Fine-Grained Sentiment Analysis with Structural Features

Cäcilia Zirn† Mathias Niepert† Heiner Stuckenschmidt† Michael Strube‡

†KR & KM Research Group
University of Mannheim
Mannheim, Germany
caecilia@informatik.uni-mannheim.de

‡Heidelberg Institute for
Theoretical Studies
Heidelberg, Germany
michael.strube@h-its.org

Abstract

Sentiment analysis is the problem of determining the polarity of a text with respect to a particular topic. For most applications, however, it is not only necessary to derive the polarity of a text as a whole but also to extract negative and positive utterances on a more fine-grained level. Sentiment analysis systems working on the (sub-)sentence level, however, are difficult to develop since shorter textual segments rarely carry enough information to determine their polarity out of context. In this paper, therefore, we present a fully automatic framework for fine-grained sentiment analysis on the subsentence level combining multiple sentiment lexicons and neighborhood as well as discourse relations to overcome this problem. We use Markov logic to integrate polarity scores from different sentiment lexicons with information about relations between neighboring segments, and evaluate the approach on product reviews. The experiments show that the use of structural features improves the accuracy of polarity predictions achieving accuracy scores of up to 69%.

1 Introduction

Sentiment analysis systems have continuously improved the quality of polarity classifications of entire product reviews. For numerous real-world applications, however, classification on such a coarse level is not suitable. Even in their most enthusiastic reviews, users still tend to mention negative aspects of a particular product. Conversely, in very negative reviews there might still be mentions of several positive aspects of the product. Moreover, different opinions can even be uttered in the same sentence. Consider, for instance, the sentence "*Despite the pretty design I would never recommend it, because the sound quality is unacceptable*" which expresses both positive and negative opinions about a product. Thus, to determine both negative and positive utterances in product reviews, classification on the subsentence level is needed.

Sentiment Analysis on Subsentence Level. As basic classification unit for our fine-grained sentiment analysis system we choose discourse segments. There are various theories describing discourse, discourse segmentation and discourse relations. The most well-known theory aiming to describe some aspects of text coherence is the Rhetorical Structure Theory (RST) introduced by Mann and Thompson (1988). According to this theory, every text consists of elementary segments that are connected by relations. Segments joined by a relation form a unit, which is itself connected to other segments. This leads to a hierarchical tree structure that spans over the whole text. The example sentence given above could be divided into the three segments s_1 = *Despite the pretty design*, s_2 = *I would never recommend it* and s_3 = *because the sound quality is unacceptable*, with a CONCESSION relation[1] holding between s_1 and s_2 and a CAUSE-EXPLANATION-EVIDENCE relation holding between s_2 and s_3. As the segments form logical units, and parts of sentences bearing different polarities are contrastive and thus do not constitute a logical unit, we claim that the discourse segment level is appropriate for fine-grained sentiment analysis.

Integrating Neighborhood Relations. As discourse segments consist of only a few tokens, they

[1]Please note that in this work we do not distinguish between CONCESSION and CONTRAST relations and consider both as CONTRAST relations. In the following, we will refer to all other kind of relations as NO_CONTRAST relations.

Proceedings of the 5th International Joint Conference on Natural Language Processing, pages 336–344,
Chiang Mai, Thailand, November 8 – 13, 2011.

rarely carry enough information to determine their polarity out of context. While it occurs that neighboring segments bear opposite polarities, like in the example given above, two segments following each other are mostly of the same polarity. Therefore, when determining the polarity of a discourse segment, we consider the polarity of the neighboring segments for the classification.

Leveraging Contrast Relations. Although mentioning positive and negative opinions next to each other constitutes a contrast, we cannot conclude that every contrast indicates a polarity change. We conducted a simple corpus study, focusing on the cue word *but* which is a strong indicator for contrast relations. Of all consecutive discourse segments connected by the word *but*, 40% express opposite and 60% express the same opinion. Of all the other discourse segment pairs, only 10% express differing opinions. From this experimental observation, we conclude that two neighboring segments not related by a contrast relation have a much higher probability of bearing opinions of the same polarity than segments connected by a contrast relation. In our experiments, we will investigate whether the distinction between CONTRAST and NO_CONTRAST relations will improve fine-grained sentiment analysis.

