Improvement of a Naive Bayes Sentiment Classifier Using MRS-Based Features

Jared Kramer
University of Washington
Seattle, WA
jaredkk@uw.edu

Clara Gordon
University of Washington
Seattle, WA
cgordon1@uw.edu

Abstract

This study explores the potential of using deep semantic features to improve binary sentiment classification of paragraph-length movie reviews from the IMBD website. Using a Naive Bayes classifier as a baseline, we show that features extracted from Minimal Recursion Semantics representations in conjunction with back-off replacement of sentiment terms is effective in obtaining moderate increases in accuracy over the baseline's n-gram features. Although our results are mixed, our most successful feature combination achieves an accuracy of 89.09%, which represents an increase of 0.76% over the baseline performance and a 6.48% reduction in error.

1 Introduction

Text-based sentiment analysis offers valuable insight into the opinions of large communities of reviewers, commenters and customers. In their survey of the field, Pang and Lee (2008) highlight the importance of sentiment analysis across a range of industries, including review aggregation websites, business intelligence, and reputation management. Detection and classification of sentiment can improve downstream performance in applications sensitive to user opinions, such as question-answering, automatic product recommendations, and social network analysis (ibid., p. 12).

While previous research in sentiment analysis has investigated the extraction of features from syntactic dependency trees, semantic representations appear to be underused as a resource for modeling opinion in text. Indeed, to our knowledge, there has been no research using semantic dependencies created by a precision grammar for sentiment analysis. The goal of the present research is to address this gap by augmenting a baseline classifier with features based on Minimal Resursion Semantics (MRS; Copestake et al., 2005), a formal semantic representation provided by the English Resource Grammar (ERG; Flickinger, 2000). An MRS is a connected graph in which semantic entities may be linked directly through shared arguments or indirectly through handle or qeq constraints, which denote equality modulo quantifier insertion (Copestake et al., 2005). This schema allows for underspecification of quantifier scope.

Using Narayanan et al.'s (2013) Naive Bayes sentiment classifier as a baseline, we test the effectiveness of eight feature types derived from MRS. Our feature pipeline crawls various links in the MRS representations of sentences in our corpus of paragraph-length movie reviews and outputs simple, human-readable features based on various types of semantic relationships. This improved system achieves modest increases in binary sentiment classification accuracy for several of the feature combinations tested.[1]

In the following sections, we summarize previous research in MRS feature extraction and sentiment classification, describe the baseline system and our modifications to it, and outline our approach to parsing our data, constructing features, and integrating them into the existing system. Finally, we report our findings, examine in more detail where our improved system succeeded and failed in relation to the baseline, and suggest avenues for further research in sentiment analysis with MRS-based features.

2 Context and Related Work

Current approaches to sentiment analysis tasks typically use supervised machine learning meth-

[1] Because this task consists of binary classification on an evenly split dataset and every test document is assigned a class, simple accuracy is the most appropriate measure of performance.

*Proceedings of the Third Joint Conference on Lexical and Computational Semantics (*SEM 2014)*, pages 22–29,
Dublin, Ireland, August 23-24 2014.

ods with bag-of-words features as a baseline, and for classification of longer documents like the ones in our dataset, such features remain a powerful tool of analysis. Wang and Manning (2012) compare the performance of several machine learning algorithms using uni- and bigram features from a variety of common sentiment datasets, including the IMDB set used in this project. They report that that SVM classifiers generally perform better sentiment classification on paragraph-length reviews, while Native Bayes classifiers produce better results for "snippets," or short phrases (ibid., p. 91). For our dataset, they obtain the highest accuracies using a hybrid approach, SVM with Naive Bayes features, which results in 91.22% accuracy (ibid., p. 93). This appears to be the best test result to date on this dataset. Although we use a Naive Bayes classifier in our project, alternative machine learning algorithms are a promising topic of further future investigation (see §6).

Two existing areas of research have direct relevance to this project: MRS feature extraction, and sentiment analysis using features based on deep linguistic representations of data. In their work on machine translation, Oepen et al. (2007) define a type of MRS triple based on elementary dependencies, a simplified "variable-free" representation of predicate-argument relations in MRS (p. 5). Fujita et al. (2007) and Pozen (2013) develop similar features for HPSG parse selection, and Pozen experiments with replacing segments of predicate values in triple features with WordNet sense, POS, and lemma information (2013, p. 32).

