Opinion Mining in Newspaper Articles by Entropy-based Word Connections

Thomas Scholz and **Stefan Conrad**
Heinrich-Heine-University
Institute of Computer Science
D-40225 Düsseldorf, Germany
{scholz,conrad}@cs.uni-duesseldorf.de

Abstract

A very valuable piece of information in newspaper articles is the tonality of extracted statements. For the analysis of tonality of newspaper articles either a big human effort is needed, when it is carried out by media analysts, or an automated approach which has to be as accurate as possible for a Media Response Analysis (MRA). To this end, we will compare several state-of-the-art approaches for Opinion Mining in newspaper articles in this paper. Furthermore, we will introduce a new technique to extract entropy-based word connections which identifies the word combinations which create a tonality. In the evaluation, we use two different corpora consisting of news articles, by which we show that the new approach achieves better results than the four state-of-the-art methods.

1 Introduction

The Web keeps many potentially valuable opinions in news articles which are partly new online articles or uploaded print media articles. Many companies or organisations such as political parties or even distinguished public figures perform a Media Response Analysis (MRA) (Watson and Noble, 2007) in order to analyse the output of their effort in public relations. So, an opinion-oriented analysis of news articles is important, because the tonality (Watson and Noble, 2007; Scholz et al., 2012a) is the key indicator of a MRA. A purely manual solution implies a big human effort for so-called media analysts, because they have to read and rate approx. 200 to 800 news articles each week.

As a consequence, an automated Opinion Mining solution is very attractive. At the same time, Opinion Mining in newspaper articles appears to be difficult, because not all parts of news articles are as subjective (Balahur et al., 2010) as reviews, for example. Also, different parts of one article can contain different opinions (Watson and Noble, 2007). Therefore, we work with extracted statements of news articles, in which a sequence of consecutive sentences has the same tonality value. At the same time, some approaches focus more on differentiating only between positive and negative news and leave out neutral examples (Taboada et al., 2011; Scholz et al., 2012b). Conversely, we have noticed that even if the used words in the news domain are quite similar, the tonality which the words express can be different, especially if neutral examples are involved (cf. section 3.1). We propose this task formulation:

Problem definition: Let $s \subseteq d$ be a statement and document d represents a newspaper article. The task is to determine the tonality y for a given statement s, consisting of k words:

$$t : s = (w_1, w_2, ..., w_k) \mapsto \\ y \in \{\text{positive,neutral,negative}\} \tag{1}$$

Normally, a statement consists of one up to four sentences. But also longer statements are possible, but they appear less frequently in a MRA. An automated approach (Scholz and Conrad, 2013) for the extraction of statements already exists. The approach applies machine learning to extract relevant sentences from a collection of news articles and combine them to statements. So, we concentrate on the tonality classification, which is not provided by

Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, pages 1828–1839,
Seattle, Washington, USA, 18-21 October 2013. ©2013 Association for Computational Linguistics

the approach for the statements extraction (Scholz and Conrad, 2013). Furthermore, we define the polarity of sentiment as the distinction between positive and negative sentiment and the subjectivity as the distinction between subjective (positive and negative) statements and neutral statements.

The following example is a positive statement from an article in The Telegraph (8th Aug 2012) which deals with the prospects of British companies in Africa:

- **Example statement (positive):** *There are structural factors behind the African growth story: a growing and sizeable population which is increasingly urbanised with disposable income; growing political stability; and a financial services industry that is still in its infancy.*

The so-called pressrelations dataset (Scholz et al., 2012a), which represents a publicly available corpus[1] of a MRA on German news articles, contains 1,521 annotated statements. Since this is the only publicly available corpus of a MRA as far as we know, we perform our experiments in German. We are aware of the fact that viewpoints play a significant role in a newspaper, but since we concentrate on the determination of the tonality, the extraction of viewpoints can be solved in a separate step (Scholz and Conrad, 2012). This is possible, because the tonality of a statement can be determined without knowledge of the viewpoint in almost all cases. The only exception is a statement with multiple viewpoints and different tonalities, but these statements are very rare (cf. also section 4.1).

Our approach learns a graph from an annotated collection of statements, in which nodes and edges model tonality-bearing word connections. For unseen statements, we recognize subgraphs of the learned graph, compare two weighting methods for extracting different tonality features, and classify the statements by a support vector machine.

In this paper, we describe four state-of-the-art techniques for Opinion Mining in the next section about related work. In the third section, we introduce our graph-based and entropy-based approach to calculate the tonality features T. We will evaluate our approach against the state-of-the-art methods in section 4, before we conclude in the last section.

2 Related Work

Opinion Mining and Sentiment Analysis represent a broad subject area (Pang and Lee, 2008).

The different contributions reach from applying Opinion Mining in reviews and recommending new multimedia products for individuals (Qumsiyeh and Ng, 2012) to sentiment analyses for different topics in social media (Wang et al., 2011) or the creation of sentiment dictionaries (Baccianella et al., 2009). In this paper, we focus on state-of-the-art methods for Opinion Mining which differ from each other in their methodology.

In the news domain, Wilson et al. (2009) developed a word-based classification approach which can extract contextual polarity. This method (denoted as **Wilson**) uses a lexicon and POS-tagging to generate word features and sentence features. Moreover, it also uses deep natural language analyses with dependency parse trees in order to calculate (general and polarity) modification features and structure features. Finally, they compute 32 features for neutral-polar classification and 10 features for the polarity classification. These features can be used by different kinds of machine learning techniques such as Ripper (Cohen, 1996) or BoosTexter (Schapire and Singer, 2000).