Collective classification. The challenge of fine-grained sentiment analysis is that shorter text segments pose a more difficult classification problem. There are various approaches to determining the polarity of text. One common approach is the look-up of terms in a sentiment lexicon with polarity scores. As discussed in the previous paragraphs, we claim that incorporating information about a segment's neighbors, the classification of small text segments can be improved on. Therefore, we simultaneously determine the most probable classification of all segments in a review. We use Markov logic to combine polarity scores from different sentiment lexicons with information about discourse relations between neighboring segments, and evaluate the method on product reviews.

2 Related Work

Methods for fine-grained sentiment analysis are developed by Hu and Liu (2004), Ding et al. (2008) and Popescu and Etzioni (2005). While the approaches of the former two operate on the sentence level, the system of the latter - Popescu and Etzioni (2005) - extracts opinion phrases on the subsentence level for product features. Their approaches have in common that they first extract features of a product, like the *size* of a camera or its *weight*. Then, they look for opinion words describing these features. Finally, the polarity of these terms and, thus, of the feature is determined. An even finer-grained system is presented in Kessler and Nicolov (2009). The approach aims at classifying both sentiment expressions as well as their targets using a rich set of linguistic features. However, they have not implemented the component that detects and analyses sentiment expressions, but focus on target detection.

Täckström and McDonald (2011) combine fully and partially supervised structured conditional models for a joint classification of the polarity of whole reviews and the review's sentences.

An approach based on assumptions similar to our intuition to integrate discourse relations is described in Kim and Hovy (2006) where the authors label sentences as reasons for or against purchasing a product. The system makes use of conjunctions like "and" to infer polarities and applies a specific rule to sentences including the word "but": if no polarity can be identified for the clause containing "but", the polarity of the previous phrase is taken and negated. In our system, we incorporate this information using discourse relations.

The impact of discourse relations for sentiment analysis is investigated in Asher et al. (2009). The authors conduct a manual study in which they represent opinions in text as shallow semantic feature structures. These are combined to an overall opinion using hand-written rules based on manually annotated discourse relations. An interdependent classification scenario to determine polarity as well as discourse relations is presented in Somasundaran and Wiebe (2009). In their approach, text is modeled as opinion graphs including discourse information. In Somasundaran et al. (2009) the authors try alternative machine learning scenarios with combinations of supervised and unsupervised methods for the same task. However, they do not determine discourse relations automatically but use manual annotations.

3 Statistical-Relational Representation

The basic idea of our approach is the integration of heterogeneous features such as polarity scores from sentiment lexicons and neighborhood relations between segments. We use concepts and algorithms from statistical relational learning and, in particular, Markov logic networks (Richardson and Domingos, 2006).

We briefly introduce Markov logic as a framework for combining numerical and structural features and describe how the problem of fine-grained sentiment analysis based on multiple lexicons and discourse relations can be represented in the language. Most probable polarity classifications are then derived by computing maximum a-posteriori (MAP) states in the ground Markov logic network.

3.1 Markov Logic Networks

Markov logic (Richardson and Domingos, 2006) can be as a first-order template language for log-linear models with binary variables. Log-linear models are parameterizations of undirected graphical models (Markov networks) which play an important role in the areas of reasoning under uncertainty (Koller and Friedman, 2009) and statistical relational learning (Getoor and Taskar, 2007). Please note that log-linear models are also known as maximum-entropy models in the NLP community (Manning and Schütze, 1999). The features of a log-linear model can be complex and allow the user to incorporate prior knowledge about what types of data are expected to be important for classification.

A Markov network $\mathcal{M}$ is an undirected graph whose nodes represent a set of random variables $\mathbf{X} = \{X_1, ..., X_n\}$ and whose edges model direct probabilistic interactions between adjacent nodes. More formally, a distribution P is a log-linear model over a Markov network $\mathcal{M}$ if it is associated with:

- a set of features $\{f_1(D_1), ..., f_k(D_k)\}$, where each D_i is a clique in $\mathcal{M}$ and each f_i is a function from D_i to $\mathbb{R}$,
- a set of real-valued weights $w_1, ..., w_k$, such that

$$P(\mathbf{X} = \mathbf{x}) = \frac{1}{Z} \exp\left(\sum_{i=1}^{k} w_i f_i(D_i)\right),$$

where Z is a normalization constant.

