features above. The higher the value of this feature, the more difficult it will be to retrieve relevant results.

All of these features are language dependent, since for example a common name in English does not have to be a common name in Spanish.

2.2 Relevancy Features

The second group of features is based on relevancy. We use the language modeling approach to determine the relevancy of the content of a tweet. Besides the search term we also use manual relevance feedback. For each query in both languages we generate a list of positive and negative relevance feedback terms. To generate the list of feedback terms we make use of the background corpus that is provided. From the background corpus that consists of 30,000 tweets crawled per company name we extract the most frequently used terms and visualize these in wordclouds using a wordcloud generator tool [1]. For each query we create two wordclouds, one wordcloud from the tweets that contain the search term and one wordcloud from the tweets that do not contain the search term.

The tweets that do contain the search term can still contain positive and negative feedback terms. In Figure 2.2 the wordcloud for 'Nivea' is shown. By clicking on a term in the cloud, the tweets containing that term are displayed. This allows us to quickly explore the large amount of tweets. In the example we clicked on the term 'song'. When we inspect the associated tweets, it turns out 'Nivea' is also the name of a band. We therefore add this term to the negative feedback terms, as well as other words related to the band Nivea, such as the song titles 'Complicated' and 'Don't mess with my men'. Positive feedback terms include 'body', 'cream', 'care', etc. There is also a number of words in the wordcloud which are general words which could appear both in relevant and

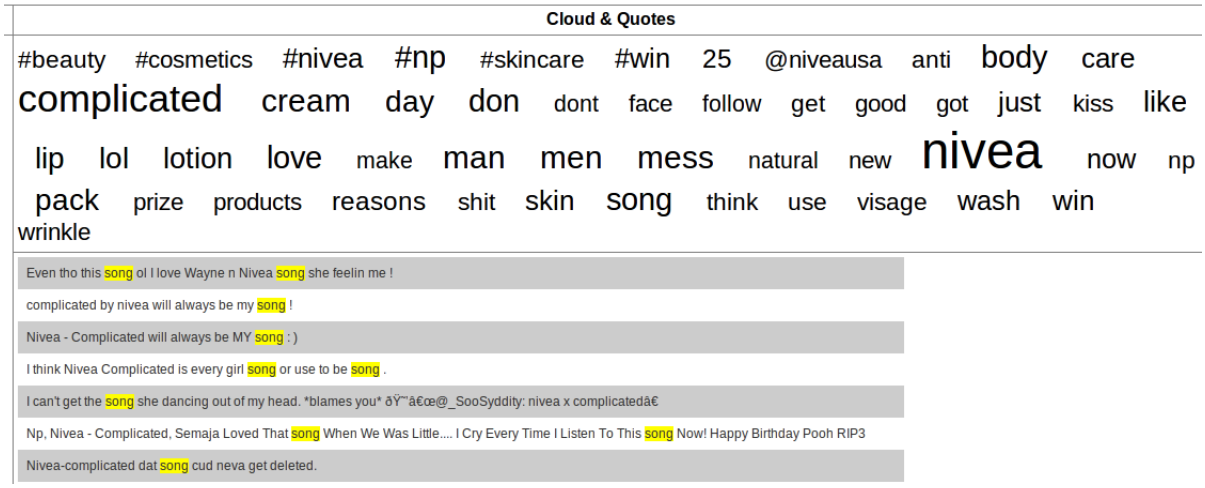

Fig. 1. The wordcloud from tweets containing the term 'Nivea'

non-relevant tweets, such as 'follow' and 'lol'. These words are not added to any of the feedback term sets.