Based on a sentiment lexicon, Taboada et al. (2011) calculate the semantic orientation of opinion-bearing words (**SO-CAL**). They begin with a fine-grained dictionary of adjectives, adverbs, verbs, and nouns which have a score from -5 to +5. “Masterpiece” has a score of +5 and “monstrosity” of -5, for example. In addition, the approach takes intensifiers, negations, and irrealis (Taboada et al., 2011) into account and thereby modifies the score of the words through rules and formulas. SO-CAL identifies some special expressions and constructions, which tell the reader, that this text part does not really contain an actual opinion or sentiment. The linguistic term for this situation is called irrealis. Also, text-level features weight the final score by mere presence of the words.

In the field of customer reviews, Ding et al. (2008) also work with a dictionary, which even includes context-dependent words (positive, neutral, and negative words) as well as rules to identify the sentiment orientation of words (**Opinion**

[1] http://www.pressrelations.de/research/

Observer). The rules deal with negations, inter-sentence conjunctions, but-clauses, and the modifier "too". Furthermore, they extract relations between opinion words and corresponding product features. Thereby, a detailed analysis of product reviews is possible.

Sarvabhotla et al. (2011) propose to extract the subjective excerpt of a text (**RSUMM**). They construct two word-vectors: An average document frequency vector represents the most important and most specific word features for the given domain. Subsequently, an average subjective measure vector selects the most subjective terms. As a result, they require hardly any natural language preprocessing except a sentence splitter and a tokenizer. The final classification is accomplished by a SVM (SVM-Light (Joachims, 1999)).

For product reviews, graph-based approaches (Goldberg and Zhu, 2006; Wu et al., 2009) can increase the performance in cross-domain tasks (Ponomareva and Thelwall, 2012). By contrast, our graph nodes do not represent documents, but words to overcome the problem of similar bag-of-words representations (cf. next section).

One important resource for Opinion Mining in news is the MPQA corpus (Wiebe et al., 2005) which contains word and phrase-based annotation for 523 news articles. Unfortunately, since the corpus does not have statements and a statement-based tonality, it is not designed as a MRA. A slightly larger corpus, the pressrelations dataset (Scholz et al., 2012a) with 617 articles, is the result of a MRA in German. We use this corpus as one part of our evaluation.

3 Learning Tonality with Entropy-based Word Connections

3.1 Graph Model for Word Connections

To solve the Opinion Mining task for a MRA, we propose a graph-based approach to capture the opinion-bearing words and modifiers such as negations. In this way, our approach is able to recognize tonality-indicating structures (subgraphs) which provide precise information about the tonality, even if statements have a very similar bag-of-words representation and at the same time different tonalities. One could also say that we create a graph instead of a sentiment dictionary from training examples, as other approaches (Kaji and Kitsuregawa, 2007; Du et al., 2010) proceed.

In figure 1, simple examples are shown with a possible graph (the nodes and edges are taken from the given statements; of course, the graphs and weights become larger in practice). These simple examples are concentrated on nouns, verbs, and adverbs, but also examples with combinations of other categories are possible, such as, for example, different combinations of adjectives, nouns, and verbs: "This is a black day for the company", "The company is in the black", "The company is in the red" and "The company prevents to be in the red". Thus, even though the word representation is quite similar, the tonality can be different.

For opinion-bearing words, we use adjectives, nouns, verbs, and adverbs, which are widely acknowledged as opinion-bearing word categories (Bollegala et al., 2011; Remus et al., 2010; Taboada et al., 2011). Furthermore, we also include negation particles. Therefore, the vocabulary V is the set of words in lemma for one set of statements S. Thus, for every lemma $w \in V$, the approach creates one node v in the graph. A node v also contains the type information (adjective, noun, verb, adverb, or negation).

The edge e_{ij} shows the appearance of node v_i and v_j in combination with tonality y by means of a weight $\varepsilon_{i,j}$ (the sequence of the values in equation 2 is also used in figure 1 and 2).

$$\varepsilon_{ij} = (y_{ij\pi}, y_{ijo}, y_{ij\nu}) \tag{2}$$

$y_{ij\pi}$ is the number of co-occurrences of node v_i and v_j in positive statements within the same sentence. In analogy, y_{ijo} belongs to sentences of neutral statements and $y_{ij\nu}$ to sentences of negative statements. Figure 1 shows a small example for this calculation, too.

3.2 Generating Features for Learning

From a learned graph, we can combine different edges to calculate tonality features for an unseen statement s. An unseen statement is a statement, which is of course not used to learn the graph. We use all edges of the subgraph G_{sl} which contains the nodes for every lemma w_i in the l-th sentence of s.

1)	This solves the crisis.	(positive)
2)	This solves the crisis slowly.	(neutral)
3)	This intensifies the crisis.	(negative)

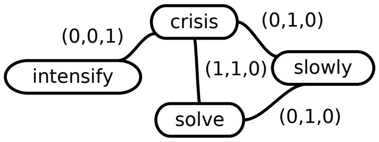

Figure 1: An example for different statements and a graph: The weights base on the three examples and their notation is (positive,neutral,negative).

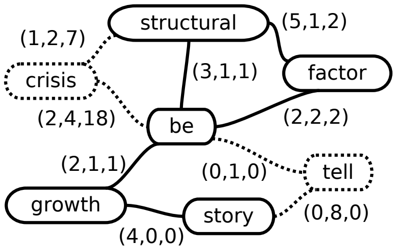

Figure 2: An example of a learned graph: The nodes and edges, which are drawn in solid lines, represent the recognized subgraph G_{sl} for the sentence “There are structural factors behind the African growth story.”.

We explain this with an example. Assuming that our learned graph is shown in figure 2. It contains seven nodes and nine edges (also the nodes and edges in dashed lines). If we further assume that an unseen statement is the example of section 1. To keep this example short, we take the part until the colon as the first sentence of the statement: “There are structural factors behind the African growth story.”

