Determining the Polarity of Postings for Discussion Search

Ingo Frommholz and **Marc Lechtenfeld**
Faculty of Engineering Sciences
University of Duisburg-Essen, Germany
{ingo.frommholz|marc.lechtenfeld}@uni-due.de

Abstract

When performing discussion search it might be desirable to consider non-topical measures like the number of positive and negative replies to a posting, for instance as one possible indicator for the trustworthiness of a comment. Systems like POLAR are able to integrate such values into the retrieval function. To automatically detect the polarity of postings, they need to be classified into positive and negative ones w.r.t. the comment or document they are annotating. We present a machine learning approach for polarity detection which is based on Support Vector Machines. We discuss and identify appropriate term and context features. Experiments with ZDNet News show that an accuracy of around 79%-80% can be achieved for automatically classifying comments according to their polarity.

1 Introduction

Discussion search has more and more come into focus of information retrieval research, for example as a task in the TREC Enterprise Track (see, e.g., [Craswell et al., 2005]). Discussion search can be applied in many scenarios – it can be used for Enterprise search within a company's mailing lists, but also in open forums where people discuss about various topics. The discussion can be restricted to a closed community of scientists and experts on the one hand to a public portal where everyone can participate on the other hand. An example for the latter is ZDNet News[1], where articles related to IT business are published. Readers of these articles can discuss the corresponding topics, and comments can be annotated again. Such discussions naturally contain additional information and different viewpoints on the source article. It is thus clear that discussion threads are an interesting source for new information which can be revealed by means of discussion search.

Many discussion search approaches do not only regard a comment as an atomic item, but also take into account the context coming from the surrounding thread structure, which has shown to be beneficial in various publications (e.g., [Frommholz and Fuhr, 2006a]). However, one problem is usually not addressed: is a comment really useful? One important aspect of "usefulness" is the question whether the content of a comment is *trustworthy*. Especially in public discussion forums, users sometimes give wrong advise or even write plain nonsense, so these comments cannot be trusted – they should not be presented to the user at all. One way to detect such comments is to look at their replies. Comments can be rejected by expressing disagreement on the content level ("I disagree...", "I don't think so...") or by expressing an opinion about the author on the meta level (a typical example here are replies like "don't feed the trolls!"; authors whose goal is simple provocation are regarded as "trolls" in many communities). The underlying assumption is: the higher the ratio between negative and positive replies, the lower the trustworthiness of the according comment (and vice versa). Therefore it would be desirable to detect the *polarity* (positive or negative) of a reply.

But how can we detect whether a reply is positive or negative? Recently, an interesting new field has emerged, which is *sentiment classification* [Pang et al., 2002]. The goal of sentiment classification is to determine whether a user has a positive or a negative attitude towards a certain entity. Sentiment classification can be applied, for instance, to film critics or product reviews to find out the sentiment conveyed in these texts w.r.t. the reviewed entities. In a discussion search scenario, we are interested in the judgements of people w.r.t. to the acceptability of a certain posting, expressed in the replies to a specific comment. This makes sentiment classification a similar, but also slightly different problem to the one of determining the polarity of postings. It is similar, since its basic idea is to perform a non-topical classification into positive and negative; it is different, since we can make use of the thread structure and apply different features than for sentiment classification. Therefore, based on the basic idea of sentiment classification, the goal of this paper is to introduce a machine learning approach which tries to determine the polarity of replies by categorising them into positive or negative ones.

The paper is structured as follows. First, we briefly examine related work. Then, we give a motivating example of the application of the polarity of postings in discussion search, using the POLAR framework. Since our machine learning approach is based on Support Vector Machines, we discuss possible features which can be extracted from comments and discussion threads in the subsequent section. These features are then applied for the evaluation of our approach, which is discussed in Section 5 and gives a first answer to the question how well automatic methods to determine the polarity of replies can perform. The last section concludes the findings.

2 Related Work

Our work is embedded in the field of non-content-oriented text classification and is similar to sentiment classification, which is sometimes also called *sentiment analysis* or *opin-*

[1] http://news.zdnet.com/

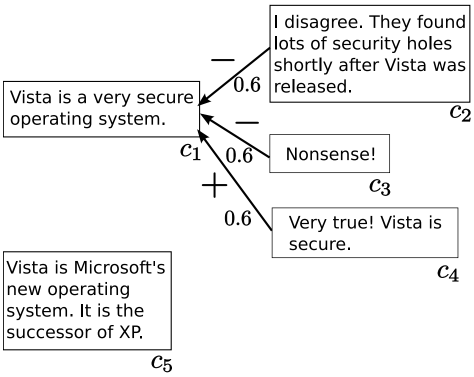
I disagree. They found lots of security holes shortly after Vista was released.
c_2
−
0.6
Vista is a very secure operating system.
c_1
−
0.6
Nonsense!
c_3
+
0.6
Very true! Vista is secure.
c_4
Vista is Microsoft's new operating system. It is the successor of XP.
c_5

Figure 1: Two discussion threads

ion mining, among others.