A Markov logic network is a set of pairs (F_i, w_i) where each F_i is a first-order formula and each w_i a real-valued weight associated with F_i. With a finite set of constants C it defines a log-linear model over possible worlds $\{\mathbf{x}\}$ where each variable X_j corresponds to a ground atom and feature f_i is the number of true groundings (instantiations) of F_i with respect to C in possible world $\mathbf{x}$. Possible worlds are truth assignments to all ground atoms with respect to the set of constants C. We explicitly distinguish between weighted formulas and *deterministic* formulas, that is, formulas that always have to hold.

Inference

There are two common types of inference tasks for a Markov logic network: Maximum a-posteriori inference and (conditional) probability inference. The latter computes the posterior probability distribution over a subset of the variables given an instantiation of a set of evidence variables. MAP inference, however, is concerned with finding a joint assignment to a subset of variables with maximal probability. Assume we are given a set $\mathbf{X}' \subseteq \mathbf{X}$ of instantiated variables and let $\mathbf{Y} = \mathbf{X} \setminus \mathbf{X}'$. Then, a most probable state of the ground Markov logic network is given by

$$\underset{\mathbf{y}}{\operatorname{argmax}} \sum_{i=1}^{k} w_i f_i(D_i).$$

Parameter Learning

Given a set of first-order formulas and a set of ground atoms, we wish to find the formulas' maximum a posteriori (MAP) weights, that is, the weights that maximize the log-likelihood of the hidden variables given the evidence. There exist several learning algorithms for Markov logic such as voted perceptron, contrastive divergence, and scaled conjugate gradient (Lowd and Domingos, 2007).

We employed the voted perceptron learner for the experiments (Richardson and Domingos, 2006; Lowd and Domingos, 2007; Riedel, 2008) which performs gradient descent steps to approximately optimize the conditional log-likelihood. In a MLN, the derivative of the conditional log-likelihood with respect to a weight w_i is the difference between the number of true groundings f_i of the formula F_i in the training data and the expected number of groundings according to the

model with weights $\mathbf{w}$

$$\mathrm{g}_i = \frac{\partial}{\partial w_i} \log P(\mathbf{Y} = \mathbf{y} | \mathbf{X}' = \mathbf{x}') = f_i - E_{\mathbf{w}}[f_i].$$

The expected number of true groundings $E_{\mathbf{w}}[f_i]$ is determined by (approximately) computing a MAP state with the current weights $\mathbf{w}$. The perceptron update rule for the set of weights $\mathbf{w}$ for epoch $t+1$ is then

$$\mathbf{w}_{t+1} = \mathbf{w}_t + \eta \mathbf{g},$$

where η is the learning rate. Online learners repeat these steps updating the weight vector for a predetermined number of n epochs.

3.2 Markov Logic Formulation

Each discourse segment s is modeled with a constant symbol $c \in C$. The set C, therefore, models the discourse segments in the text under consideration and comprises the set of constants of the Markov logic network. The segments s_1, s_2, and s_3 depicted in Figure 1, for instance, would be modeled using the constant symbols c_1, c_2 and c_3. We represent the polarity of a segment using two non-observable predicates $positive$ and $negative$. Note that the state of variables modeling non-observable ground predicates is only known during weight learning. We first formulate the fact that a segment is positive or negative but cannot be $positive$ and $negative$ at the same time using the following deterministic formulas:

$$\begin{aligned} \forall x : \neg positive(x) &\Rightarrow negative(x) \\ \forall x : negative(x) &\Rightarrow \neg positive(x) \end{aligned}$$

Furthermore, the model incorporates several numerical *a-priori* features such as the polarity scores of individual segments provided by external lexical resources. We introduce these features in the experimental section in more detail. For each of these features ℓ we wish to include in the model, we first add the following deterministic equivalence formulas

$$\begin{aligned} \forall x : positive\_source_\ell(x) &\Leftrightarrow positive(x) \\ \forall x : negative\_source_\ell(x) &\Leftrightarrow negative(x) \end{aligned}$$

Now, in order to include a-priori polarity scores, we add the weighted formula $positive\_source_\ell(x)$ and *scale* the contribution of a *true* ground atom $positive\_source_\ell(s)$ with the a-priori polarity score of the particular segment s. This way, the parameter learning algorithm balances the contributions of the different sources according to their accuracy on the training data. The framework "Markov theBeast"[2] which we used for our experiments allows to add such real-valued features (Riedel, 2008).