While there has not yet been any research on using MRS features in sentiment analysis, there has been work on extracting features from deep representations of data for sentiment analysis. In working with deep representations such as MRSes or dependency parses, there are myriad sub-graphs that can be used as features. However these features are often quite sparse and do not generalize well. Joshi & Rose (2009) improve performance of a sentiment classifier by incorporating triples consisting of words and grammatical relations extracted from dependency parses. To increase the generalizability of these triples, they perform back-off by replacing words with part-of-speech tags. Similarly, Arora et al. (2010) extract features from dependency parses by using sentiment back-off to identify potentially meaningful portions of the dependency graph. Given this success combining back-off with sub-graph features, we design several feature types following a similar methodology.

2.1 The IMBD Dataset

We use a dataset of 50,000 movie reviews crawled from the IMDB website, originally developed by Maas et al. (2011). The dataset is split equally between training and test sets. Both training and test sets contain equal numbers of positive and negative reviews, which are defined according to the number of stars assigned by the author on the IMBD website: one to four stars for negative reviews, and seven to ten stars for positive reivews. The reviews vary in length but generally contain between five and fifteen sentences. The Natural Language ToolKit's (NLTK; Loper and Bird, 2002) sentence tokenizer distinguishes 616,995 sentences in the dataset.

Unlike previous research over this dataset, we divide the 25,000 reviews of the test set into two development sets and a final test set. As such, our results are not directly comparable to those of Wang & Manning (2012).

2.2 The Baseline System

The system we use as a baseline, created by Narayanan et al. (2013), implements several small but innovative improvements to a simple Naive Bayes classifier. In the training phase, the baseline performs simple scope of negation annotation on the surface string tokens. Any word containing the characters `not`, `n't` or `no` triggers a "negated" state, in which all following n-grams are prepended with `not_`. This continues until either a punctuation delimiter (?.,!:;) or another negation trigger is encountered.

During training, when an n-gram feature is read into the classifier, it is counted toward $P(f|c)$, and the same feature with `not_` prepended is counted toward $P(f|\hat{c})$, where c is the document class and $\hat{c}$ is the opposite class. Singleton features are then pruned away. Finally, the system runs a set of feature-filtering trials, in which the pruned features are ranked by mutual information score. These trials start at a base threshold of 100,000 features, and the number of features is increased stepwise in increments of 50,000. The feature set that produces the highest accuracy in trials over a development data set is then retained and used to classify the test data. Table 1 shows the ten most informative features, ranked by mutual informa-

Top N-Grams	
1. worst	6. awful
2. bad	7. great
3. not_the worst	8. waste
4. the worst	9. excellent
5. not_worst	10. not_not_even

Table 1: Top MI-ranked baseline n-gram Features.

tion score, out of the 12.1 million n-gram features generated by our baseline.

Before modifying the baseline system's code, we reproduced their reported accuracy figure of 88.80% over the entire 25,000 review test set. However, it appears the baseline system used the test data as development data. In order to address this, we split the data as into development sets as described above. When we ran the baseline system over our final test set, we obtained accuracies of 88.34% pre-feature filtering and 88.29% post-feature filtering; our division of the original test set into development and test sets accounts for this discrepancy.

3 Methodology

Our approach to this task consisted of three general stages: obtaining MRSes for the dataset, implementing a feature pipeline to process the MRSes, and integrating the new features into the classifier. In this section we will describe each of these processes in turn.

3.1 Parsing with the ERG

Because most of the reviews in our data set appear to be written in Standard English, we perform minimal pre-processing before parsing the dataset with the ERG. We use NLTK's sentence tokenization function in our pipeline, along with their HTML-cleaning function to remove some stray HTML-style tags we encountered in the data.

To obtain MRS parses of the data, we use ACE version 0.9.17, an "efficient processor for DELPH-IN HPSG grammars."[2] ACE's simple command line interface allows the parsing pipeline to output MRS data in a single line to a separate directory of MRS data files. We used the 1212 ERG grammar image[3] and specified root conditions that would allow for parses of the informal and fragmented sentences sometimes found in our dataset: namely, the root_informal, root_frag and root_inffrag ERG root nodes.