In contrast to the work of [Hatzivassiloglou and McKeown, 1997], [Kim and Hovy, 2004], [Kamps et al., 2004] and [Esuli and Sebastiani, 2005], for example, our method as well as the following sentiment classification approaches try to classify whole documents instead of single words or phrases.

Turney [Turney, 2002] uses an unsupervised learning algorithm to categorise reviews. He determines the semantic orientation of single phrases with the help of search engines and uses the average semantic orientation of the extracted phrases to classify a document as *recommended* or *not recommended.* Pang *et al.* [Pang et al., 2002] use supervised machine learning techniques to determine whether a movie review is positive or negative.

A domain which is more similar to ours is investigated in [Gamon, 2004]. Gamon uses Support Vector Machines to classify customer feedback data that is less coherent, shorter and less well-formed than reviews that are often written by professional writers.

3 Applying Polarity in Discussion Search – A Motivating Example

In this section, we give an example of how the information about the polarity of comments can be exploited for information retrieval. Since the example is modelled within the POLAR framework, it will be briefly introduced first.

3.1 The POLAR Framework

POLAR (Probabilistic Object-oriented Logics for Annotation-based Retrieval) [Frommholz and Fuhr, 2006b; Frommholz, 2008] is a framework supporting document and discussion search based on user annotations (e.g. comments). Within POLAR, it is possible to model discussion threads, to exploit the thread context for retrieval and, being a logic-based framework, easily integrate additional evidence into the retrieval function. POLAR also provides means to specify whether a reply is positive or negative w.r.t. the object it belongs to.

We introduce the relevant parts of POLAR by giving an example which should also motivate the usage of the posting polarity for retrieval. Consider the situation illustrated in Figure 1. In this figure, we can see two fictitious discussion threads. The first thread consists of the comments $c_1, \ldots, c_4$, the second one only of c_5. c_2, c_3 and c_4 are replies to c_1; let us assume we determined that c_2 and c_4 are negative comments w.r.t. c_1, and c_4 is a positive comment supporting c_1 (this is denoted by the “$+$” and “$-$” signs). After common procedures like stopword elimination and indexing, above scenario can be modeled in POLAR as follows (we omit terms which are not important for the further considerations):

```
1  c1[ 0.7 vista ...
2      0.6 -*c2  0.6 -*c3  0.6 *c4 ]
3  c2[ 0.6 vista ...]
4  c3[...]   c4 [ 0.8 vista ...]
5  c5[ 0.45 vista ... ]
6  0.5 °vista
```

The first two lines model the comment c_1. Line 1 says that the term ‘vista’ appears with the weight 0.7 (based on the term frequency (*tf*)) in c_1. The second line shows which comments reply to c_1. These are c_2, c_3 and c_4. Each reply is accessed with a probability of 0.6; we need this value later as a propagation factor. While, e.g., “`*c4`” says that c_4 is a positive reply to c_1, “`-*c2`” (and “`-*c3`“) means that c_2 (c_3, respectively) is a negative reply to c_1. Line 3 models c_2; ‘vista’ has a term weight of 0.6 here. The fourth line models c_3 and c_4 (with 0.8 being the term weight of ‘vista’ in c_4), respectively, and line 5 represents c_5. The last line encodes the termspace value of ‘vista’, which can be based, for instance, on its inverse document frequency. So we interpret 0.5 as the *idf* value of ‘vista’.

3.2 Discussion Search

Once discussion threads are modelled in POLAR as described above, they can be queried in several ways. For example, POLAR offers a retrieval function which estimates the probability $P(d \rightarrow q)$ that a document d implies the query q [van Rijsbergen, 1986]. Let us assume we want to retrieve comments about ‘vista’. We can express this in POLAR as:

```
1  q[ vista ]
2  ?- D->q
```

The first line models the query q, while the second line (the POLAR query, introduced by ”`?-`“) returns a ranking of comments based on the implication probability. Let us further assume that the estimation of $P(d \rightarrow q)$ is simply based on the *tf* × *idf* value of the query terms. In this case, the POLAR query returns

```
0.350 c1      # 0.5 * 0.7
0.300 c2      # 0.5 * 0.6
0.225 c5      # 0.5 * 0.45
```

For example, the retrieval status value of c_1 is the term frequency of ‘vista’ in c_1, which is 0.7, multiplied with the *idf* value of ‘vista’, which is 0.5. The other values are calculated analogously. (”#“ starts a comment.)

So far no magic is involved. We now present a simple retrieval strategy implemented in POLAR which shall illustrate the role of replies and their polarity.