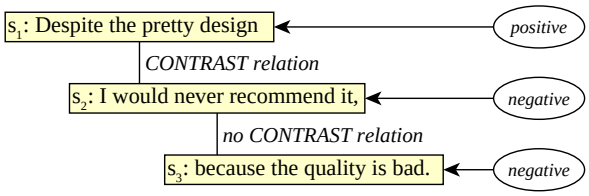

Figure 1: Sentiment polarities of and discourse relations between the segments of a sentence.

The novel contribution of the present paper, however, is the addition of *structural features*, that is, features that model specific dependencies holding between the segments of a review. We distinguish two different types of such features, namely, *neighborhood relations* and *discourse relations*.

3.2.1 Neighborhood Relations

The intuition behind *neighborhood relations* is that neighboring segments are more likely to have the same polarity. We model the fact that a segment *precedes* another segment with the observable predicate pre. Each sentence is represented as a set of ground predicates instantiated by constants modeling consecutive sentence segments. The sentence depicted in Figure 1, for instance, would be represented with the two ground atoms $pre(c_1, c_2)$ and $pre(c_2, c_3)$. The following formulas are included in the Markov logic formulation to model the dependency of preceding segments.

$$\begin{aligned} \forall x, y : pre(x, y) \wedge positive(x) \Rightarrow positive(y) \\ \forall x, y : pre(x, y) \wedge negative(x) \Rightarrow negative(y) \end{aligned}$$

The weights of the above formulas (a subset of the parameters of the model) are learned during training.

3.2.2 Discourse Relations

While there are numerous types of possible discourse relations we decided to only distinguish between contrast relations ($contrast$) and all other types of relations ($nconstrast$) due to their potential impact on polarity changes between discourse segments. In principle, however, it is possible to

[2]The code can be downloaded at `http://code.google.com/p/thebeast/`

Topic	p	n	total
Cell Phones & Service	1392	1785	3177
Gourmet Food	990	616	1606
Kitchen & Housewares	1188	1405	2593
Sum	3570	3806	7376

Table 1: Amount of positive (p) and negative (n) segments.

extend the model to also incorporate additional relations. The sentence shown in Figure 1, for instance, would be represented with the two ground atoms $contrast(c_1, c_2)$ and $ncontrast(c_2, c_3)$.

In order to leverage contrast relations, we included the following formulas in the Markov logic formulation, modeling how the absence of contrast relations between segments influences their potential polarity changes.

$$\begin{aligned}
\forall x, y : contrast(x, y) \ \wedge \ positive(x) \quad &\Rightarrow negative(y) \\
\forall x, y : contrast(x, y) \ \wedge \ negative(x) \quad &\Rightarrow \ positive(y) \\
\forall x, y : ncontrast(x, y) \wedge positive(x) \quad &\Rightarrow \ positive(y) \\
\forall x, y : ncontrast(x, y) \wedge negative(x) \quad &\Rightarrow negative(y)
\end{aligned}$$

Again, the weights of the above formulas are learned in the training phase.

The classification of a given set of segments is now equivalent to computing a maximum a-posteriori (MAP) state of the respective ground Markov logic network.

4 Experiments

In what follows, we describe the individual components of our sentiment analysis system and the data we used to experimentally evaluate it. For the evaluation, we first combine real-valued polarity scores derived from sentiment lexicons using Markov logic networks and classify all segments of a product review. We then investigate whether the addition of certain structural features improves the performance of the system.

4.1 Data

We chose a subset of the the Multi-Domain Sentiment Dataset arranged by Blitzer et al. (2007) and annotated it for our purpose. The Multi-Domain Sentiment Dataset consists of user-written product reviews downloaded from the web page `http://amazon.com`. The reviews are subdivided according to their topics. We included the three categories "Cell Phones & Service", "Gourmet Food" and "Kitchen & Housewares".

