

Prototypical Opinion Holders: What We can Learn from *Experts* and *Analysts*

Michael Wiegand and Dietrich Klakow

Spoken Language Systems

Saarland University

D-66123 Saarbrücken, Germany

{Michael.Wiegand|Dietrich.Klakow}@lsv.uni-saarland.de

Abstract

In order to automatically extract opinion holders, we propose to harness the contexts of prototypical opinion holders, i.e. common nouns, such as *experts* or *analysts*, that describe particular groups of people whose profession or occupation is to form and express opinions towards specific items. We assess their effectiveness in supervised learning where these contexts are regarded as labeled training data and in rule-based classification which uses predicates that frequently co-occur with mentions of the prototypical opinion holders. Finally, we also examine in how far knowledge gained from these contexts can compensate the lack of large amounts of labeled training data in supervised learning by considering various amounts of actually labeled training sets.

1 Introduction

Building an opinion holder (OH) extraction system on the basis of supervised classifiers requires large amounts of labeled training data which are expensive to obtain. Therefore, alternative methods requiring less human effort are required. Such methods would be particularly valuable for languages other than English as for most other languages sentiment resources are fairly sparse.

In this paper, we propose to leverage contextual information from prototypical opinion holders (protoOHs), such as *experts* or *analysts*. We define prototypical opinion holders as common nouns denoting particular groups of people whose profession or occupation is to form and express opinions towards specific items. Mentions of these nouns are disproportionately often OHs:

1. *Experts* agree it generally is a good idea to follow the manufacturers' age recommendations.

2. Shares of Lotus Development Corp. dropped sharply after *analysts* expressed concern about their business.

Since protoOHs are common nouns they should occur sufficiently often in a large text corpus in order to gain knowledge for OH extraction. We examine different ways of harnessing mentions of protoOHs for OH extraction. We compare their usage as labeled training data for supervised learning with a rule-based classifier that relies on a lexicon of predictive predicates that have been extracted from the contexts of protoOHs. Moreover, we investigate in how far the knowledge gained from these contexts can compensate the lack of large amounts of actually labeled training data in supervised classification by considering various amounts of labeled training sets.

2 Related Work

There has been much research on supervised learning for OH extraction. Choi et al. (2005) explore OH extraction using CRFs with several manually defined linguistic features and automatically learnt surface patterns. The linguistic features focus on named-entity information and syntactic relations to opinion words. Kim and Hovy (2006) and Bethard et al. (2004) examine the usefulness of semantic roles provided by FrameNet¹ for both OH and opinion target extraction. More recently, Wiegand and Klakow (2010) explored convolution kernels for OH extraction and found that tree kernels outperform all other kernel types. In (Johansson and Moschitti, 2010), a re-ranking approach modeling complex relations between multiple opinions in a sentence is presented. Rule-based OH extraction heavily relies on lexical cues. Bloom et al. (2007), for example, use a list of manually compiled communication verbs.

¹framenet.icsi.berkeley.edu

3 Data

As a large unlabeled (training) corpus, we chose the North American News Text Corpus. As a labeled (test) corpus, we use the MPQA corpus.² We use the definition of OHs as described in (Wiegand and Klakow, 2010). The instance space are all noun phrases (NP) in that corpus.

4 Method

In this paper, we propose to leverage contextual information from prototypical opinion holders (protoOHs) by which we mean common nouns denoting particular groups of people whose profession or occupation it is to form and express opinions towards specific items. The set of protoOHs that we use are listed in Table 1. It has been created ad-hoc. We neither claim completeness nor have made any attempts to tune it to our data.

Though mentions of protoOHs are likely to present OHs, not every mention is an OH:

3. Canada offered to make some civilian *experts* available.

We try to solve this problem by exclusively looking at contexts in which the protoOH is an *agent* of some predicate. Bethard et al. (2004) state that 90% of the OHs are realized as agents on their dataset. This heuristic would exclude Sentence 3 as *some civilian experts* should be considered the *patient of make available* rather than its agent.

We use grammatical dependencies from a syntactic parser rather than the output of a semantic parser for the detection of agents as in our initial experiments with semantic parsers the detection of agents of predicate adjectives and nouns was deemed less reliable. The grammatical dependency relations that we consider implying an agent are illustrated in the left half of Table 2.

We consider two different methods for extracting an OH from the contexts of protoOHs: supervised learning and rule-based classification.