Parsing with these conditions resulted in 81.11% coverage over the entire dataset. After manual inspection of sentences that failed to parse, we found that irregularities in spelling and punctuation accounted for the majority of these failures and further cleaning of the data would yield higher coverage.

3.2 Feature Design

Our main focus in feature design is capturing relevant semantic relationships between sentiment terms that extend beyond the trigram boundary. Our entry point into the MRS is the elementary predication (EP), and our pipeline algorithm explores the three main EP components: arguments and associated variables, label, and predicate symbol. We also use the set of handle constraints in crawling the links between EPs.

We use two main categories of crawled MRS features: Predicate-Relation-Predicate (PRP) triples, a term borrowed from (Pozen, 2013), and Shared-Label (SL) features. Our feature template consists of eight feature subtypes, including plain EP symbols (type 1), five PRP features (types 2 through 6) and two SL features (types 7 and 8). Table 2 gives examples of each type, along with the unpruned counts of distinct features gathered from our training data. The examples for types 1 through 6 are taken from the abridged MRS example in Figure 1. Note that an *&* character separates predicate and argument components in the feature strings. The type 7 and 8 examples are taken from MRS of sentences featuring the phrases *successfully explores* and *didn't flow well*, respectively.

In our feature extraction pipeline, we use Goodman's pyDelphin[4] tool, a Python module that allows for easy manipulation and querying of MRS constituents. This tool allows our pipeline to quickly process the ERG output files, obtain argument and handle constraint information, and output the features for each MRS into a feature file to be read by our classifier. If the grammar has not returned an analysis for a particular sentence, the

[2] Available at http://sweaglesw.org/linguistics/ace/. Accessed January 15, 2014.

[3] Available at http://www.delph-in.net/erg/. Accessed January 15, 2014.

[4] Available at https://github.com/goodmami/pydelphin. Accessed January 20, 2014.

There is nothing redeeming about this trash.

```
 [LTOP: h0
INDEX:e2 [e SF:prop TENSE:pres MOOD:indicative PROG:- PERF:-]
<[_be_v_there_rel<6:8> LBL:h1 ARG0:e2 ARG1:x4] [thing_rel<9:16> LBL:h5 ARG0:x4] [_no_q_rel<9:16>
LBL:h6 ARG0:x4 RSTR:h7 BODY:h8] ["_redeem_v_for_rel"<17:26> LBL:h5 ARG0:e9 ARG1:x4 ARG2:x10]
["_about_x_deg_rel"<27:32> LBL:h11 ARG0:e12 ARG1:u13] [_this_q_dem_rel<33:37> LBL:h11 ARG0:x10 RSTR:h14
BODY:h15] ["_trash_n_1_rel"<38:44> LBL:h16 ARG0:x10]>
HCONS: <h0 qeq h1 h7 qeq h5 h14 qeq h16>]
```

Figure 1: Sample abridged MRS, with mood, tense, and other morphosemantic features removed. Each EP is enclosed in square brackets, bold type denotes predicate values.

Type	Description	Example	Count
1	Pred value	`_no_q_rel`	4,505,389
2	PRP: all	`_no_q_rel&RSTR&"_redeem_v_for_rel"`	10,255,021
3	PRP: string preds only	`"_redeem_v_for_rel"&ARG2&"_trash_n_1_rel"`	941,831
4	PRP: first pred back-off	`"_POS_v__rel"&ARG2&"_trash_n_1_rel"`	635,047
5	PRP: seond pred back-off	`"_redeem_v_for_rel"&ARG2&"_NEG_n__rel"`	621,929
6	PRP: double back-off	`"_POS_v__rel"&ARG2&"_NEG_n__rel"`	20,962
7	SL: handle not a `neg_rel` arg	`"_successful_a_1_rel"&"_explore_v_1_rel"`	589,887
8	SL: handle a `neg_rel` arg	`neg_rel&"_flow_v_1_rel"&"_well_a_1_rel"`	43,427

Table 2: Sample features (Note: Types 1 - 6 are taken from the MRS in Figure 1)

pipeline simply does not output any features for that sentence.