Annotation-based Trustworthiness

In the following approach, the ratio between positive and negative replies is used as a simple measure for the trustworthiness of a comment. We introduce the following rules:

```
1  0.5 uncond_trust(c1)
2  0.5 uncond_trust(c2)
3  0.5 uncond_trust(c3)
4  0.5 uncond_trust(c4)
5  0.5 uncond_trust(c5)
6  trust(C) :- uncond_trust(C)
7  trust(C) :- C[*A] & trust(A)
8  !trust(C) :- C[-*A] & trust(A)
```

The first 5 lines define the unconditional, a priori degree of trust every comment has (even without replies). In this case, the probability that a comment is trustworthy is 0.5. Line 6 says that a comment is trustworthy if we trust it unconditionally. The last two lines are interesting, as they bias the unconditional trust according to the number, access probability and trustworthiness of positive and negative comments. Each positive reply raises the probability that we trust the corresponding comments (line 7), while the last line means that the probability that we do *not* trust (indicated by the ' ! ' in the rule head) a comment is raised with every negative reply[2]. In our example, the derived probability that we trust c_2, c_3, c_4 and c_5 is their unconditional trust value of 0.5. The trust value of c_1 is based on its unconditional trust and biased by the two negative and one positive replies. The probability that we can trust c_1 is $0.5 + 0.6 \cdot 0.5 - 0.5 \cdot 0.6 \cdot 0.5 = 0.65$ (taking the unconditional trust value of c_1 and the product of the access probability and trust value of c_4; these probabilities are combined with the inclusion-exclusion formula known from probability theory). The probability that we cannot trust c_1 is $0.6 \cdot 0.5 + 0.6 \cdot 0.5 - 0.6 \cdot 0.5 \cdot 0.6 \cdot 0.5 = 0.51$ (based on the trust values of c_2 and c_3). The final probability is calculated as the probability that we have positive and not negative evidence for the trustworthiness, so we get $0.65 \cdot (1 - 0.51) = 0.3185$ for c_1.

The trustworthiness value can be combined with the topical relevance:

```
?- D->q & trust(D)
 0.1500 (c2)  # 0.5 * 0.6 * 0.5
 0.1125 (c5)  # 0.5 * 0.45 * 0.5
 0.1115 (c1)  # 0.5 * 0.7 * 0.3185
```

c_1, although topically the most relevant comment, is now ranked on the last position due to its low trustworthiness value.

Measuring the trustworthiness as above is a simple example that illustrates how to beneficially integrate the polarity of replies into a retrieval function. There exist more elaborated approaches to measure the trustworthiness. For instance, [Cabanac et al., 2007] propose a framework for evaluating the social validation of collaborative annotations, and positive (confirmation) and negative (refutation) judgements found in replies are part of this framework. But without knowledge about the polarity of comments, all these approaches would not work. Unfortunately, we usually do not know the polarity of replies, since it normally cannot be specified in today's forum systems. The question thus is how we can determine the polarity automatically. Another question is how well such a method will perform – does it make sense to try to determine the polarity automatically, or is this task too challenging? To give first answers to these questions, a machine learning approach, based on Support Vector Machines, and its evaluation are presented in the next sections.

4 Machine Learning for Polarity Detection

In the following, we present our machine learning approach for determining the polarity of replies. We describe the collection our considerations are based upon, present the classifier we have chosen and discuss the features we have utilised to classify a posting with reference to its polarity. In the following section, we present and discuss the results of our experiments. A more thorough discussion on the approach, the features and the experiments can be found in [Lechtenfeld, 2007].

4.1 Collection

For the feature definition and evaluation, we used a snapshot of ZDNet News, which was harvested from December 2004 to July 2005. The snapshot consists of 4,704 articles and 91,617 comments. Only the comments were used for our experiments. Besides the comment itself, the ZDNet News discussion functions allow for the specification of a comment title. With each comment, metadata like the creator of the comment as well as the creation date are recorded. Each article is tagged by the ZDNet editors with one or more categories, indicating the topics an article is about (e.g., "Linux" or "Windows"). One of the inherent features of ZDNet is that we find many controversial discussions there, which contain many potential negative replies.

4.2 Automatic Classification

The aim of this work is to automatically determine the polarity of a given posting. We utilised a supervised machine learning approach with the help of a *Support Vector Machine* (SVM), which gave very good results for both *topic classification* [Joachims, 1998; Sebastiani, 2002] and sentiment classification [Pang et al., 2002].

We used the linear kernel, since most text categorisation problems are linearly separable [Joachims, 1998] and it was successfully used in a related noisy domain [Gamon, 2004]. The optimal C-Parameter depends on the specific problem and has to be determined empirically [Hsu et al., 2003].

4.3 Feature Extraction

As document representation we used two large feature sets: *term features*, which represent the text of the posting on the one hand, and *context features* containing information about the thread context of a posting on the other hand.

Term Features

Term features are grouped into four sets: token features, sentiment features, source features and posting features. The groups and their corresponding features are shown in Table 1.

Group	Features
Token	Presence of tokens (unigrams), punctuation
Sentiment	Emoticon (EMOTICON_), negation (NOT_)
Source	Title terms (TITLE_), quotation terms (QUOT_), terms of the previous posting (PREV_)
Posting	Posting length

Table 1: Term features

Token A document is represented as a sets of words in conjunction with unigrams. This representation, which has proven to be successfull for sentiment classification with machine learning techniques [Pang et al., 2002], indicates for each token whether it occurs in the given posting or not.

[2] Without going into detail here, POLAR is based on four-valued logics and an open world assumption, so the probabilities of a proposition a and $\neg a$ are independent of each other (see also [Frommholz, 2008]).

The text was tokenised with the help of white space characters, punctuation marks and special characters. Upper and lower case was ignored. Punctuation marks have been treated as separate term features because they can be seen as quasi-words with its own meaning [Hess, 2006], which can communicate different speech acts and have proven themselves as beneficial to determine the polarity [Pang et al., 2002].