Each category consists of up to 100 reviews. A review is already classified as positive or negative according to the amount of stars the user has chosen for the product along with their review. To achieve a balanced corpus, we picked the 20 longest positive and the 20 longest negative reviews for each of the two topics, resulting in a complete amount of 120 reviews. Table 1 lists the three categories and their respective numbers of segments.

4.1.1 Gold Standard

Three independent annotators were instructed to label all passages of a review as `positive`, `negative` or `neutral`. Here, a passage is defined as a sequence of words sharing the same opinion. Each word of a review belongs to exactly one passage. The annotators were instructed to choose arbitrary passage boundaries independent of sentence or clause limits. The inter-annotator agreement among the three annotators varies from κ = 0.40 to κ = 0.45 for negative reviews, which is considered only fair agreement, and from κ = 0.60 to κ = 0.84 for positive reviews which is considered strong agreement according to Fleiss kappa (Fleiss, 1971). In our experiments, we only use the two classes `positive` and `negative`. Because of the individual segmentations, we processed the corpus word by word to determine the final polarity labels. For each word, we considered the three polarity labels the annotators had chosen for the respective passages containing the word. If one of the labels `positive` or `negative` was used in the majority, we chose this as the final label. Whenever the majority of the annotators picked `neutral` or each of the annotators chose a different label the general polarity of the entire review as given by the data set was taken as final label. This is because we estimate the user chose the star-rating according to his overall opinion on the product he is reviewing. This general opinion is expressed by the review text and, therefore, the "standard" label for the review represents the overall opinion. The numbers of positive and negative segments according to the gold standard are shown in Table 1.

The final output of our fine-grained sentiment analysis system are discourse segments labeled as positive or negative. To compare them to the gold standard, we determine the polarity labels of all tokens belonging to the segment in the gold standard and take the most-chosen label. Again, if there is

the same amount of positive and negative labels, we take the overall polarity of the whole review as label.

4.2 Polarity Features

For each segment, we estimate prior positivity and negativity scores using state-of-the-art sentiment classification methods. There are two basic ways to classify polarity. One of the most common approaches is to train a classifier on labeled data that works with a bag-of-words model or uses similar features. However, named approach will have difficulties with the short text segments our system is focused on.

Another method for polarity classification is to look up terms in a pre-compiled sentiment lexicon that lists terms and their polarities. We chose the latter method for several reasons. First, lexicon-based methods do not rely on large amounts of training data. Second, lexicons can easily be exchanged or added which makes the approach more flexible. Third, the use of Markov logic allows us to combine several lexicons without additional effort. To compute the positivity and negativity score for a segment according to a lexicon, we first look up the positivity as well as the negativity of each term of the segment in this lexicon. Then, we average the positivity as well as the negativity scores. This leads to one positivity score and one negativity score per lexicon for each segment. We use a simple heuristic to consider negated polarity terms such as in *not good*. To this end, we manually compiled a list of negation terms[3]. Every time we detect such a negation indicator within a segment, we switch the positivity and the negativity scores of all terms occurring after said negation. We employ the following lexicons:

- **SentiWordNet** (SWN)
  SWN (Esuli and Sebastiani, 2006) is a lexical resource that contains positivity-scores, negativity-scores and objectivity-scores for WordNet (Fellbaum, 1998) synsets. The scores are between 0.0 and 1.0 and all three scores for a synset sum up to 1.0. For our system, we only regard positivity scores and negativity scores. We use a part-of-speech tagger and take the first word sense.

- **Taboada and Grieve's Turney Adjective List** (TGL)
  Taboada and Grieve (2004) created a list containing adjectives and their polarities based on a method described by Turney (2002). They first query a search engine for the adjective together with some manually chosen clearly positive adjectives, using the `near`-operator, then they do the same with a list of negative adjectives. Finally, they calculate the point-wise mutual information (Church and Hanks, 1990) between the queries.

- **Unigram Lexicon** (UL)
  There are terms whose polarity depends on the context they are used in. Consider for instance the word *large*: a *large screen* is good while a *large cell phone* is likely bad. To take domain-dependence into account, we compile a list of common positive and negative unigrams as well as punctuation marks for each of the three topics separately. Since we need 40 reviews per topic for the evaluation only the remaining reviews are used to compile the unigram lexicon. From this data, we calculate the ratio of all occurrences of a unigram in positive reviews to its occurrences in negative reviews and use this ratio as the positivity and negativity scores, respectively.