3.2.1 MRS Crawling

In their revisiting of the 2012 SEM scope of negation shared task, Packard et al. (2014) improve on the previous best performance using a relatively simple set of MRS crawling techniques. We make use of two of these techniques, “argument crawling” and “label crawling” in extracting our PRP and SL features (ibid., p. 3). Both include selecting an “active EP” and adding to its scope all EPs that conform to certain specifications. Argument crawling selects all EPs whose distinguished variable or label is an argument of the active EP, while label crawling adds EPs that share a label with the active EP (ibid., p. 3).

Our features are constructed in a similar fashion; for every EP in an MRS, the pipeline selects all EPs linked to the current EP and constructs features from this group of “in-scope” EPs. PRP and SL features are obtained through one “layer” of argument and label crawling, respectively. After observing a number of noisy and uninformative features in our preliminary feature vectors, we excluded a small number of EPs from being considered as the “active EP” in our pipeline algorithm: `udef_q_rel`, `proper_q_rel`, `named_rel`, `pron_rel`, and `pronoun_q_rel`. More information about what exactly these EPs represent can be found in Copestake et al. (2005).

3.2.2 PRP Features

These feature types are a version of the dependency triple features used in Oepen et al. (2007) and Fujita et al. (2007). We define the linking relation as one in which the value of any argument of the first EP matches the distinguished variable or label of the second EP. For handle variables, we count any targets of a qeq constraint headed by that variable as equivalent. We use the same set of EP arguments as Pozen (2013) to link predicates in our PRP features: `ARG`, `ARG1-N`, `L-INDEX`, `R-INDEX`, `L-HANDL`, `R-HANDL`, and `RESTR` (p. 31).

We also use a set of negative and positive word lists from the social media domain, developed by Hu and Liu (2004), for back-off replacement in PRP features. Our pipeline algorithm attempts back-off replacement for all EPs in all PRP triples. If the surface string portion of the predicate value

Feature Types	Pre-Feature Filtering	Post-Feature Filtering
baseline (n-grams only)	88.337	88.289
1	88.289	88.517
2	87.857	87.809
3	88.589	88.757
4	88.673	88.757
5	**88.709**	**88.817**
6	88.337	88.301
7	88.193	88.205
8	88.361	88.265

Table 3: Individual MRS feature trial results

Feature Types	Pre-Feature Filtering	Post-Feature Filtering
baseline (n-grams only)	88.337	88.289
n-grams with back-off	87.293	87.503
MRS only (all types)	88.253	87.977
n-grams, **4**, **5**	88.709	88.781
n-grams, **3**, **4**, **5**, 7, 8	**88.961**	88.853
n-grams, 1, **4**, **5**	88.637	88.865
n-grams, **3**, **4**, **5** 8	88.853	88.961
n-grams, **3**, **4**, **5**, 7	88.889	88.973
n-grams, 1, **3**, **4**, **5**	88.793	89.021
n-grams, **3**, **4**, **5**	88.865	**89.093**

Table 4: Combination feature results

matches any of the entries in the lexicon, the pipeline produces a back-off predicate value by replacing that portion with NEG or POS and stripping the sense category marker. These replacements appear in various positions in feature types 4, 5, and 6 (see Table 2).

3.2.3 SL Features

To further explore the relationships in the MRS, we include this second feature category in our feature template, which links together EPs that share a handle variable. We limit SL features to groups of EPs linked by a handle variable that is also an argument of another EP, or the target of a qeq constraint of such a variable. Our pipeline is therefore able to extract both PRP and SL features in a single pass through the arguments of each EP. Feature type 7 consists of shared-label groupings of two or more EPs, where the handle is not the ARG1 of a neg_rel EP. Type 8 includes groups of one or more EPs where the handle is a neg_rel argument, with neg_rel prepended to the feature string.

Features of type 7 tend to capture relationships between modifiers, such as adverbs and adjectives, and modified entities. Features of type 8 were intended to provide some negation information, though our goals of more fully analyzing scope of negation in our dataset remain unrealized at this point. We reasoned that the lemmatization of string predicate values might provide some useful back-off for the semantic entities involved in negation and modification.

4 Evaluation

To test our MRS features, we adapted our baseline to treat them much like the n-gram features. As with n-grams, each MRS feature is counted toward the probability of the class of its source document, and a negated version of that feature, with not_ prepended, is counted toward the opposite class. We ran our feature filtering trials using the first development set, then obtained preliminary accuracy figures from our second development set. We began with each feature type in isolation and used these results to inform later experiments using combinations of feature types. The numbers reported here are the results over the final, held-out test set.