Stop words, which are frequently domain dependent, were not removed, since they can be important indicators for the class affiliation in the examined domain [Forman, 2002] and play an important role in sentiment classification (for example in the investigated domain in [Pang et al., 2002]). Likewise, no stemming was performed, since it can make a big difference for text classification whether nouns are used in the singular or in the plural or negations and prepositions are present or not (as stated in [Riloff, 1995]).

Sentiment Emoticons can express the frame of mind explicitly and therefore they can supply a good reference on the polarity of a posting. Different representations of one emoticon (like ':-)' and ':)') are treated as a single emoticon feature (e.g. `EMOTICON_Smile` feature).

The polarity of a sentence can be inverted by the use of only one word, for example by the word 'not' or other kinds of negation. However, by using the set of words representation in conjunction with unigrams the context of a term gets lost. In order to consider negation, the technique proposed by Das and Chen [Das and Chen, 2001] is applied, which adds the tag `NOT_` to all tokens from the negation token (e.g. 'not') to the first punctuation mark.

Source The title of a posting can be used to concisely summarise the content of the discussion entry. So all tokens of the title were tagged with the prefix `TITLE_`.

Parts of a comment can be quotations of passages of the replied comment. We also experimented with tagging these quotation terms with the prefix `QUOT_`, since quotations can give us information about the annotated object. Considering that ZDNet does not support quotations directly and the resulting noisy domain, we used the Longest Common Substring [Gusfield, 1997] search and some simple correction rules to identify quotations.

In addition to this, discussion postings are short and fragmentary, and therefore can be understood in many cases only with the help of the context consisting of the previous postings. Due to the dependence on the textual context, we also added all the terms of the previous posting tagged with the prefix `PREV_` to the term feature set.

Posting A feature that describes the posting as a whole is the length of the posting text. This feature may give us a hint about the polarity, because the average length of a positive posting may be shorter than the length of a negative posting, since one would expect reasons or examples for the different opinion. This assumption is also expressed in [Yih et al., 2004].

Context Features

The following features, which are summarised in Table 2, are based on the fact that the investigated posting is always part of a discussion thread. With the help of these features, we try to benefit from the relationships between the given posting and the other postings of the thread.

Group	Features
Reference	Do hyperlinks exist? Is the previous author mentioned? Is it an article annotation?
Structure	Do quotations exist? Number of quotations; Ratio of quotations length to previous posting length; Ratio of quotations length to posting length; Are quotations and annotations alternating? Is the author sequence A-B-A or A-A present?
Response Behaviour	Number of direct/total answers; Number of answers to previous posting; Response time within a specific time? Response duration within a specific time? Response creation time (day of the week, hour, time of the day)
Topic	Topic terms describing the article (Tag `TOPIC_`

Table 2: Context features

Reference *References* to other postings or external sources can be indicators for the polarity of the given posting. Hyperlinks can refer to (neutral) background information or email addresses can hint at a (neutral) information request or an information transfer. So a binary feature indicates whether such a reference is used in the posting.

Internal references to other postings of the thread can show that the given posting actually relates to the previous discussion entry. Such a reference can be done, for example, by mentioning the author of another posting. Therefore, a binary feature indicates whether the name of the author of the previous posting is mentioned in the given comment.

Regarding the whole collection, it can be observed that comments answering directly to an article (and not to another comment) are often neutral, because they annotate only one aspect of the article or the overall topic and not the article itself. So we use a binary feature that indicates whether the document the posting is replying to is an article or not.

Structure *Quotations* of other entries are another kind of reference. Because a quotation is often conflicting [Agrawal et al., 2003], this kind of reference can be an indicator for a negative polarity. A binary feature specifies whether the posting contains a quotation and a numerical feature counts how many quotations there are in the comment.

The *ratio of the quoted text to the whole previous text* can give a clue whether the investigated posting annotates only single statements or, for example, whether it refers to the previous comment as a whole. The *ratio of the quoted text to the full given text* can give a clue whether the investigated posting mainly refers to other discussion entries, or if it perhaps contains common explanations or if it is a statement about the overall topic and therefore is neutral.

Another binary feature measures whether *quotations alternate with replying statements* in the given posting, because this could be a indication of a (contrary) opinion.

There can also be dispute on the thread level, for example when indicated by the *author sequence* A-B-A. That means that author B replies to the posting of author A and then author A writes an answer to the posting of author B, and so on. That sequence may indicate that the last entry of author A is a contrary (negative) comment of a disputation with author B. Hence, two binary features verify whether the two previous postings exist, and a binary feature indicates the existence of the author sequence A-B-A. Another binary feature verifies, whether the author sequence A-A exists. If an author writes an answer to his or her own posting, this posting is most likely a neutral additional note or a neutral try to call attention to the (perhaps yet unanswered) posting.

Response Behaviour Two numerical features are the *number of direct and total answers* to the given posting. They could be a hint about the polarity because provoking postings are probably annotated more often than neutral statements, which perhaps will not be answered at all. The number of replies that are written to the previous posting are counted by an additional numerical feature since this could indicate that the given posting is one of many negative answers.