4.2.1 Discourse Parsing

We employ the discourse parser HILDA developed by duVerle and Prendinger (2009). It performs two tasks: First, it splits the review text into discourse segments which constitute the basic entities our system classifies. Second, it determines the discourse relations between segments. The actual output of HILDA is the discourse tree of a text. We convert the tree structure to a linear sequence of relations between neighboring segments. HILDA uses the set of relation labels described by Soricut and Marcu (2003) which is coarser-grained than RST and consists of 18 labels. For the experiments, we distinguished two types of relations: relations labeled as $contrast$ and all other relations. We refer to this class as $ncontrast$. We model these two relations in the Markov logic framework as described in Section 3.2.

We want to investigate whether the use of a discourse parser is improving fine-grained sentiment analysis. The discourse segments determined by

[3] We used the negation indicators *no, cannot, not, none, nothing, nowhere, neither, nor, nobody, hardly, scarcely, barely* and all negations of auxiliaries modals ending on *n't*, like *don't* or *won't*.

	positive			negative			
	P	R	F	P	R	F	A
majority baseline	0.00	0.00	0.00	51.60	100.00	68.07	51.60
SVM	57.05	43.06	49.08	56.44	69.47	62.28	56.66
MLN_polarity	53.21	69.58	60.31	59.90	42.62	49.80	55.67
MLN_neighborhood	66.38	72.94	**69.50**	72.02	65.34	**68.52**	**69.02**
MLN_contrast	61.39	73.47	66.89	69.48	56.65	62.41	64.79

Table 2: Results (%) for the different systems. P = precision, R = recall, F = F-measure, A = accuracy

the discourse parser constitute the basic units for our sentiment classification system. Evaluating the correctness of discourse parsing is a hard task. However, it is not of prime importance for our task that the segments are correct according to any discourse theory but that they do not include passages containing differing labels according to the gold standard. An analysis of the data shows that only 3.2% of the segments contain contradictory labels. We therefore concluded that it is appropriate to use the discourse segments as basic units for the evaluation of our system.

4.3 Experimental Setting

The goal of our system is to label discourse segments of a review as `positive` or `negative`. We employed the Markov logic network implementation "Markov theBeast"(Riedel, 2008).

We compare three different Markov logic networks. First, we only take into account the real-valued polarity features. We consider this ML formulation ("MLN_polarity") a baseline to evaluate the quality of the evidence collected from the sentiment lexicons. To compare the performance of this system to the state of the art in classification algorithms, we train a Support Vector Machine (SVM) (Platt, 1998; Keerthi et al., 2001; Hastie and Tibshirani, 1998) on the polarity features. In a second Markov logic formulation (MLN_neighborhood), we incorporate structural information about neighboring segments using the formulas described in section 3.2.1. In order to assess the impact of explicitly distinguishing between `contrast` and `ncontrast` relations, we use the Markov logic formulation described in Section 3.2.2 (MLN_contrast).

We learn the weight parameters of the Markov logic networks by running the voted perceptron online learner for 20 epochs (Riedel, 2008). We then evaluate each of the classification algorithms with 10-fold cross validation.

4.4 Results

Table 2 lists the evaluation results for the different classifiers. To determine statistical significance of the relative effectiveness of two classifiers we applied a paired t-test at a significance level of $p < 0.01$. The classifiers exclusively using polarity features have comparable accuracy values. While the SVM is showing a bias towards classifying segments as negative ML_polarity shows the opposite trend. Although the accuracy of SVM is slightly higher the relative difference of the accuracy values is not statistically significant. Including neighborhood relations increases the effectiveness relative to both non-structure based classifiers significantly. MLN_neigborhood achieves an F-measure of 69.50% for positive segments and 68.52% for negative segments with an overall accuracy of 69.02%. It also significantly outperforms the majority baseline which achieves an accuracy of 51.60%. Contrary to our hypothesis, distinguishing between `contrast` and `ncontrast` relations did *not* improve the effectiveness relative to MLN_neigborhood. MLN_contrast achieves a slightly lower accuracy than MLN_neighborhood although the difference is not statistically significant. These results suggest that the correlation of `contrast` relations and polarity changes is not significant. Furthermore, the number of contrast relations in product reviews is too small to have a significant impact. Finally, employing a discourse parser as a component of a sentiment analysis poses the problem that misclassifications might as well be caused by erroneous decisions of the component. Figure 2 depicts the accuracy values for the different classifiers on each of the ten cross-validation folds.