In Figure 2.2 the wordcloud of the tweets that do not contain the search term as a separate word is shown. This can happen for example when the search term is part of a username. In this case we inspect the tweets to see if the username belongs to a Twitter account about Nivea only, or owned by Nivea. If not, we add the username to the negative feedback terms. Also, in case of doubt the public Twitter user profile can be checked. In Figure 2.2 we clicked on the username '@nivea_mariee_'. From the displayed tweets we conclude this account is not relevant for Nivea, so we add '@nivea_mariee_' to the negative feedback terms. The other terms are treated the same way as the terms in the previous wordcloud, so for each term we decide whether to add it to the positive or negative feedback terms or neither. There is actually quite some overlap between the two clouds, since all of the tweets are search results returned by Twitter for the query 'Nivea'. If we would have some training data available for this company we could look at the words which are used more often in positively or negatively rated tweets.

To calculate the relevancy score we use the following formula:

$$\log P(R|D,Q) = \log P(R|D) + \sum_{t \in Q_{pos}} (P(t|Q) \log P(t|D) - P(t|Q) \log P(t|C)) - \sum_{t \in Q_{neg}} (P(t|Q) \log P(t|D) - P(t|Q) \log P(t|C))$$

where R stands for the probability a document D is relevant given query Q, t stands for a term in a document or a query, and C for the background collection. The positive query terms Q_{pos} consist of the search terms plus the positive feedback terms. The document probabilities $P(t|D)$ are smoothed as

Fig. 2. The wordcloud from tweets that do not contain the term 'Nivea'

follows to avoid taking the log of 0:

$$P(t|D) = \alpha P(t|D) + (1 - \alpha)P(t|C)$$

where we use a value of 0.1 for the smoothing parameter α.

Q_{neg} contains the negative feedback terms. All probabilities are calculated in the log space, so that the numbers do not become too small. $P(R|D)$ is the prior probability of a document being relevant. Here we use a length prior based on the number of characters in a tweet to favour longer tweets.

The relevancy scores of each query are normalised using min-max normalization, where the minimum score is simulated by a document that contains only the negative relevance feedback terms, and the maximum score is simulated by a document that contains only the search words and the positive feedback terms.

The machine learning algorithm uses the following relevancy features:

- Relevancy score
- Tweet content contains a query term
- Tweet content contains a positive feedback term
- Tweet content contains a negative feedback term
- Username equals the query
- Username contains a negative feedback term

2.3 Tweet Features

The last group of features are mostly general tweet features. Similar features have been used in [3].

- Length of tweet in characters

– Tweet contains a link
– Tweet contains a link to the domain of the entity
– Tweet is a direct message, i.e. starts with a username
– Tweet is a retweet
– Tweet is question

2.4 Sentiment Features

To determine the sentiment of a tweet we make use of the scores generated by the SentiStrength algorithm [2]. SentiStrength generates sentiment scores based on predefined list of words and punctuation with associated positive or negative term weights. The rated training tweets are used to add words to the dictionary and to optimize term weights. Each language has its own lists of words. The tweets are scanned and and all words bearing sentiment, negating words, words boosting sentiment, question words, slang and emoticons are tagged. Using positive and negative weights which can be optimized using training data, the classification of a tweet is determined.

All together we now have 21 features we can use to determine relevancy and polarity for reputation of tweets. We use the data mining tool Weka [4] for training and testing the classifiers.

3 Experiments

The profiling task consists of two separate sub tasks: filtering and determining the polarity for reputation.

3.1 Experimental Set-Up

We have created and submitted 5 runs to the RepLab workshop. For each run we made three choices of options:

1. The Machine Learning classifier to use: Naive Bayes (NB), Support Vector Machine (SVM) or Decision Tree (J48).
2. Which attributes to use as features for the classifier. To determine the relevancy we always user all attributes as described in the previous section. To determine polarity for reputation we use either all attributes, or only the SentiStrength scores.
3. Whether to train two separate classifiers for English tweets and Spanish tweets (separate) or train only one classifier that handles both Spanish and English tweets (merged).

Table 1 shows which options are used in the 5 submitted runs.