Our approach recognizes the nodes for “be”, “structural”, “factors”, “growth”, and “story”. Thus, the subgraph G_{sl} for the first sentence ($l = 0$) would be the graph which is drawn in solid lines in figure 2. In this example, it is a connected graph, but it does not have to be.

We could also look for complete or connected graphs in the statement instead of using all edges. The largest complete graph would consist of the nodes “structural”, “factor”, and “be” in our example. But using all edges achieves better results, because this method provides all information. In addition, this method is quicker (search for largest complete or connected graph can be omitted, which would be an additional check).

If we have found our subgraphs G_{sl}, we can then compute the vectorial sum of all edges for one node υ_i and we get the probability for a tonality y, if we observe υ_i in the l-th sentence:

$$P(pos|\upsilon_i) = \frac{\sum\limits_{e_{ij} \in Gsl} y_{ij\pi}}{\sum\limits_{e_{ij} \in Gsl} y_{ij\pi} + y_{ij\nu}} \tag{3}$$

$$P(neg|\upsilon_i) = \frac{\sum\limits_{e_{ij} \in Gsl} y_{ij\nu}}{\sum\limits_{e_{ij} \in Gsl} y_{ij\pi} + y_{ij\nu}} \tag{4}$$

$$P(sub|\upsilon_i) = \frac{\sum\limits_{e_{ij} \in Gsl} y_{ij\pi} + y_{ij\nu}}{\sum\limits_{e_{ij} \in Gsl} y_{ij\pi} + y_{ijo} + y_{ij\nu}} \tag{5}$$

$$P(neu|\upsilon_i) = \frac{\sum\limits_{e_{ij} \in Gsl} y_{ijo}}{\sum\limits_{e_{ij} \in Gsl} y_{ij\pi} + y_{ijo} + y_{ij\nu}} \tag{6}$$

For the subjective class (sub), we add the appearance in positive statements ($y_{ij\pi}$) and negative statements ($y_{ij\nu}$). Otherwise we take the appearances in statements of the same class. The denominators of the polarity refer only to positive and negative appearances, while the denominators for the subjectivity refer to every tonality.

By calculating the vectorial sum, we combine several edges in order to estimate precise tonality scores. In this way, we can get the correct tonality score for the noun “crisis”, if a sentence contains also “solve” and “slowly” (→ more neutral) or “intensify” (→ more negative) (cf. figure 1). And we get the correct tonality score for the adjective “structural”, if a sentence includes also “crisis” (→ negative) or the nodes “factor”, “be”, “growth”, and “story” (→ positive) (cf. figure 2).

We distinguish between different word categories (we have noticed that this creates better results than just having a single feature for one statement). Thus, every category gets its own feature and every node only has a tonality value, if it belongs to the category of the feature. This does not mean that we only consider edges which connect two nodes with the same category; we divide the influence of different categories into different features:

$$T_{cat,z}(\upsilon_i) = \begin{cases} f_z(\upsilon_i) & \text{if } \upsilon_i \in cat \\ 0 & \text{if } \upsilon_i \notin cat \end{cases} \quad (7)$$

$cat \in \{adj, adv, n, v\}$ indicates the category of the node (adjectives, adverbs, nouns, or verbs) and z specifies the type of feature. One type shows the difference between positive and negative polarity ($z = pol$), for the other type we replace the positive class by the subjective one (the sum of positive and negative) and the negative by a neutral one in order to differentiate between neutral and non-neutral examples ($z = sub$). As a result, we calculate eight features (see table 1) for the tonality, two for each important word category. For the weighting, we apply and compare two methods, presented in the next sections.

3.2.1 Kullback-Leibler Weighting

For the final score, we can use the Kullback-Leibler divergence (relative entropy) (Kullback and Leibler, 1951) of P_2 from P_1:

$$D_{KL}(P_1||P_2) = \sum_{x \in X} P_1(x) \log \frac{P_1(x)}{P_2(x)} \quad (8)$$

To measure the information about tonality, we can define our tonality scores based on the divergence between the two category pairs:

$$f_{pol}(\upsilon_i) = D_{KL}(P(pos|\upsilon_i)||P(neg|\upsilon_i)) \quad (9)$$

$$f_{sub}(\upsilon_i) = D_{KL}(P(sub|\upsilon_i)||P(neu|\upsilon_i)) \quad (10)$$

Here, we measure the information lost, if $P(neg|\upsilon_i)$ approximates $P(pos|\upsilon_i)$, for example. The Kullback-Leibler is an asymmetric measure, so a switch of the distributions would give a different result. This is one reason why we prefer our second method, but we evaluate both in order to find out how important the choice of the weighting method is.

3.2.2 Entropy-summand Weighting

Also, the basic idea of the entropy (Shannon, 1948) can be applied to extract the importance of the edges for the tonality.

$$H(X) = -\sum_{i=1}^{n} p(x_i) \log_2(p(x_i)) \quad (11)$$

Here, the $p(x_i)$ refer to the probabilities in the equations 3 to 6. We add or subtract the entropy-summand of the assumed tonality class for one node υ_i to/from a perfect state (normalized to 1 and -1):

$$f_{pol}(\upsilon_i) = \begin{cases} 1 + P(pos|\upsilon_i) * \log_2(P(pos|\upsilon_i)) \\ \quad \text{if } P(neg|\upsilon_i) \leq P(pos|\upsilon_i) \\ -1 - P(neg|\upsilon_i) * \log_2(P(neg|\upsilon_i)) \\ \quad \text{otherwise} \end{cases} \quad (12)$$

$$f_{sub}(\upsilon_i) = \begin{cases} 1 + P(sub|\upsilon_i) * \log_2(P(sub|\upsilon_i)) \\ \quad \text{if } P(neu|\upsilon_i) \leq P(sub|\upsilon_i) \\ -1 - P(neu|\upsilon_i) * \log_2(P(neu|\upsilon_i)) \\ \quad \text{otherwise} \end{cases} \quad (13)$$

In this way, we measure how much disorder one node υ_i provides for a certain tonality class. For a clearly positive node (appears only in positive statements), e.g., the disorder will be 0 and so $f_{pol}(\upsilon_i) = 1$ and also $f_{sub}(\upsilon_i) = 1$.