The *response times* of replies could also be indicators for their polarity, since an answer to a wrong or even provoking comment is probably published faster. We estimate the response time with the time between the creation of the original posting and the creation of the answer, since we cannot measure the real reaction time (this would require knowledge about the time a user reads the comment). Binary features are used for a specified number of hours after the posting was added with an increasing length of a specific number of minutes. For example, a binary feature reflects if a reply was created within the first 15 minutes, another one if it was created within the first 30 minutes, and so on in 15-minute-steps.

Conclusions might be drawn from the *activity duration* within which all replies have been created. Controversial postings are perhaps annotated more often and faster than for example questions that are acceptably answered with a few postings. Therefore a binary feature indicates whether all responses are created within a specific number of hours, here 12 hours.

Also the *creation time* of a reply could contain helpful information. On certain days of the week or times of the day there are particularly many negative postings, for example in the form of provocations. We used one binary feature for every day of the week, which states whether the given posting is created on the respective day of the week. In addition we used a specified number of binary features for a specific period of time, for example to consider the time or the hour of the day.

Topic The *topic* of the first posting (here the ZDNet article), which initiates the discussion, could also be a good indicator for the polarity of a given posting, because it specifies the topic of the following discussion. And postings in threads about a polarising topic are probably contrary to a greater extent. So we used the ZDNet tags, which can be assigned to any news article by ZDNet as an additional term-like feature.

5 Experiments and Results

In this section we describe the experiments we have done to determine the performance of the suggested approach. After the evaluation data is described and the evaluation criterion and measure are mentioned, the various experiments are explained, which evaluate the performance of the term and context features and determine in how far the different feature groups contribute to the measured accuracies.

5.1 Testbed Creation

We used the ZDNet collection introduced in the previous section for our experiments. To create the testbed, we asked colleagues, students and friends working in IT business to manually classify comments whether their overall sentiment is positive or negative w.r.t. the comment they reply to. Furthermore, we also introduced a class *Neutral*. The three classes were defined as follows:

Positive A comment should be judged positive, if it *explicitly* talks positive about the comment it replies to, e.g. by identifying phrases like "good idea", "I agree" or "That helped me much".

Negative In negative comments, we *explicitly* find evidence that the author thinks negatively about the previous comment. This negative evidence manifests itself in phrases like "I disagree", "I don't think so", "That's not true", etc.

Neutral Comments should be judged neutral, if there can not be found any *explicit* evidence that it is positive or negative.

Neutral comments play a special role later in the classification, although we do not use this class for the final categorisation and application in POLAR. Instead, we assume that neutral comments are implicitly positive. For example, if the author of a comment just provides some background information (for instance, a comment "It's Paris in Texas, not in France" would add some background information to the content of a previous comment which talks about Paris), we would have a neutral comment which is implicitly positive, as there is no disagreement or counterargument in such a comment.

Multiple classification was possible, e.g. if a comment was half positive and half negative, both a positive and negative category could be chosen (resulting in two judgements for this comment). 173 positive, 637 neutral and 431 negative judgements were made. This results in 1,241 judgements given in total by the 10 assessors. A posting could be judged as positive and negative simultaneously. Such postings have a mixed polarity and therefore an ambiguous overall polarity. To train a classifier for the task of classifying postings into positive or negative comments as good as possible, training examples are needed that are representative examples for the categories to learn. Therefore we reclassify such postings as neutral. If we also consider that each category should be uniformly distributed, the available instances that can be used for the evaluation are limited to the number of instances associated to the smaller category. So the available instances for training and testing count 300 examples - 150 positive and 150 negative.

5.2 Evaluation Criterion and Measure

To verify the effectiveness of the presented approach, we performed stratified ten-fold cross-validations for all variants of the experiments. Thus the available example set was randomly divided into ten equal-sized subsets. In each of the ten validation iterations, another subset was used to validate the model that was derived from the remaining nine (training) subsets. The overall performance was estimated by averaging the performance of the ten validations. As evaluation measure we took the classification *accuracy*, which is defined as the number of the correctly classified instances divided by the total number of instances. For our experiments we used uniformly distributed training and test sets. Therefore the random-choice baseline to beat is 0.5.

5.3 Term Features

Table 3 summarises the results of the experiments, which were done to determine the accuracy of a classifier that uses only the content of a given posting to classify it with reference to its polarity.

Experiment	Unigrams	Bigrams
Text classification	0.6367	0.63
Feature selection	0.69	0.6867
Tagged title terms	0.6533	0.6677
Removed quotations	0.6567	0.6833
Tagged quotations	0.6733	0.68
Previous comment	**0.76**	**0.7533**

Table 3: Accuracies of some term feature experiments

In the first experiments we used the term feature groups *Token*, *Sentiment* and *Posting* (described in Section 4.3) to do a common sentiment classification without considering the specifics of the investigated domain of discussions (line 1 in Table 3). To determine a C kernel parameter that is optimal for the examined classification task and domain, we tried different power-of-two numbers among others and reached the best results with unigrams[3] of 0.6367 (using a C value of 0.5).