To the best of our knowledge, there is no sen-

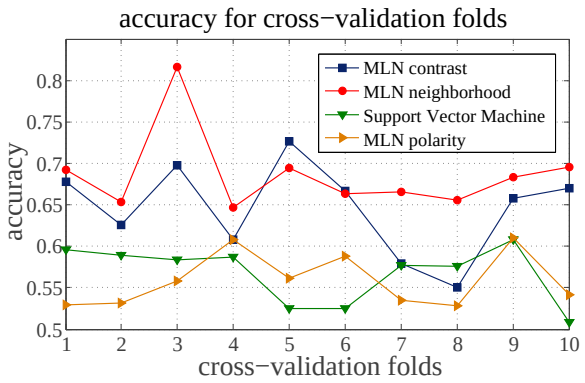

Figure 2: Accuracy values of the various algorithms for the 10 different cross-validation folds.

timent analysis system operating on the discourse segment level to which we could compare our results. However, the task is similar to that approached by Kim and Hovy (2006) whose system achieves an accuracy of 57% classifying whole sentences of reviews as positive or negative. In Täckström and McDonald (2011), the authors present a semi-supervised approach classifying sentences as positive, negative or neutral. Their approach achieves an accuracy of up to 59.1%. Considering the fact that our system is working on subsentence level we find our results promising.

5 Conclusion and Future Work

In this work, we addressed the problem of fine-grained sentiment analysis on subsentence level, achieving an accuracy of 69%. We proposed a sentiment classification method that uses Markov logic as a means to integrate polarity information from different sources and to explicitly use information about the structure of text to determine the polarity of text segments. The approach has a number of advantages. It is flexible enough to incorporate polarity scores from various sources. We used two pre-existing sentiment lexicons. To capture domain-dependent knowledge, we compiled an individual lexicon for each domain from training data. The presented approach, however, is not restricted to these sources and can include any source of polarity features. It allows for an easy combination of various existing methods into a single polarity judgement. Moreover, its major advantage is the inclusion of structural information. Again, this ability is more or less independent from a concrete method. In our work we used an existing discourse parser, however, other methods for determining the discourse structure could be used as well. Finally, the Markov logic representation can be used in a supervised and in an unsupervised setting. The experiments described in the paper are based on the supervised setting: we used a manually annotated corpus to learn weights for the formulas in the Markov logic model. In cases, where no annotated corpus is available, we could still set the weights by hand and experiment with different settings until a good setting is found.

Concerning fine-grained sentiment analysis the main result of our work is that the use of general structures found in the text systematically improves the results. As described in the paper, it turned out, however that the relation between the contrast relation and the change of polarity is not as close as we had expected. This means that the classical discourse relations are not necessarily the best choice concerning text structures to be taken into account. However, we think that focusing on cue words for discourse connectives is worth being investigated to determine features that allow us to more accurately predict such polarity changes. Further, in the work reported here, we only considered positive and negative polarity. This raises some questions concerning the treatment of segments that do not have a clear polarity. In future work, we will therefore extend our experiments to the case where segments can be classified as `positive`, `negative` or `neutral`.

Acknowledgments

We would like to thank Anette Frank for useful comments in the early phase of this work, and the annotators for annotating the product reviews.

References

Nicholas Asher, Farah Benamara, and Yannick Mathieu. 2009. Appraisal of opinion expressions in discourse. *Lingvisticae Investigations*, 31(2):279–292.

John Blitzer, Mark Dredze, and Fernando Pereira. 2007. Biographies, Bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification. In *Proc. of ACL-07*, pages 440–447.

Kenneth Ward Church and Patrick Hanks. 1990. Word association norms, mutual information, and lexicography. *Computational Linguistics*, 16(1):22–29.

Xiaowen Ding, Bing Liu, and Philip S. Yu. 2008. A holistic lexicon-based approach to opinion mining. In *Proc. of WSDM-08*, pages 231–240.