Our final test accuracies indicate that three feature types produce the best gains in accuracy: back-off PRPs with first- and second-predicate replacement (types 4 and 5), and PRPs with string predicates only (type 3). Table 3 displays isolated feature test results, while Table 4 ranks the top seven feature combinations in ascending order by post-feature filtering accuracies. The bolded feature types show that all of the best combination runs include one or more of the top three features mentioned above. Notable also are the accuracies for MRS-based features alone, which fall very close to the baseline. The best accuracies for pre- and post-feature filtering tests appear in bold.

The highest accuracy, achieved by running a feature-filtered combination of the baseline's n-gram features and feature types 3, 4, and 5, resulted in a 0.80% increase over the baseline performance with feature filtering, and a 0.76% increase in the best baseline accuracy overall (obtained without feature filtering). The experimental best run successfully categorizes 63 more of the 8333 test documents than the baseline best run. Although these gains are small, they account for a 6.48% reduction in error.

Most Informative MRS Features
`not_"_NEG_a__rel"&ARG1&_the_q_rel`
`"_NEG_a__rel"&ARG1&_the_q_rel`
`"_POS_a__rel"&ARG1&_the_q_rel`
`not_"_POS_a__rel"&ARG1&_the_q_rel`
`"_POS_a__rel"&ARG1&_a_q_rel`
`not_"_POS_a__rel"&ARG1&_a_q_rel`
`not_"_NEG_a__rel"&ARG1&"_movie_n_of_rel"`
`"_NEG_a__rel"&ARG1&"_movie_n_of_rel"`
`_a_q_rel&RSTR&"_POS_a__rel"`
`not__a_q_rel&RSTR&"_POS_a__rel"`
`not_"_NEG_a__rel"&ARG1&udef_q_rel`
`"_NEG_a__rel"&ARG1&udef_q_rel`
`superl_rel&ARG1&"_POS_a__rel"`
`not_superl_rel&ARG1&"_POS_a__rel"`
`_and_c_rel&LHNDL&"_POS_a__rel"`

Table 5: Most informative MRS features

5 Discussion

5.1 The Most Successful Experiments

The test accuracies indicate that our back-off replacement method, in combination with the simple predicate-argument relationships captured in PRP triples, is the most successful aspect of feature design in this project. However, as our error analysis indicates, back-off is the likely source of many of our system's errors (see §5.2). Table 5 lists the 15 most informative MRS features from our best run based on mutual information score, all of which are of feature type 4 or 5. Note that the `not_` prepended to some features is a function of way our classifier reads in binary features (as described in §2.2), not an indication of grammatical negation. The success of these partial back-off features confirms our intuition that the semantic relationships between sentiment-laden terms and other entities in the sentence offer a reliable indicator of author sentiment. When we performed back-off replacement directly on the surface strings and ran our classifier with n-grams only, we obtained accuracies of 87.29% pre-feature filtering and 87.50% post-feature filtering, a small decrease from the baseline performance (see Table 4). This lends additional support to the idea that the *combination* of sentiment back-off and semantic dependencies is significant. These results also fit with the findings of of Joshi and Rose (2009), who determined that back-off triple features provide "more generalizable and useful patterns" in sentiment data than lexical dependency features alone (p. 316).

Despite these promising results, we found that the separate EP values (type 1), PRP triples without replacement (type 2), PRPs with double replacement (type 6) and SL features (types 7 and 8) have very little effect on accuracy by themselves. For type 1, we suspect that EP values alone don't contribute enough information beyond basic n-gram features. We had hypothesized that the lemmatization in these values might provide some helpful back-off. However, this effect is likely drowned out by the lack of any scope of negation handling in the MRS features.

We attribute the failure of the SL features to the fact that they often capture EPs originating in adjacent tokens in the surface string, which does not improve on the n-gram features. Lastly, we believe the relative sparsity of double back-off features was the primary reason they did not produce meaningful results.

These results also call into question the usefulness of the feature filtering trials in our baseline. By design, these trials produce performance increases on the dataset on which they are run. However, filtering produces small and inconsistent gains for the final held-out test set.