The next experiments regard that the noisy domain of online discussion entries with the large number of typing errors and spelling mistakes could complicate the text classification. As demonstrated in [Gamon, 2004] a sentiment classification in a noisy domain can nevertheless be performed, if feature reduction techniques are used in combination with large initial feature vectors. Consequently we tried the following two kinds of feature reduction to improve the best accuracy achieved in the first experiments. We applied the chi-square test with different threshold values[4] and filtered out features with a weak statistical dependence between the occurrence of a feature and the class affiliation. Furthermore we only used features occurring in the document collection at least one, two, or tree times, respectively. This method also has the advantage that (rare) typing errors are removed automatically. The best accuracy of 0.69 was yielded, if all features were used that (occur in the posting at least one time and) have a chi-square value of at least 1.0 (line 2 in Table 3).

In a third step, the sources of the terms were considered by using the *Source* features (see Section 4.3). The results are shown in the lines 3-6 in table 3. Tagging the terms that occur in the title results in a lower accuracy of 0.6533 (compared to line 2). To avoid that quotations of other authors with a potentially contrary polarity could influence the polarity determination of the given posting, we tagged all terms with a prefix that are part of a quotation. If the quotation terms were removed from the feature sets and the parameters for the feature selection were adapted due to the lower number of term features, the accuracy decreases to 0.6567 in the best case. Using the tagged quotation terms resulted in an accuracy of 0.6733 (both compared to l. 2 again). Because a posting potentially has a context-dependent meaning, we also added the terms occurring in the previous entry – tagged with a prefix – to the term feature set. In the case that no feature reduction was done, the best accuracy of 0.76 was measured.

5.4 Context Features

Table 4 summarises the results of the experiments, which were done to determine the accuracy of a classifier that analyses the context of a given posting.

Experiment	Accuracy
Initial context feature set	0.67
Feature selection	0.67
Without references features	0.6733
Without structure features	0.56
Without response features	0.4433
With 4 hour period of time	0.70
With (Article) topic features	**0.7733**
Topic features only	0.74

Table 4: Accuracies of some context feature experiments

Before different values for the feature parameters were inspected, a feature set was used that contains *all context features* shown in Table 2, without the topic feature. Only on the basis of this contextual information, postings can be classified with reference to their polarity with an accuracy of 0.67 (see line 1 of Table 4), if no scaling is performed and the best kernel parameter ($C = 0.5$) is used. Most values for the*feature selection* did not have much influence on the accuracy. Because the classification performance could not be improved by using feature selection (see line 2 of Table 4), the subsequent experiments do not make use of it anymore.

The following experiments investigate how much several context feature groups contribute to the measured accuracy of 0.67. A feature set without the *reference features* does not lead to a lower accuracy, but even to a slightly increased accuracy (0.6733, line 3 of Table 4).

If the *quotation features* of the structure feature group are missing, the accuracy does not decrease (0.67). Likewise, removing the *author sequence* feature A-A has no negative effect on the performance. But the removal of the sequence feature A-B-A results in a decreased accuracy of 0.6533. Without all structure features, the performance decreases to an accuracy of 0.56 (see line 4 of Tab. 4).

If the features are missing that count the *number of direct responses* to the given posting or the *total number of responses* to the previous posting, an accuracy of 0.6667, and 0.65, respectively is measured. The accuracy decreases to 0.67, if not only the direct responses are considered, but also all postings that are written to answer one of the following comments. Removing the binary features created to measure the *response time* of one reply results in a lower accuracy of 0.59. An accuracy of 0.65 can be measured, if the *activity duration* feature is missing that verifies whether all responses are written within the first 12 hours. The removal of the day-of-the-week features of the *creation time* feature group leads to a lower performance of 0.6467. Without the 24 hour features that state whether the posting was created in the specific hour of the day the accuracy decreases to 0.6567. By adding binary features

[3] For detailed information about the results using bigrams see [Lechtenfeld, 2007].

[4] Used threshold values: 0 (no filtering), 0.8, 0.9...1.3.

to the feature set that represent a longer period of time like a timeframe of 4 hours (beginning at 2 o'clock) representing hours of work and closing time, the accuracy increases (from 0.67) to **0.70** (line 6 of Tab. 4).

If the *topic* features that describe the topic of the initial discussion entry (the article) are added to the best features set described so far, the accuracy increases (from 0.70) to **0.7733** (line 7 of Tab. 4), if the C parameter is adapted to 0.125.

5.5 Combination of Term and Context Features

The usage of the term features and the context features led to a similar accuracy of 0.76 and 0.7733, respectively. In the following we examine how well a combined feature set consisting of the best term and context features can be used to classify postings with reference to their polarity. With the new optimal C value (0.03125) and parameter for the feature selection (at least three occurrences) the postings can be classified with an accuracy of **0.79** into positive and negative comments.

5.6 Discussion

In the following some selected results of the experiments described in the last three subsections are discussed.

Term Features The polarity classification of discussion postings based on the comment text can benefit from a feature reduction with chi-square. Likewise the textual context of the given posting seems to play an important role. The additional usage of the (tagged) terms of the previous postings increases the classification performance clearly. With the best accuracy of 0.76, a polarity classification can be done, which is far beyond the random-choice baseline of 0.5.