Table 1. Runs and Overall Results Profiling

Run ID	Classifier	Attributes	Languages	Accuracy
OXY_1	SVM	All	Merged	0.387
OXY_2	NB	All	Merged	**0.412**
OXY_3	J48	All	Merged	0.346
OXY_4	J48	SentiStrength	Merged	0.359
OXY_5	J48	All	Separate	0.344

3.2 Overall Results Profiling

The overall results of the profiling task combining filtering and polarity for reputation are presented in Table 1. Our run OXY_2, using a Naive Bayes classifier and using all possible features in a single classifier for both English and Spanish tweets performs best. On average over the topics, 41.2% of the tweets is annotated correctly. From all submitted runs to the workshop this is the best score. In the remainder of this section we take a closer look at the results of the two subtasks: filtering and polarity.

3.3 Results Filtering

The results of the filtering task are shown in table 2. Our best run, OXY_2 is the Naive Bayes classifier using all of the attributes described in Section 2. Comparing the scores to the overall workshop results, we see that all our runs outperform all other runs when we look at the measure of accuracy, but we score mediocre looking at the F(R,S)-filtering measure. The reason for this is that the F(R,S)-filtering measure does not reward.

There are some Twitter specific issues that we take into account:

- The Twitter search returns not only results where the search terms are found in the content of the tweet, but also where the search terms are found as a part of the username, or in an external link. We do not use the information contained in the external links. To calculate our relevancy score we only take into account the content of the tweet. In fact, if the search term occurs in the username, this can mean three things:
  1. The Twitter account is owned by the company, e.g. 'Lufthansa_USA. Tweets from these account can all be considered relevant.
  2. The company name is used in a different context, it could be for example also a last name, e.g. 'dave_gillette'. Tweets from these accounts mostly non-relevant.
  3. The username is referring to the company name because he or she likes to be associated with the company, e.g. 'Mr_Bmw'. The number of relevant tweets from these accounts varies, some only tweet about the company they are named after, others hardly do.

Table 2. Filtering results

Run ID	Accuracy	R-Filtering	S-Filtering	F(R,S)-Filtering
OXY_1	0.798	0.217	0.174	0.139
OXY_2	**0.809**	**0.235**	**0.272**	**0.197**
OXY_3	0.790	0.198	0.128	0.096
OXY_4	0.790	0.198	0.128	0.096
OXY_5	0.778	0.157	0.111	0.085

Table 3. Polarity results

Run ID	Accuracy	R-Filtering	S-Filtering	F(R,S)-Filtering
OXY_1	0.368	0.236	0.221	0.219
OXY_2	0.358	0.279	**0.268**	0.245
OXY_3	**0.381**	0.269	0.178	0.194
OXY_4	0.347	**0.290**	0.252	**0.256**
OXY_5	0.375	0.288	0.198	0.211

It is relatively easy to manually retrieve the official Twitter accounts for a company, and regard their tweets as relevant. It is harder to distinguish between the other two cases, but luckily they can be treated in the same way. That is, do not regard the occurrence of the company name in the username on its own as a signal for relevancy, but check whether the content of the tweet also contains relevant terms.
For our submitted runs we tried to separate relevant and non-relevant Twitter accounts by adding the usernames to the positive and negative relevance feedback. Although these are high quality indicators of relevance, their coverage in the dataset is small. Therefore for example we do not see them in the generated decision trees.

3.4 Results Polarity for Reputation

The results of the task to determine polarity for reputation can be found in Table 3. There is not one run that performs best at all evaluation measures. The run OXY_3 performs best looking at accuracy with an accuracy of 0.381, but run OXY_4 performs best looking at the F(R,S)-filtering score with a score of 0.256. In general the scores are not very impressive. Classifying all tweets as positive results in an accuracy of 0.438, which is a better accuracy than any of our runs.

An advantage of the SentiStrength approach is that we can see why a tweet is classified into a certain sentiment. Let's look at some examples. The following tweet is correctly tagged as positive:
'@NIVEA_Australia love your products thanks[2]for following :) [1 emoticon] [sentence: 2,-1]'

The word 'thanks' is recognized as a positive term, as well as the happy emoticon :). The word 'love' however is not tagged as positive.