3.3 Final Scores and Classification

To compute the eight final features values (four for each z-class), we calculate the average scores of all nodes, which share the same category, over all sentences of the statement. If no nodes/edges could be recognized in an unseen statement, all features would be zero. We use a SVM[2] to classify the statements by the extracted features. This works according to the one-versus-all strategy for a non-binary classification, which achieved slightly better results than a one-versus-one strategy or a subjective-objective classification first and then a positive-negative classification. Linear kernels are used and the parameters are the default ones. This

[2] Rapidminer standard implementation (http://rapid-i.com/)

Polarity Features	Subjectivity Features
$T_{v,pol}$: polarity for edges with verbs	$T_{v,sub}$: subjectivity for edges with verbs
$T_{n,pol}$: polarity for edges with nouns	$T_{n,sub}$: subjectivity for edges with nouns
$T_{adv,pol}$: polarity for edges with adverbs	$T_{adv,sub}$: subjectivity for edges with adverbs
$T_{adj,pol}$: polarity for edges w. adjectives	$T_{adj,sub}$: subjectivity for edges w. adjectives

Table 1: Polarity and subjectivity features based on word connections

means that every class has the same priority, for instance.

By using only 8 features, we actually achieve better results if compared with the use of one edge as a feature, because we abstract from individual word combinations in order to prevent overfitting. We will demonstrate that in section 4, where this method of using all edges as features is denoted as the **graph edges** method. Another positive aspect of restricting the number of features to a constant limit is that we save computing time (for the calculation of distances within machine learning, e.g.), because the graphs can be large (cf. section 4).

4 Experiments

4.1 Data and Experimental Setup

We use two different datasets for our evaluation: The pressrelations dataset[3] (called **PDS**) contains 1,521 statements (446 positive, 492 neutral, 583 negative), and a real world dataset contains 8,500 statements (2,125 positive, 2,125 negative, 4,250 neutral) from 5,352 news items about a financial service provider, the so-called **Finance** dataset. Up to ten media analysts (professional experts in the field of MRA) annotate the extracted statements with a tonality. We have investigated their inter-annotator agreement. So, four analysts annotate the same statements from a small part of the statements. They achieve an agreement of 81.8% by using the simple accuracy metric. The PDS has an inter-annotator agreement of 88.06% (Cohen's kappa) (Scholz et al., 2012a). We do not use the viewpoint information contained in the PDS. This is not a problem, because the tonality of statements can be estimated without knowledge of the viewpoint in the most cases.

Nevertheless, a statement can have two different viewpoints in a MRA. This is the case for 116 statements (approx. 7.62%) of the pressrelations dataset and 279 statements of the Finance dataset (approx. 3.28%). Statements can have two different tonalities for different viewpoints, but this is rarely the case (for less than 3.56% of the pressrelations statements and less than 0.17% of the statements of the Finance dataset). One of these examples is the following statement, which is a translated statement of the PDS:

- **Example:** *The logical consequence would be a substantial increase of the subsidies, which the SPD fraction has demanded several times. But the government has limited the funding for 2011 and a too slight rise is planned for 2012.* (Code A: positive, SPD; Code B: negative, CDU)

At the time of the creation of this dataset, the SPD is the biggest opposing party of the CDU in Germany. The CDU is the governing party under its chairwoman Chancellor Merkel. We keep these statements within the dataset, because this case can occur in a MRA. However, we will show that this situation does not irritate our approach too much.

We use approx. 30% of the statements, that is 420 statements (the first 140 positive, neutral, or negative statements) or 2,500 statements (the first 625 positive or negative and the first 1,250 neutral statements) in order to create our graph (the graph has 41,470 or 154,001 edges, resp.). For POS-tagging, identification of negations, and lemmatisation, we apply the TreeTagger (Schmid, 1995). Unless otherwise stated, 20% of the remaining statements (220 and 1,200 statements) are the training set for the SVM and the rest is test set. The size of the test is so large, because we are aiming at a real significance of the solution which can actually be operated in practice.

[3] http://www.pressrelations.de/research/

4.2 Adapting the State-of-the-Art Approaches for a German MRA

For the approaches of Ding et al. (2008), Wilson et al. (2009), and Taboada et al. (2011) we need a sentiment dictionary. Thus, we use the same statements which we use for the creation of our graphs for the creation of a dictionary as one variant.

To create the lexicon of subjectivity clues for the method of **Wilson** et al. (2009), all words which appear more often in neutral statements get the *prior polarity* neutral. For all other words, we calculate the number of appearances in positive statements minus the appearances in negative statements divided by all appearances. A positive word has a value of over 0.2, a negative word has a value of less than -0.2 and the rest has the prior polarity *both*. A positive word with a value above 0.6 belongs to the reliability class *strongsubj*, the other positive words are *weaksubj*. We treat the negative words analogously. We use the Stanford Parser for German (Rafferty and Manning, 2008) to calculate the dependency trees for the sentences (Wilson et al., 2009), in order to extract the General Modification Features, the Polarity Modification Features and the Structure Features. The lists of intensifiers, copular verbs, modals, negations, and polarity shifters are translated by a domain expert, who also added such elements which are not direct translations, but have the same function. The result of this method is a classification of words and phrases. Thus, for a statement classification, we classify the words of the statements and the class of the most frequently used words is the class of the statement (ambiguous statements are classified as the most frequent class). According to the authors, we apply the best machine learning techniques for the word classification (BoosTexter for tonality classification and Ripper for Subjectivity Analysis with parameters as in (Wilson et al., 2009)).