Context Features Since the reference features were designed to classify neutral postings, it is not surprising that they do not improve the accuracy of a Positive/Negative classifier. Similarly, the author sequence feature A-A does not contribute to the measured accuracy, whereas the sequence feature A-B-A has a larger influence on the performance and improves the accuracy. Interestingly this feature identified more positive postings (e.g. in the form of acknowledgements) than negative ones. Even if also the quotation features are not very important for the task of polarity detection, in total, the structure features improve the performance of the classifier. The features that represent the response behaviour seem to be good indicators for the polarity. While the number of responses do not play an important role for the measured performance, the response time and the creation time contribute to the measured accuracy, so that it decreases to 0.4433, if the response features are not used (see line 5 of Table 4). The additional usage of the topic features increases the accuracy by a large amount. Even with a feature set that only consists of the topic term feature, the polarity of a posting can be determined with an accuracy of 0.74 (see line 8 of Table 4). Overall, a posting can be classified only with the help of the context features with an accuracy of 0.7733, which is also far beyond the random-choice baseline of 0.5. The feature groups representing the response and creation time as well as the article topic are the best indicators for the performance measured in this work.

5.7 Importance of Neutral Postings

In the POLAR framework, we distinguish between positive and negative replies, but not neutral ones. A posting is assumed to be implicitly positive if there is no explicit sentiment expressed (e.g. by phrases like "I disagree"). Such a posting would have been categorised as "neutral" by our manual assessors during testbed creation. In the experiments so far, these postings (almost 50 percent of the manually classified ones) were ignored, but in [Koppel and Schler, 2005] it is pointed out that taking into account neutral instances could also improve the classification into positive and negative. Therefore we decided to exploit neutral instances as well for our task. We used the *Optimal Stack* approach described in [Koppel and Schler, 2005]. The meta classifier consists of the three binary classifiers *Positive/Negative*, *Positive/Neutral* and *Neutral/Negative* (for details see [Lechtenfeld, 2007]). The results of these classifiers are used to perform the final classification into *Positive/Negative*.

As first results show, a classification into the three categories can be done with an accuracy of 0.6711, which clearly surpasses the random-voice baseline of 0.33 for a three category problem. This result was achieved by performing a majority decision (instead of using the optimal stack) and by using the best features and feature extraction parameters for the *binary* classification. Therefore the accuracy of the multi-class classifier could most likely be improved by determining an optimal feature set with its best feature extraction parameters with reference to the three class problem. If the neutral postings are used to train the binary classifier, the classification into positive and negative postings can be improved to the best measured accuracy of **0.80**, if the Optimal Stack uses only the context features.

6 Conclusion and Outlook

The main motivation of the categorisation approach presented here was to show if it is basically possible to categorise replies w.r.t. their polarity. This task is similar to sentiment classification and it turns out to be a more difficult task than classical text categorisation, but we are able to achieve an accuracy of around 79%-80%. The results of our experiments look promising, which also makes the inclusion of measures like the trustworthiness into the retrieval function realistic. The presented experiments were meant to gain a feeling of how well a classification into positive and negative comments could perform. Future work might refine this approach or develop new ones to improve these first results. Possible next steps are reported now.

The performance of the polarity classifier could be improved by additional features that classify the author of a posting. So, it is interesting to know, whether the authors of the given and the previous posting are members of the same opposite camp [Agrawal et al., 2003] or not. Furthermore a classification of a given author as spammer or troll could be helpful to identify neutral (and even not negative) postings. Also the classification of postings as specific speech or dialog acts could be indicators for the polarity of a given posting.

Since postings are not always annotated as a whole and often only comment quoted sections, a more detailed classification of judgements and identification of the annotated objects at the sentence or paragraph level could possibly improve the determination of the usefulness of a posting. So postings with alternating paragraphs of quotations and replies could be seen as a set of separate annotations of partial postings.

In addition, as already stated in Section 5.7, neutral postings could play an important role and thus should be taken into account. Further experiments with a feature set that is more adapted will probably increase the performance of the three-class classification. Generally, a larger quantity of training examples could improve the prediction accuracy of the (binary and three-class) classifiers.

Besides improving the polarity classification, the discussion search example in Section 3 needs to be evaluated. One focus should lie on the question how the categorisation performance affects the retrieval effectiveness.

References

[Agrawal et al., 2003] Agrawal, R., Rajagopalan, S., Srikant, R., and Xu, Y. (2003). Mining newsgroups using networks arising from social behavior. In *Proceedings of the WWW 2003*, pages 529–535, New York, NY, USA. ACM Press.

[Cabanac et al., 2007] Cabanac, G., Chevalier, M., Chrisment, C., and Julien, C. (2007). Collective annotation: Perspectives for information retrieval improvement. In *Proceedings of the RIAO 2007*, Pittsburgh, PA, USA. C.I.D. Paris, France.

[Craswell et al., 2005] Craswell, N., de Vries, A. P., and Soboroff, I. (2005). Overview of the TREC 2005 Enterprise Track. In Voorhees, E. M. and Buckland, L. P., editors, *The Fourteenth Text REtrieval Conference (TREC 2005)*, Gaithersburg, MD, USA. NIST.