4.1 Supervised Learning

The simplest way of using the contexts of agentive protoOHs is by using supervised learning. This means that on our unlabeled training corpus we consider each NP with the head being an agentive protoOH as a positive data instance and all the remaining NPs occurring in those sentences as negative instances. With this definition we train

advocate, agitator, analyst, censor, consultant, critic, defender, demonstrator, examiner, expert, inspector, marketer, observer, opponent, optimist, pessimist, proponent, referee, respondent, reviewer, supporter, surveyor
--

Table 1: ProtoOHs considered in the experiments.

a supervised classifier based on *convolution kernels* (Collins and Duffy, 2001) as this method has been shown to be quite effective for OH extraction (Wiegand and Klakow, 2010). Convolution kernels derive features automatically from complex discrete structures, such as syntactic parse trees or part-of-speech sequences, that are directly provided to the learner. Thus a classifier can be built without the taking the burden of implementing an explicit feature extraction. We chose the best performing set of tree kernels (Collins and Duffy, 2001; Moschitti, 2006) from that work. It comprises two tree kernels based on constituency parse trees and a tree kernel based on semantic role trees. Apart from a set of sequence kernels (Taylor and Christianini, 2004), this method also largely outperforms a traditional vector kernel using a set of features that were found predictive in previous work. We exclude sequence and vector kernels in this work not only for reasons of simplicity but also since their addition to tree kernels only results in a marginal improvement. Moreover, the features in the vector kernel heavily rely on task-specific resources, e.g. a sentiment lexicon, which are deliberately avoided in our low-resource classifier as our method should be applicable to any language (and for many languages sentiment resources are either sparse or do not exist at all).

In addition to Wiegand and Klakow (2010), we have to discard the content of candidate NPs (e.g. the candidate opinion holder NP [$NP_{Cand}[NNS\ advocates]$] is reduced to [NP_{Cand}]), the reason for this being that in our automatically generated training set, OHs will always be protoOHs. Retaining them in the training data would cause the learner to develop a detrimental bias towards these nouns (our resulting classifier should detect any OH and not only protoOHs).

4.2 Rule-based Classifier

Instead of training a supervised classifier, we can also construct a rule-based classifier on the basis of the agentive protoOHs. The classifier is built on the insight that the most predictive cues for OH extraction are predicates (Wiegand and Klakow,

²www.cs.pitt.edu/mpqa/databaserelease

Learning/Extraction Phase		Rule-Based Classification	
Pattern	Example	Pattern	Example
protoOH <NSUBJ> verb	<i>Experts criticized</i> _{PRED_V} the proposal.	NP <NSUBJ> extracted verb	<i>Clinton criticized</i> _{PRED_V} Chavez.
protoOH <NSUBJ> adj	<i>Experts are critical</i> _{PRED_A} of the proposal.	NP <NSUBJ> extracted adj	<i>Clinton is critical</i> _{PRED_A} of Chavez.
protoOH <by-OBJ> verb	The proposal was <i>criticized</i> _{PRED_V} by <i>experts</i> .	NP <by-OBJ> extracted verb	Chavez was <i>criticized</i> _{PRED_V} by <i>Clinton</i> .
protoOH <by-OBJ> noun	They faced <i>criticism</i> _{PRED_N} by <i>experts</i> .	NP <by-OBJ> extracted noun	Chavez ignored the <i>criticism</i> _{PRED_N} by <i>Clinton</i> .
protoOH <POSS> noun	The <i>experts'</i> <i>criticism</i> _{PRED_N} ...	NP <POSS> extracted noun	Chavez ignored <i>Clinton's</i> <i>criticism</i> _{PRED_N} .

Table 2: Agentive patterns for finding predictive predicates (left half) and for classification (right half).

2010). We, therefore, mine the contexts of agentive protoOHs (left half of Table 2) for discriminant predicates (i.e. verbs, nouns, and adjectives). That is, we rank every predicate according to its correlation, i.e. we use *Pointwise Mutual Information*, of having agentive protoOHs as an argument. The highly ranked predicates are used as predictive cues. The resulting rule-based classifier always classifies an NP as an OH if its head is an agent of a highly ranked discriminative predicate (as illustrated in the right half of Table 2).