For **Opinion Observer** (Ding et al., 2008), we also identify neutral words if they appear more often in neutral than in subjective statements and subjective words are positive if they appear more often in positive than in negative statements and vice versa for negative words. In contrast to Opinion Mining in customer reviews, we exchange product features through statements and calculate the orientation of opinions for all statements with their opinion orientation algorithm. For this purpose, we adapt the negation rules, the but-clause rule, the inter-sentence conjunction rule, and the "too" rules for German (by translating important words such as "but" or the negations).

SO-CAL (Taboada et al., 2011) needs dictionaries with sentiment values from -5 to +5 with intervals of one. Thus, we use the same scores as the Wilson method and a word with a value above 0.818 to 1 gets a sentiment score of +5 and so on. This means, that neutral words also exist. Our domain expert translated the list of intensifiers (amplifiers and downtoners) and negations, as well as the expert also added missing elements. The authors propose two approaches for the negation search. We use the second, more conservative approach, because this approach works better according to the authors. Also, we use the value 4 for the negation shift. Furthermore, we implement the algorithm of irrealis blocking and translate the list of irrealis markers (modal verbs, conditional markers, negative polarity items, private-state verbs (Taboada et al., 2011)).

For all dictionary-based methods (Wilson, Opinion Observer, SO-CAL), we also evaluate an additional variant which use a sentiment dictionary and not the statements which we use to construct the graphs on each fold. We apply the SentiWS (Remus et al., 2010) for this purpose. As the SentiWS has sentiment values between -1 and 1, we apply similar procedures to construct the method-specific dictionaries as described above: For SO-CAL, it is the same procedure by using the SentiWS values, positive words has a score above 0.33 for Wilson and Opinion Observer, *strongsubj* words have an absolute value above 0.66 and so on. The methods are denoted as *method* (dictionary).

RSUMM (Sarvabhotla et al., 2011) needs less specific adaptation, because only a sentence splitter and a tokenizer are needed. So, RSUMM is very language-independent. We test two versions of this method: one includes the optimization step to estimate the best values for X and Y (notated as RSUMM(X%, Y%)) and the other version (RSUMM(100%)) does without this step, because we believe that every sentence is important in the statements and also because more words mean more information about the tonality in our domain.

We use the sets for the creation of the graphs and lexicons as the validation dataset (VDS) (Sarvabhotla et al., 2011) and the subjectivity dataset (SDS) (Sarvabhotla et al., 2011). As in (Sarvabhotla et al., 2011), we apply the SVMLight package[4] for classification.

Opinion Observer (Ding et al., 2008) and SO-CAL (Taboada et al., 2011) do not use supervised learning. Therefore, we have also added our SVM in order to classify the statements based on the scores of Opinion Observer and SO-CAL (as shown in tables with (+ SVM)).

4.3 Results

Table 2 and 4 (left side) show the results on the pressrelations dataset (PDS) and table 3 and 4 (right side) show the results on Finance. Table 2 and 3 present the tonality classification (positive, neutral, negative) and table 4 displays the Subjectivity Analysis (subjective, neutral).

Word connections (Entropy-summand) achieve the best results with 63.45% accuracy on PDS (more than 15% better than Wilson, which is the best of the 'classical' state-of-the-art methods) and best results on Finance with 65.17% (more than 4% better than RSUMM, which comes in second). The weighting of the edges through the Entropy-summand performs better than the Kullback-Leibler weighting on both datasets, so we use the Entropy-summand weighting for all further experiments.

Also, the improved methods (RSUMM(100%), Opinion Observer (+ SVM), and SO-CAL(+ SVM)) get better results in the majority of cases (the improvement of SO-CAL is more than 13% on PDS and more than 4% on Finance, e.g.). Furthermore, the variants of the methods, which are expanded by a general sentiment dictionary, perform rather worse. The 'classical' Opinion Observer performs better with a general sentiment dictionary, while Wilson tends to achieve worse results in this variant.

Wilson (without an additional dictionary) achieves an accuracy of 42.91% on PDS (Subjectivity Analysis 69.36%) and 48.67% on Finance (Subjectivity Analysis 60.96%) for their word classification. The accuracy of the dictionary variant is 43.44% on PDS and 40.12% on Finance. Therefore, the tonality classification by the most frequent word class seems appropriate for this task and method, because this method achieves better results in the classification of statements than on the word level.

The findings of RSUMM are ambiguous. The 'classical' RSUMM with parameter optimization does not perform very well on PDS, but it performs well on Finance with a high proportion of sentences and words (RSUMM(90%,95%)). Also, if we use all sentences and all features (RSUMM(100%)) we obtain better results on Finance and PDS. This fits in with our assumption that every sentence of a statement is important and that more words lead to more tonality information. The number of word features for RSUMM(100%) is 4,985 features for one statement on PDS and 13,608 features on Finance. After the parameter optimization the size is 974 word features on PDS (RSUMM(80%,20%)) and 12,248 features on Finance (RSUMM(90%,95%)).

The outcomes of this study suggest that methods which include machine learning techniques tend to perform better than unsupervised techniques. The results of the approaches which we expand with a SVM support this conclusion. As mentioned before, only the graph edges obtain a not so high accuracy. This shows the importance of the aggregation of the edges and entropy-based weighting.