Error Types	
Misleading back-off	31
Plot summary / Noise	20
Obscure Words / Data Sparsity	7
Data Error	3
Nonsensical Review	3
Reason Unsure	40

Table 6: Error types from top MRS experiment

5.2 Error Analysis

We manually inspected the 104 reviews from the final test set that were correctly classified by the best run of the baseline system but incorrectly classified by the best run of our improved system. This set contains 50 false negatives, and 54 false positives. We classified them according to five subjective categories: misleading back-off, in which many of the sentiment terms have a polarity opposite to the overall review; excess plot sum-

	Incorrectly classified	**Correctly classified**
Negative docs	"_POS_a__rel"&ARG1&_the_q_rel	"_NEG_n__rel"&ARG0&udef_q_rel
	"_POS_a__rel"&ARG1&_a_q_rel	"_NEG_a__rel"&ARG1&_the_q_rel
	"_POS_a__rel"&ARG1&udef_q_rel	"_NEG_a__rel"&ARG1&udef_q_rel
	"_NEG_n__rel"&ARG0&udef_q_rel	"_POS_a__rel"&ARG1&_a_q_rel
	_a_q_rel&RSTR&"_POS_a__rel"	_the_q_rel&RSTR&"_NEG_n__rel"
Positive docs	"_NEG_n__rel"&ARG0&udef_q_rel	"_POS_a__rel"&ARG1&_a_q_rel
	"_NEG_v__rel"&ARG1&pronoun_q_rel	"_POS_a__rel"&ARG1&_the_q_rel
	"_NEG_v__rel"&ARG1&pron_rel	_a_q_rel&RSTR&"_POS_a__rel"
	"_NEG_a__rel"&ARG1&udef_q_rel	"_NEG_n__rel"&ARG0&udef_q_rel
	"_NEG_a__rel"&ARG1&_the_q_rel	"_POS_a__rel"&ARG1&udef_q_rel

Table 7: Most frequent features in test data by polarity and classification result

mary or off-topic language; use of obscure words not likely to occur frequently in the data; miscategorization in the dataset; and confusing or nonsensical language. The counts for these categories appear in Table 6.

The prevalence of errors in the first category is revealing, and relates to certain subcategories of review that confound our sentiment back-off features. For horror films in particular, words that would generally convey negative sentiment (*creepy*, *horrible*, *gruesome*) are instead used positively. This presents an obvious problem for sentiment back-off, which relies on the assumption that words are generally used with the same intent.

To explore this further, we collected counts of the most frequent features in these 104 reviews, and compared them to feature counts for correctly classified documents of the same class. The stark contrast between the back-off polarities of the features extracted and the polarity of the documents suggests that these feature types are overgeneralizing and misleading the classifier (see Table 7). While the course-grained polarity of sentiment terms is often a good indicator of overall review polarity, our system has difficulty with cases in which many sentiment terms do not align with the review sentiment. Our back-off PRP features do not include scope of negation handling, so even if these terms are negated, our classifier in its current form is unable to take advantage of that information.

Further manual observation of the feature vectors from these documents suggests that the sentiment lexicon contains elements that are not suited to the movie review domain; *plot*, for example is classified as a negative term. These results point to the need for a more domain-specific sentiment lexicon, and perhaps additional features that look at the combination of sentiment terms present in a review. LDA models could provide some guidance in capturing and analyzing co-occurring groups of sentiment terms.

6 Conclusions and Future Work

Our attempt to improve binary sentiment classification with MRS-based features is motivated by a desire to move beyond shallow approaches and explore the potential for features based on semantic dependencies. Our preliminary results are promising, if modest, and point to back-off replacement as a useful tool in combination with the relationships captured by predicate triples.

There are a number of potential areas for improvement and further development of our approach. In light of Wang and Manning's (2012) results using an SVM classifier on the same dataset, one obvious direction would be to experiment with this and other machine learning algorithms. Additionally, the ability to account for negation in the MRS features types as in Packard et al. (2014) would likely mitigate some of the errors caused by the back-off PRP features

Another possibility for expansion would be the development of features using larger feature subgraphs. Because of concerns about runtime and data sparsity, we crawl only one level of the MRS and examine a limited set of relationships. The success of Socher et al.'s (2013) Recursive Neural Tensor Network suggest that with enough data, it is possible to capture the complex compositional effects of various sub-components. Given their success with syntactic dependencies, and the research presented here, we believe semantic dependencies will be a fruitful avenue for future research in sentiment analysis. This project has been an exciting first step into uncharted territory, and suggests the potential to further exploit the MRS in sentiment analysis applications. Nonetheless, the performance gains we were able to observe demonstrate the power of using semantic representations produced by a linguistically motivated, broad-coverage parser as an information source in a semantically sensitive task such as sentiment analysis.