The supervised kernel-based classifier from §4.1 learns from a rich set of features. In a previous study on reverse engineering making implicit features within convolution kernels visible (Pighin and Moschitti, 2009), it has been shown that the learnt features are usually fairly small subtrees. There are plenty of structures which just contain one or two leaf nodes, i.e. sparse lexical information, coupled with some further structural nodes from the parse tree. These structures are fairly similar to low-level features, such as bag of words or bag of ngrams, in the sense that they are weak predictors and that there are plenty of them. For such types of features, it has been shown in both subjectivity detection (Lamov et al., 2009) and polarity classification (Andreevskaia and Bergler, 2008) that they generalize poorly across different domains. On the other hand, very few high-level features describing the presence of certain semantic classes or opinion words perform consistently well across different domains. These features can either be incorporated within a supervised learner (Lamov et al., 2009) or a lexicon-based rule-based classifier (Andreevskaia and Bergler, 2008). We assume that our rule-based classifier

based on discriminant predicates (they can also be considered as some kind of semantic class) used in combination with very common grammatical relations will have a similar impact as those high-level features used in the related tasks mentioned above. Domain-independence is also an important issue in our setting, since our training and test data originate from two different corpora (which can be considered two different domains).

4.2.1 Self-training

A shortcoming of the rule-based classifier is that it incorporates no (or hardly any) domain knowledge. In other related sentiment classification tasks, i.e. subjectivity detection and polarity classification, it has been shown that by applying self-training, i.e. learning a model with a supervised classifier trained on low-level features (usually bag of words) using the domain-specific instances labeled by a rule-based classifier, more in-domain knowledge can be captured. Thus, one can outperform the rule-based classifier (Wiebe and Riloff, 2005; Tan et al., 2008).

Assuming that the same can be achieved in OH extraction, we train a classifier with convolution kernels (=low level features) on the output of the rule-based classifier run on our target corpus. The set of labeled data instances is derived from the sentences of the MPQA corpus in which the rule-based classifier predicts at least one OH, i.e. the instances the classifier labels as OHs are used as positive instances while the remaining NPs are labeled as negative. Unlike §4.1 we do not discard the content of the candidate NPs. In these labeled training data, OHs are not restricted to protoOHs. We, therefore, assume that among the

domain-specific features the supervised classifier may learn could be useful prior weights towards some of these domain-specific NPs as to whether they might be an OH or not.

4.2.2 Generalization with Clustering and Knowledge Basis

We also examine in how far the coverage of the discriminant predicates can be increased with the usage of clustering. Turian et al. (2010) have shown that in semi-supervised learning for named-entity recognition, i.e. a task which bears some resemblance to the present task, features referring to the clusters corresponding to groups of specific words with similar properties (induced in an unsupervised manner) help to improve performance.

In the context of our rule-based classifier, we augment the set of discriminant predicates by all words which are also contained in the cluster associated with these discriminant predicates. Hopefully, due to the strong similarity among the words within the same cluster, the additional words will have a similar predictiveness as the discriminant predicates. Unlike our extraction phase for OH extraction in which only the correlation between predicates and protoOHs are considered (Table 2), we may find additional predicates as the clustering is induced from completely unrestricted text.

The extension of discriminant predicates can also be done by taking into account manually built general-purpose lexical resources, such as WordNet.³ One simply adds the entire set of synonyms of each of the predicates.

4.3 Incorporation into Supervised Classifiers with Actually Labeled Data

We also want to investigate the effectiveness of the knowledge from our rule-based classifier that has been learned on the unlabeled corpus (§4.2) in supervised learning using actually labeled training data from our target corpus, i.e. the MPQA corpus. In particular, we will examine in how far this knowledge (when used as a feature in supervised learning) can compensate the lack of a sufficiently large labeled training set. For that experiment the labeled corpus, i.e. MPQA corpus, will be split into a training set and a test set.

Again, we use the supervised learner based on tree kernels (§4.1). We also augment the tree kernels themselves with additional information by

³wordnet.princeton.edu

following Wiegand and Klakow (2010) who add for each word that belongs to a predictive semantic class another node that directly dominates the pertaining leaf node and assign it a label denoting that class. While Wiegand and Klakow (2010) made use of manually built lexicons, we use our predictive predicates extracted from contexts of protoOHs. For instance, if *doubt* is such a predicate, we would replace the subtree $[VBP\ doubt]$ by $[VBP\ [PRED_{OH}\ doubt]]$. Moreover, we devise a simple vector kernel incorporating the prediction of the rule-based classifier. All kernels are combined by plain summation.

5 Experiments

The documents were parsed using the Stanford Parser.⁴ Semantic roles were obtained by using the parser by Zhang et al. (2008).

5.1 Supervised Learning

All experiments using convolution kernels were done with the *SVM-Light-TK* toolkit.⁵ We test two versions of the supervised classifier. The first considers any mention of a protoOH as an OH, while the second is restricted to only those mentions of a protoOH which are an agent of some predicate. We also experimented with different amounts of (pseudo-)labeled training data from our unlabeled corpus varying from 12500 to 150000 instances. We found that from 25000 instances onwards the classifier does not notably improve when further training data are added. The results of the classifier (using 150000 data instances) are listed in Table 3. The restriction of protoOHs to agents increases performance as expected (see §4).