We evaluate the influence of the different input sizes and so we performed experiments with 5%, 10%, 40%, and 80% training for machine learning as well as 210 and 840 statements for the creation of dictionaries/graphs on PSD (0.17% training for 210 statements and 0.32% training for 840 statements in order to create the same size of training according to the results of 420 statements). The results are shown in table 5. Opinion Observer and SO-CAL are written in italics, because the results on the left side (size of the training set) belongs to their (+ SVM) variants and the results on the right side are the 'classical' methods with no supervised learning. These experiments show that our word connections remain very stable if the training set is decreased. However, it does not benefit from more training, especially when the training set is very large (80%). Opinion Observer and RSUMM(80%,20%) has the same problem. Nevertheless, it still receives the second-best results, even if another method gets a higher accuracy. However, in our opinion, it is more important

[4] http://svmlight.joachims.org/

Method	Accuracy	Positive		Neutral		Negative	
		prec	rec	prec	rec	prec	rec
Wilson	0.4784	0.358	0.5	0.5423	0.5054	0.5540	0.4444
Wilson (dictionary)	0.4609	0.377	0.3366	0.3664	0.2963	0.5346	0.6223
Opinion Observer	0.3806	0.3732	0.1732	0.3481	0.8267	0.6098	0.1693
Opinion Observer (dictionary)	0.4468	0.5083	0.1993	0.4005	0.8693	0.576	0.2822
RSUMM(80%,20%)	0.403	-	0.0	-	0.0	0.403	1.0
SO-CAL	0.3279	0.3676	0.7353	0.2626	0.3551	0.8461	0.0248
SO-CAL (dictionary)	0.2852	0.2987	0.8464	0.2072	0.1307	0.0075	0.0002
Opinion Observer (+ SVM)	0.3825	-	0.0	0.252	0.1084	0.4037	0.8743
Opinion Observer (dictionary + SVM)	0.3235	0.52	0.2122	0.1322	0.0804	0.346	0.6
RSUMM(100%)	0.4801	0.4586	0.3025	0.8298	0.1354	0.4609	0.8789
SO-CAL (+ SVM)	0.4608	0.463	0.3061	0.3543	0.5699	0.6486	0.48
SO-CAL (dictionary + SVM)	0.3995	0.8235	0.0571	0.3559	0.9371	0.6306	0.2
graph edges	0.5482	0.4313	0.551	0.6578	0.5175	0.5831	0.5714
our approach (Kullback-Leibler)	0.5778	0.5	0.302	0.6642	0.6154	0.5534	0.74
our approach (Entropy-summand)	**0.6345**	0.5346	0.4735	0.6989	0.6818	0.6442	0.7086

Table 2: Results of the experiments on the **PDS**

Method	Accuracy	Positive		Neutral		Negative	
		prec	rec	prec	rec	prec	rec
Wilson	0.5602	0.4206	0.188	0.6358	0.7329	0.4706	0.5872
Wilson (dictionary)	0.4088	0.3678	0.3291	0.5618	0.339	0.3367	0.6132
Opinion Observer	0.4357	0.3641	0.0947	0.5033	0.713	0.2449	0.222
Opinion Observer (dictionary)	0.4583	0.3275	0.186	0.5325	0.664	0.3404	0.3193
RSUMM(90%,95%)	0.6092	0.4433	0.4840	0.731	0.6145	0.5866	0.7233
SO-CAL	0.3478	0.2992	0.5993	0.384	0.373	0.8519	0.046
SO-CAL (dictionary)	0.2905	0.2669	0.9207	0.4429	0.1203	0.001	0.0007
Opinion Observer (+ SVM)	0.4852	0.3384	0.0914	0.496	0.9269	-	0.0
Opinion Observer (dictionary + SVM)	0.4577	0.3299	0.187	0.5118	0.6649	0.3384	0.3177
RSUMM(100%)	0.6088	0.4428	0.4823	0.731	0.6145	0.5854	0.7233
SO-CAL (+ SVM)	0.3921	0.2986	0.7479	0.4573	0.1074	0.599	0.601
SO-CAL (dictionary + SVM)	0.4762	0.3862	0.341	0.544	0.6206	0.3878	0.3244
graph edges	0.5875	0.4437	0.3633	0.6444	0.7096	0.5816	0.5708
our approach (Kullback-Leibler)	0.561	0.3868	0.5445	0.7659	0.5524	0.5201	0.5951
our approach (Entropy-summand)	**0.6517**	0.53	0.5675	0.7714	0.6527	0.5946	0.7351

Table 3: Results of the experiments on **Finance**

Method	Accuracy	Subjective		Objective		Accuracy	Subjective		Objective	
		prec	rec	prec	rec		prec	rec	prec	rec
Wilson	0.6818	0.7251	0.8602	0.4970	0.2975	0.6307	0.6228	0.6649	0.6399	0.5966
Wilson (dictionary)	0.7029	0.7742	0.8636	0.2871	0.179	0.5247	0.5296	0.7944	0.5069	0.2305
Opinion Observer	0.4496	0.7698	0.2724	0.3481	0.8267	0.5047	0.508	0.2963	0.5033	0.713
Opinion Observer (dictionary)	0.5422	0.8635	0.3885	0.4005	0.8693	0.5405	0.5538	0.417	0.5325	0.664
RSUMM(80%,20%)/(90%,95%)	0.3269	-	0.0	0.3269	1.0	0.6919	0.7307	0.6170	0.6630	0.7682
SO-CAL	0.5250	0.7373	0.4686	0.3632	0.6449	0.6127	0.616	0.5983	0.6095	0.627
SO-CAL (dictionary)	0.4378	0.7928	0.235	0.3481	0.8693	0.5155	0.5571	0.1513	0.509	0.8797
Opinion Observer (+ SVM)	0.6061	0.6636	0.8454	0.252	0.1084	0.494	0.4665	0.0636	0.496	0.9269
Opinion Observer (dictionary + SVM)	0.4109	0.88	0.1479	0.3508	0.958	0.5327	0.5732	0.2667	0.5204	0.8003
RSUMM(100%)	0.7083	0.7014	0.9865	0.8298	0.1354	0.6975	0.7424	0.6137	0.6654	0.7829
SO-CAL (+ SVM)	0.5153	0.7485	0.4252	0.3702	0.7028	0.6231	0.7415	0.3814	0.582	0.8663
SO-CAL (dictionary + SVM)	0.3598	0.878	0.0605	0.3345	0.9825	0.511	0.5481	0.1421	0.5055	0.8822
graph edges	0.7037	0.6983	0.9882	0.8205	0.1119	0.6302	0.7821	0.3639	0.5840	0.898
our approach (Kullback-Leibler)	0.7662	0.8215	0.8353	0.6449	0.6224	0.7006	0.6753	0.7761	0.735	0.6247
our approach (Entropy-summand)	**0.7707**	0.8478	0.8050	0.6329	0.6993	**0.739**	0.7179	0.7898	0.7649	0.6878