Acknowledgments

Thanks to our professor, Emily Bender, for providing her expertise and guidance at all stages of this project.

We're grateful to Michael Goodman for making his pyDelphin module freely available, guiding us in using it, and providing timely troubleshooting and support via email for the duration of this project. Thanks to Woodley Packard, who provided helpful advice on getting the best use out of ACE.

References

S. Arora, E. Mayfield, C Penstein-Rosé, and E Nyberg. 2010. Sentiment Classification using Automatically Extracted Subgraph Features. In *Proceedings of the NAACL HLT 2010 Workshop on Computational Approaches to Analysis and Generation of Emotion in Text,* pp. 131 - 139. Los Angeles, CA.

A. Copestake, D. Flickinger, C. Pollard, and I. A. Sag. 2005. Minimal Recursion Semantics: An Introduction. *Research on Natural Language and Computation*, 3(4), pp. 281 - 332.

D. Flickinger. 2000. On building a more efficient grammar by exploiting types. *Natural Language Engineering,*, 6(1), pp. 15 - 28.

S. Fujita, F. Bond, S. Oepen, T. Tanaka. 2010. Exploiting semantic information for HPSG parse selection. *Research on Language and Computation*. 8(1): 1-22

M. Hu and B. Liu. 2004. Mining and Summarizing Customer Reviews. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD-2004).* Seattle, WA.

M. Joshi and C Penstein Rosé. 2009. Generalizing Dependency Features for Opinion Mining. In *Proceedings of the ACL-IJCNLP 2009 Conference Short Papers,* pp. 313 - 316. Suntec, Singapore.

E. Loper, and S., Bird. 2002. NLTK: The Natural Language Toolkit. In Proceedings of the ACL Workshop on Effective Tools and Methodologies for Teaching Natural Language Processing and Computational Linguistics. Philadelphia: Association for Computational Linguistics

L. Jia, C. Yu, and W. Meng. 2009. The effect of negation on sentiment analysis and retrieval effectiveness. In *Proceedings of the 18th ACM conference on Information and knowledge management (CIKM '09)*, pp. 1827 - 1830. Hong Kong, China.

A. L. Maas, R. E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and C. Potts. 2011. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pp. 142 - 150. Portland, Oregon.

V. Narayanan, I. Arora, and A. Bhatia. 2013. Fast and accurate sentiment classification using an enhanced Naive Bayes'model. *Intelligent Data Engineering and Automated Learning IDEAL function Lecture Notes in Computer Science*, 8206:194 - 201.

S. Oepen, E. Velldal, J. Lonning, P. Meurer, V. Rosn, and D. Flickinger. 2007. Towards Hybrid Quality Oriented Machine Translation. In *Proceedings of the 11th International Conference on Theoretical and Methodological Issues in Machine Translation.*

W. Packard, E. M. Bender, J. Read, S. Oepen and R. Dridan. 2014. Simple Negation Scope Resolution Through Deep Parsing: A Semantic Solution to a Semantic Problem. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*. Baltimore, MD.

B. Pang and L. Lee. 2008. Opinion Mining and Sentiment Analysis. *Foundations and Trends in Information Retrieval*, 2(1-2):1 - 135.

Z. Pozen. 2013. Using Lexical and Compositional Semantics to Improve HPSG Parse Selection. *Master's Thesis, University of Washington.*

R. Socher, A. Perelygin, J. Wu, J. Chuang. C. Manning, A. Ng, and C. Potts 2013. Recursive Deep Models for Semantic Compositionality Over a Sentiment Treebank In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing,* pp. 1631-1642. Seattle, WA.

S. Wang and C. D. Manning. 2012. Baselines and Bigrams: Simple, Good Sentiment and Topic Classication. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics,* pp. 90 94. Jeju, Republic of Korea.

A Yeh. 2000. More accurate tests for the statistical significance of result differences. In *Proceedings of the 18th Conference on Computational Linguistics (COLING),* pp. 147 - 153.