5.2 The Different Rule-based Classifiers

In order to build a rule-based classifier, we first need to determine how many of the ranked predicates are to be used. This process is done separately for verbs, nouns, and adjectives. For verbs, F-Score reaches its maximum at approximately 250 which is the value we chose in our subsequent experiments. In a similar fashion, we determined 100 for both nouns and adjectives.

Table 4 lists the most highly ranked verbs that are extracted.⁶ As an indication of the intrinsic

⁴nlp.stanford.edu/software/lex-parser.shtml

⁵disi.unitn.it/moschitti

⁶The ranked predicates are available at: www.lsv.uni-saarland.de/ranlp/data.tgz

Classifier	Prec	Rec	F1	Prec	Rec	F1
<i>Supervised</i>	<i>all contexts</i>			<i>agentive contexts</i>		
	27.62	15.36	19.75	41.45	28.75	33.95
<i>Rule-based</i>	<i>without heuristics</i>			<i>with heuristics</i>		
AL	40.18	33.32	36.43	46.04	30.94	37.00
SL	35.21	34.90	35.05	49.64	31.66	38.66
AL+SL	35.00	55.36	42.89	45.16	50.65	47.75
V250	39.75	51.24	44.77	46.25	46.94	46.60
V250+A100	39.88	53.43	45.67	46.56	48.89	47.70
V250+N100	39.18	54.08	45.44	45.40	49.62	47.42
V250+A100+N100	39.31	55.93	46.17	45.71	51.57	48.47

Table 3: Performance of the different classifiers.

quality of the extracted words, we mark the words which can also be found in task-specific resources, i.e. communication verbs from the Appraisal Lexicon (AL) (Bloom et al., 2007) and opinion words from the Subjectivity Lexicon (SL) (Wilson et al., 2005). Both resources have been found predictive for OH extraction (Bloom et al., 2007; Wiegand and Klakow, 2010).

Table 3 (lower part) shows the performance of the rule-based classifiers based on protoOHs using different parts of speech. As hard baselines, the table also shows other rule-based classifiers using the same dependency relations as our rule-based classifier (see Table 2) but employing different predicates. As lexical resources for these predicates, we again use AL and SL. The table also compares two different versions of the rule-based classifier being the classifier as presented in §4.2 (left half of Table 3) and a classifier additionally incorporating the two **heuristics** (right half):

- If the candidate NP follows *according to*, then it is labeled as an OH.
- The candidate NP can only be an OH if it represents a person or a group of persons.

These are commonly accepted heuristics which have already been used in previous work as features (Choi et al., 2005; Wiegand and Klakow, 2010). The latter rule requires the output of a named-entity recognizer⁷ for checking proper nouns and WordNet for common nouns.

As far as the classifier built with the help of protoOHs is concerned, adding highly ranked adjectives and nouns consistently improves the performance (mostly recall) when added to the set of

⁷We use the Stanford tagger:
nlp.stanford.edu/software/CRF-NER.shtml

say [†]	see [†]	wonder*	recommend ^{†*}
expect*	question	complain ^{†*}	view ^{†*}
believe ^{†*}	contend*	consider*	concede*
predict*	speculate*	accuse*	attribute
agree*	point	praise ^{†*}	acknowledge*
argue*	fear ^{†*}	describe [†]	testify
call	worry*	claim ^{†*}	hope*
estimate	charge	tell	disagree*
warn [†]	forecast	change	conclude
note [†]	find [†]	cite [†]	look*
think ^{†*}	doubt*	anticipate	write
suggest ^{†*}	caution [†]	try*	criticize*

Table 4: List of verbs most highly correlating with protoOHs; [†]: included in AL; * : included in SL.

highly ranked verbs. The heuristics further improve the rule-based classifier which is achieved by notably increasing precision.

None of the baselines is as robust as the best rule-based classifier using protoOHs (i.e. V250+A100+N100). Considering our discussion in §4.2, it comes as no surprise that the best (pseudo-)supervised classifier does not perform as well as our best rule-based classifier (induced by protoOHs). The fact that, in addition to that, our proposed method also largely outperforms the rule-based classifier relying on both AL and SL when no heuristics are used and is still slightly better when they are incorporated supports the effectiveness of our method.

5.2.1 Performance of Subsets of ProtoOHs

In the previous section, we evaluated predicates often co-occurring with the entire set of protoOHs (Table 1). Therefore, we should also check how individual protoOHs or special subsets perform in order to find out whether the simple approach of considering the entire set is the optimal setting. For these experiments we use the configuration: V250+N100+A100 without heuristics.