Table 4: Subjectivity Analysis on **PDS** (left side) and on **Finance** (right side)

Method	0.05	0.1	0.2	0.4	0.8	210	420	840
Wilson	0.4388	0.4743	0.4784	0.5514	0.5795	**0.5275**	0.4784	0.5553
Opinion Observer	0.3403	0.3683	0.3825	0.3979	0.3591	0.3585	0.3806	0.3822
SO-CAL	0.4579	0.439	0.4608	0.4402	0.4818	0.3509	0.3279	0.2702
RSUMM(80%,20%)	0.4063	0.4046	0.403	0.3949	0.3636	0.3226	0.403	0.4557
RSUMM(100%)	0.2964	0.448	0.4801	0.5265	**0.6318**	0.489	0.4801	0.5529
our approach (Entropy-summand)	**0.5717**	**0.5883**	**0.6345**	**0.6278**	0.5818	0.5224	**0.6345**	**0.6452**

Table 5: Different sizes of the training set and the dictionaries/graphs

Features	Level(Wilson)	Level(SO-CAL)	Features	Level(Wilson)	Level(SO-CAL)
$T_{v,pol}$	$-----$	nsc	$T_{v,sub}$	nsc	$+++++$
$T_{n,pol}$	$-----$	$---$	$T_{n,sub}$	$-----$	$++$
$T_{adv,pol}$	$-----$	$-$	$T_{adv,sub}$	$-----$	nsc
$T_{adj,pol}$	$-----$	$-----$	$T_{adj,sub}$	$-----$	nsc
$T_{cat,pol}$	$-----$	nsc	$T_{cat,sub}$	$--$	$+++++$
$T_{cat,z}$(all)	$+++++$	$+++++$			

Table 6: Significance of the tonality features T to the baselines Wilson and SO-CAL

to obtain good results on small training sizes, because over 75% for training would mean that a possible practical implementation would not save much human effort.

4.4 Statistical Significance of the Features

We perform a 10-fold cross validation with our method, Wilson (as the best 'classical' state-of-the-art-method) and SO-CAL (+ SVM) on the pressrelations dataset in order to evaluate the contribution of single tonality features. Our approach (Entropy-summand with all features) achieves an accuracy of 61.94%, while Wilson gets 56.36% and SO-CAL 46.68%. As an analogy to Wilson et al. (2009), we carry out a two-sided t-test with Wilson and SO-CAL (+ SVM) as baselines. The results are shown in table 6. The pluses indicate a significant increase to the baseline, the minuses show a significant decrease. For one sign, changes are significant at the level $p \leq 0.1$, two signs mean $p \leq 0.05$, three signs $p \leq 0.025$, four signs $p \leq 0.01$ and five signs indicate $p \leq 0.005$. "nsc" stands for no significant change.

As shown in table 6, the features with type $z = sub$ are more important than the polarity features. In the categories, the nouns and verbs are more significant than adjectives and adverbs (adverbs are a little stronger in the polarity difference). Combining all features produces a very significant increase against both baselines.

5 Conclusion

We have shown that the word connections outperform state-of-the-art-methods in most cases of tonality classification for a MRA. As a major advantage, our approach does not need much training data. The combination of all tonality features is a significant increase against both baselines, too. The findings show that the word connections in combination with the entropy weighting allow to learn the tonality structure of different word combinations accurately, even though the training size is small. This is a major advantage for a solution, which operates in practice for media analysts, which have to analyse articles for a MRA.

So, this approach in combination with an extraction of statements (Scholz and Conrad, 2013) and the determination of viewpoints (Scholz and Conrad, 2012) represents a fully automated solution in order to perform Opinion Mining for a MRA.

Acknowledgments.

This work is funded by the German Federal Ministry of Economics and Technology under the ZIM-program (Grant No. KF2846501ED1). The authors want to thank the anonymous reviewers for their very helpful comments.

References

Stefano Baccianella, Andrea Esuli, and Fabrizio Sebastiani. 2009. Multi-facet rating of product reviews. In

Proc. of the 31th European Conf. on IR Research on Advances in Information Retrieval, ECIR '09, pages 461–472.

Alexandra Balahur, Ralf Steinberger, Mijail Kabadjov, Vanni Zavarella, Erik van der Goot, Matina Halkia, Bruno Pouliquen, and Jenya Belyaeva. 2010. Sentiment analysis in the news. In *Proc. of the 7th intl. conf. on Language Resources and Evaluation (LREC'10)*, pages 2216–2220.