[Das and Chen, 2001] Das, S. R. and Chen, M. Y. (2001). Yahoo! for Amazon: Sentiment parsing from small talk on the Web. In *Proceedings of EFA 2001, European Finance Association Annual Conference*, Barcelona, ES.

[Esuli and Sebastiani, 2005] Esuli, A. and Sebastiani, F. (2005). Determining the semantic orientation of terms through gloss classification. In *Proceedings of CIKM 2005*, pages 617–624, Bremen, Germany. ACM Press.

[Forman, 2002] Forman (2002). Choose your words carefully: An empirical study of feature selection metrics for text classification. In *European Conference on Principles of Data Mining and Knowledge Discovery, PKDD, LNCS*, volume 6.

[Frommholz, 2008] Frommholz, I. (2008). *A Probabilistic Framework for Information Modelling and Retrieval Based on User Annotations on Digital Objects*. PhD thesis, University of Duisburg-Essen. Submitted.

[Frommholz and Fuhr, 2006a] Frommholz, I. and Fuhr, N. (2006a). Evaluation of relevance and knowledge augmentation in discussion search. In *Proceeding of the ECDL 2006*, pages 279–290.

[Frommholz and Fuhr, 2006b] Frommholz, I. and Fuhr, N. (2006b). Probabilistic, object-oriented logics for annotation-based retrieval in digital libraries. In *Proceedings of the JCDL 2006*, pages 55–64.

[Gamon, 2004] Gamon, M. (2004). Sentiment classification on customer feedback data: noisy data, large feature vectors, and the role of linguistic analysis. In *Proceeding of COLING 2004*, pages 841–847, Geneva, CH.

[Gusfield, 1997] Gusfield, D. (1997). *Algorithms on Strings, Trees, and Sequences - Computer Science and Computational Biology*. Cambridge University Press.

[Hatzivassiloglou and McKeown, 1997] Hatzivassiloglou, V. and McKeown, K. R. (1997). Predicting the semantic orientation of adjectives. In *Proc. of the 35h Annual Meeting of the ACL and the 8th Conference of the European Chapter of the ACL*, pages 174–181, Madrid, Spain. Association for Computational Linguistics.

[Hess, 2006] Hess, M. (2006). Lerneinheit Tokenisierung. Web-basiertes virtuelles Laboratorium zur Computerlinguistik. `http://www.ifi.unizh.ch/groups/CL/hess/classes/le/token.0.l.pdf`.

[Hsu et al., 2003] Hsu, C.-W., Chang, C.-C., and Lin, C.-J. (2003). A practical guide to support vector classification.

[Joachims, 1998] Joachims, T. (1998). Text categorization with support vector machines: learning with many relevant features. In Nédellec, C. and Rouveirol, C., editors, *Proceedings of ECML 98*, pages 137–142, Heidelberg et al. Springer.

[Kamps et al., 2004] Kamps, J., Marx, M., ort. Mokken, R., and de Rijke, M. (2004). Using WordNet to measure semantic orientation of adjectives. In *Proceedings of LREC-04, 4th International Conference on Language Resources and Evaluation*, volume IV, pages 1115–1118, Lisbon, PT.

[Kim and Hovy, 2004] Kim, S.-M. and Hovy, E. (2004). Determining the sentiment of opinions. In *Proceedings COLING-04, the Conference on Computational Linguistics*, Geneva, CH.

[Koppel and Schler, 2005] Koppel, M. and Schler, J. (2005). The importance of neutral examples for learning sentiment. In *Proceedings of FINEXIN-05, Workshop on the Analysis of Informal and Formal Information Exchange during Negotiations*, Ottawa, CA.

[Lechtenfeld, 2007] Lechtenfeld, M. (2007). Sentiment Classification in Diskussionen. Master's thesis, Universität Duisburg-Essen, FB Ingenieurwissenschaften. In German.

[Pang et al., 2002] Pang, B., Lee, L., and Vaithyanathan, S. (2002). Thumbs up? Sentiment classification using machine learning techniques. In *Proc. of the 2002 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 79–86.

[Riloff, 1995] Riloff, E. (1995). Little words can make a big difference for text classification. In *Proceedings of SIGIR 1995*, pages 130–136. ACM Press.

[Sebastiani, 2002] Sebastiani, F. (2002). Machine learning in automated text categorization. *ACM Computing Surveys*, 34(1):1–47.

[Turney, 2002] Turney, P. (2002). Thumbs up or thumbs down? Semantic orientation applied to unsupervised classification of reviews. In *Proceedings of ACL 2002*, pages 417–424. Association for Computational Linguistics.

[van Rijsbergen, 1986] van Rijsbergen, C. J. (1986). A non-classical logic for information retrieval. *The Computer Journal*, 29(6):481–485.

[Yih et al., 2004] Yih, W., Chang, P., and Kim, W. (2004). Mining online deal forums for hot deals. In *Proceedings of WI-04, the IEEE/WIC/ACM International Conference on Web Intelligence*, pages 384–390. IEEE Computer Society.