Incorrectly classified as negative due to the term 'swear' is the following tweet:
'Best[2]smelling body wash known to man ,I swear[-2]!!!![-0.6 punctuation emphasis][sentence: 2,-3] #apricot #nivea http://instagr.am/p/Jv5o0UG8Gy [sentence: 1,-1]'

A problem is that words have different meanings in different context. Apparently in our training data the word 'love' occurs in both positive and negative tweets, since it was not tagged as positive. A bigger training set might solve part of this problem. When we train the classifier on the test data, the word 'love' gets a positive value of 3.

Another problem occurs when also other companies or products are mentioned in the tweet, e.g. in the tweet:
'This boots hand cream is just rubbish[-3][–1 booster word]!![-0.6 punctuation emphasis][sentence: 1,-3] Gonna buy my nivea back today mchew [sentence: 1,-1]'
At the moment we only calculate polarity over the whole tweet. When other entities occur in the tweet, polarity should only be calculated over the part of the tweet that deals with the entity that is the topic of the search.

As we discussed earlier, a problem for our classifier is the lack of training data. First of all, the amount of training data is small. Secondly, the companies in the training and test set are different. When our classifier is trained on a sufficiently large dataset for a company or a industry such as the automotive industry, and tested with data from the same company or industry, better results will be obtained. The best submitted run in the workshop attains an accuracy of 0.487, so we can conclude that classifying tweets on polarity of reputation is indeed a difficult task.

4 Conclusions

The tasks in the RepLab workshop allow us to work on some relatively new and challenging problems, i.e. how to detect the relevancy and polarity of reputation of tweets. In our submission we make use of manually selected feedback terms. In contrast to web search tasks it is more likely we will be able to obtain manual feedback terms from users since the queries remain the same for an extended period of time. The effort of providing feedback terms is paid back by receiving a better set of search results.

We make use of a machine learning approach to predict the relevancy and polarity of tweets. We include query dependent features, relevancy features, tweet features, and sentiment features. The features are language independent, i.e. the same 22 features are calculated for English and Spanish tweets.

Our approach works very well to predict the relevancy of tweets. An average accuracy of 80.9% is achieved using a Naive Bayes classifier. Predicting the polarity of reputation turns out to be a much harder task. Our best run achieves

an accuracy of 38.1% using the J48 decision tree classifier. One of the main problems with our machine learning approach is the limited amount of training data. When the training and test data would contain the same companies, the results would already improve.

The best run of the profiling task, that is the combination of the filtering and polarity task is achieved by our run that uses a Naive Bayes classifier which uses all possible features in a single classifier for both English and Spanish tweets. It reaches an accuracy of 41.2%, which is the highest accuracy of all official RepLab submissions.

References

1. R. Kaptein. Using Wordclouds to Navigate and Summarize Twitter Search Results . In *The 2nd European Workshop on Human-Computer Interaction and Information Retrieval (EuroHCIR)*, 2012.
2. M. Thelwall, K. Buckley, and G. Paltoglou. Sentiment strength detection for the social web. *JASIST*, 63(1):163–173, 2012.
3. M. Tsagkias and K. Balog. The University of Amsterdam at WePS3. In M. Braschler, D. Harman, and E. Pianta, editors, *CLEF (Notebook Papers/LABs/Workshops)*, 2010.
4. I. H. Witten, E. Frank, and M. A. Hall. *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann, Amsterdam, 3. edition, 2011.
5. M. Yoshida, S. Matsushima, S. Ono, I. Sato, and H. Nakagawa. ITC-UT: Tweet Categorization by Query Categorization for On-line Reputation Management. In M. Braschler, D. Harman, and E. Pianta, editors, *CLEF (Notebook Papers/LABs/Workshops)*, 2010.