Danushka Bollegala, David Weir, and John Carroll. 2011. Using multiple sources to construct a sentiment sensitive thesaurus for cross-domain sentiment classification. In *Proc. of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1*, HLT '11, pages 132–141.

William W. Cohen. 1996. Learning trees and rules with set-valued features. In *Proc. of the 13th national conference on Artificial intelligence - Volume 1*, AAAI'96, pages 709–716. AAAI Press.

Xiaowen Ding, Bing Liu, and Philip S. Yu. 2008. A holistic lexicon-based approach to opinion mining. In *Proc. of the Intl. Conf. on Web search and web data mining*, WSDM '08, pages 231–240.

Weifu Du, Songbo Tan, Xueqi Cheng, and Xiaochun Yun. 2010. Adapting information bottleneck method for automatic construction of domain-oriented sentiment lexicon. In *Proc. of the 3rd ACM intl. conf. on Web search and data mining*, WSDM '10, pages 111–120.

Andrew B. Goldberg and Xiaojin Zhu. 2006. Seeing stars when there aren't many stars: graph-based semi-supervised learning for sentiment categorization. In *Proc. of the 1st Workshop on Graph Based Methods for Natural Language Processing*, TextGraphs-1, pages 45–52.

Thorsten Joachims. 1999. Advances in kernel methods. chapter Making large-scale support vector machine learning practical, pages 169–184. MIT Press, Cambridge, MA, USA.

Nobuhiro Kaji and Masaru Kitsuregawa. 2007. Building lexicon for sentiment analysis from massive collection of html documents. In *Proc. of the 2007 Joint Conf. on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 1075–1083.

Solomon Kullback and Richard A. Leibler. 1951. On information and sufficiency. *Annals of Mathematical Statistics*, 22(1):79–86.

Bo Pang and Lillian Lee. 2008. Opinion mining and sentiment analysis. *Foundations and Trends in Information Retrieval*, 2(1-2):1–135.

Natalia Ponomareva and Mike Thelwall. 2012. Do neighbours help?: an exploration of graph-based algorithms for cross-domain sentiment classification. In *Proc. of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, EMNLP-CoNLL '12, pages 655–665.

Rani Qumsiyeh and Yiu-Kai Ng. 2012. Predicting the ratings of multimedia items for making personalized recommendations. In *Proc. of the 35th intl. ACM SIGIR conf. on Research and development in information retrieval*, SIGIR '12, pages 475–484.

Anna N. Rafferty and Christopher D. Manning. 2008. Parsing three german treebanks: lexicalized and unlexicalized baselines. In *Proc. of the Workshop on Parsing German*, PaGe '08, pages 40–46.

Robert Remus, Uwe Quasthoff, and Gerhard Heyer. 2010. Sentiws – a publicly available german-language resource for sentiment analysis. In *Proc. of the 7th Intl. Conf. on Language Resources and Evaluation (LREC'10)*, pages 1168–1171.

Kiran Sarvabhotla, Prasad Pingali, and Vasudeva Varma. 2011. Sentiment classification: a lexical similarity based approach for extracting subjectivity in documents. *Inf. Retr.*, 14(3):337–353.

Robert E. Schapire and Yoram Singer. 2000. Boostexter: A boosting-based systemfor text categorization. *Mach. Learn.*, 39(2-3):135–168.

Helmut Schmid. 1995. Improvements in part-of-speech tagging with an application to german. In *Proc. of the ACL SIGDAT-Workshop*, pages 47–50.

Thomas Scholz and Stefan Conrad. 2012. Integrating viewpoints into newspaper opinion mining for a media response analysis. In *Proc. of the 11th conf. on Natural Language Processing (KONVENS 2012)*, pages 30–38.

Thomas Scholz and Stefan Conrad. 2013. Extraction of statements in news for a media response analysis. In *Proc. of the 18th Intl. conf. on Applications of Natural Language Processing to Information Systems 2013 (NLDB 2013)*, pages 1–12.

Thomas Scholz, Stefan Conrad, and Lutz Hillekamps. 2012a. Opinion mining on a german corpus of a media response analysis. In *Proc. of the 15th International Conference on Text, Speech and Dialogue (TSD 2012)*, pages 39–46.

Thomas Scholz, Stefan Conrad, and Isabel Wolters. 2012b. Comparing different methods for opinion mining in newspaper articles. In *Proc. of the 17th Intl. conf. on Applications of Natural Language Processing to Information Systems 2012 (NLDB 2012)*, pages 259–264.

Claude E. Shannon. 1948. A mathematical theory of communication. *The Bell system technical journal*, 27:379–423.

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

Xiaolong Wang, Furu Wei, Xiaohua Liu, Ming Zhou, and Ming Zhang. 2011. Topic sentiment analysis in twitter: a graph-based hashtag sentiment classification approach. In *Proc. of the 20th ACM intl. conf. on Information and knowledge management*, CIKM '11, pages 1031–1040.

Tom Watson and Paul Noble, 2007. *Evaluating public relations: a best practice guide to public relations planning, research & evaluation*, chapter 6, pages 107–138. PR in practice series. Kogan Page.

Janyce Wiebe, Theresa Wilson, and Claire Cardie. 2005. Annotating expressions of opinions and emotions in language. *Language Resources and Evaluation*, 39(2-3):165–210.

Theresa Wilson, Janyce Wiebe, and Paul Hoffmann. 2009. Recognizing contextual polarity: An exploration of features for phrase-level sentiment analysis. *Computational Linguistics*, 35(3):399–433.

Qiong Wu, Songbo Tan, and Xueqi Cheng. 2009. Graph ranking for sentiment transfer. In *Proc. of the ACL-IJCNLP 2009 Conference Short Papers*, ACLShort '09, pages 317–320.