David duVerle and Helmut Prendinger. 2009. A novel discourse parser based on support vector classification. In *Proc. of ACL-IJCNLP-09*, pages 665–673.

Andrea Esuli and Fabrizio Sebastiani. 2006. SentiWordNet: A publicly available lexical resource for opinion mining. In *Proc. of LREC '06*, pages 417–422.

Christiane Fellbaum, editor. 1998. *WordNet: An Electronic Lexical Database*. MIT Press, Cambridge, Mass.

Joseph L. Fleiss. 1971. Measuring nominal scale agreement among many raters. *Psychological Bulletin*, 76(5):378–382.

Lise Getoor and Ben Taskar. 2007. *Introduction to Statistical Relational Learning*. MIT Press.

Trevor Hastie and Robert Tibshirani. 1998. Classification by pairwise coupling. In Michael I. Jordan, Michael J. Kearns, and Sara A. Solla, editors, *Advances in Neural Information Processing Systems*, volume 10, pages 451–471. MIT Press.

Minqing Hu and Bing Liu. 2004. Mining and summarizing customer reviews. In *Proc. of ACM SIGKDD '04*, pages 168–177.

S.S. Keerthi, S.K. Shevade, C. Bhattacharyya, and K.R.K. Murthy. 2001. Improvements to Platt's SMO algorithm for SVM classifier design. *Neural Computation*, 13(3):637–649.

Jason S. Kessler and Nicolas Nicolov. 2009. Targeting sentiment expressions through supervised ranking of linguistic configurations. In *Proc. of ICWSM-09*, pages 90–97.

Soo-Min Kim and Eduard Hovy. 2006. Automatic identification of pro and con reasons in online reviews. In *Proc. of COLING-ACL-06 Poster Session*, pages 483–490.

Daphne Koller and Nir Friedman. 2009. *Probabilistic Graphical Models: Principles and Techniques*. MIT Press.

Daniel Lowd and Pedro Domingos. 2007. Efficient weight learning for Markov logic networks. In *Proc. of ECML/PKDD-07*, pages 200–211.

William C. Mann and Sandra A. Thompson. 1988. Rhetorical structure theory. Toward a functional theory of text organization. *Text*, 8(3):243–281.

Christopher D. Manning and Hinrich Schütze. 1999. *Foundations of statistical natural language processing*. MIT Press.

J. Platt. 1998. Fast training of support vector machines using sequential minimal optimization. In B. Schoelkopf, C. Burges, and A. Smola, editors, *Advances in Kernel Methods - Support Vector Learning*, pages 185–208. MIT Press.

Ana-Maria Popescu and Oren Etzioni. 2005. Extracting product features and opinions from reviews. In *Proc. HLT-EMNLP '05*, pages 339–346.

Matthew Richardson and Pedro Domingos. 2006. Markov logic networks. *Machine Learning*, 62(1-2):107–136.

Sebastian Riedel. 2008. Improving the accuracy and efficiency of MAP inference for Markov logic. In *Proc. of UAI-08*, pages 468–475.

Swapna Somasundaran and Janyce Wiebe. 2009. Recognizing stances in online debates. In *Proc. of ACL-IJCNLP-09*, pages 226–234.

Swapna Somasundaran, Galileo Namata, Janyce Wiebe, and Lise Getoor. 2009. Supervised and unsupervised methods in employing discourse relations for improving opinion polarity classification. In *Proc. EMNLP-09*, pages 170–179.

Radu Soricut and Daniel Marcu. 2003. Sentence level discourse parsing using syntactic and lexical information. In *Proc. of HLT-NAACL-03*, pages 149–156.

Maite Taboada and Jack Grieve. 2004. Analyzing appraisal automatically. In *Proceedings of the AAAI Spring Symposium on Exploring Attitude and Affect in Text: Theories and Applications,* Palo Alto, Cal., 22–24 March 2004, pages 158–161.

Oscar Täckström and Ryan McDonald. 2011. Semi-supervised latent variable models for sentence-level sentiment analysis. In *Proc. of ACL-11*, pages 569–574.

Peter Turney. 2002. Thumbs up or thumbs down? Semantic orientation applied to unsupervised classification of reviews. In *Proc. of ACL-02*, pages 417–424.