We found that the performance of individual protoOHs varies and that the performance cannot be fully ascribed to the frequency of a protoOH with agentive contexts. For example, though *proponent* and *demonstrator* occur similarly often with those contexts, we obtain an F-Score of 44.75 when we use the predicates from the context of the former while we only obtain an F-Score of 32.70 when we consider the predicates of the latter.

We also checked whether it would be more effective to use only a subset of protoOHs and compared the performance produced by the five best protoOHs, the five most frequent protoOHs, and

Type	without heuristics			with heuristics		
	Prec	Rec	F1	Prec	Rec	F1
Baseline	39.31	55.93	46.17	45.71	51.57	48.47
+Clus	35.87	63.23	45.78	44.17	58.01	50.15
+WN	37.52	59.46	46.01	44.35	54.42	48.87
+SelfTr	39.14	62.71	48.20	44.38	59.61	50.88

Table 5: Performance of extended rule-based classifiers.

the entire set of protoOHs. The performance of the different subsets is very similar (i.e. 46.44, 46.28, and 46.17), so we may conclude that the configuration that we proposed, namely to consider all protoOHs, is more or less the optimal configuration for this method.

5.2.2 Self-training and Generalization

Table 5 shows the performance of our method when extended by either self-training (SelfTr) or generalization. For generalization by clustering (Clus), we chose Brown clustering (Brown et al., 1992) which is the best performing algorithm in (Turian et al., 2010). The clusters are induced on our unlabeled corpus (see §3). We induced 1000 clusters (optimal size). For the knowledge-based generalization (WN), we used synonyms from WordNet 3. For both Clus and WN, we display the results extending only the most highly ranked V100+N50+A50 since it provided notably better results than extending all predicates, i.e. V250+N100+A100 (our baseline). The table shows that only self-training consistently improves the results. The impact of generalization is less advantageous since by increasing recall precision drops more dramatically. Only Clus in conjunction with the heuristics manages to preserve sufficient precision.

5.3 Incorporating Knowledge from ProtoOHs into Supervised Learning

As a maximum amount of labeled training data we chose 60000 instances (i.e. NPs) which is even a bit more than used in (Wiegand and Klakow, 2010). In addition, we also test 1%, 5%, 10%, 25% and 50% of the training set. From the remaining data instances, we use 25000 instances as test data. In order to deliver generalizing results, we randomly sample the training and test partitions five times and report the averaged results.

We compare four different classifiers, a plain classifier using only the convolution kernel config-

uration from previous experiments (TKPlain), the augmented convolution kernels (TKAug) where additional nodes are added indicating the presence of an OH predicate (§4.3), the augmented convolution kernels with the vector kernel encoding the prediction of the best rule-based classifier (induced by protoOHs) without heuristics (TKAug+VK) and the classifier incorporating those heuristics (TKAug+VK[heur]). Instead of just using one feature encoding the overall prediction we use several binary features representing the occurrence of the individual groups of predicates (i.e. verbs, nouns, or adjectives) and prediction types (direct predicate or predicate from cluster extension). We also include the prediction of the self-trained classifier. The performance of these different classifiers is listed in Table 6. Recall from §4.1 that we want to examine cases in which no task-specific resources and no or few labeled training data are available. This is why the different classifiers presented should primarily be compared to our own baseline (TKPlain) and not the numbers presented in previous work as they always use the maximal size of labeled training data and additionally task-specific resources (e.g. sentiment lexicons).

The results show that using the information extracted from the unlabeled data can be usefully combined with the labeled training data. Tree augmentation causes both precision and recall to rise. This observation is consistent with (Wiegand and Klakow, 2010) where, however, AL and SL are considered for augmentation. When the vector kernel with the prediction of the rule-based classifier is also included, precision drops slightly but recall is notably boosted resulting in an even more increased F-Score. The results also show that for the setting that we have in focus, i.e. using only few labeled training data, our proposed method is particularly useful. For example, when TKPlain is as good as the best classifier exclusively built from unlabeled data (50.88% in Table 5), i.e. at 10%, there is a very notable increase in F-Score when the additional knowledge is added, i.e. the F-Score of TKAug+VK[heur] is increased by approx. 4% points. The degree of improvement towards TKPlain decreases the more labeled training data are used. However, when 100% of the labeled data are used, all of the other classifiers using additional information still outperform TKPlain.⁸

⁸The improvement is statistically significant using pair-

Training Size	TKPlain (Baseline)			TKAug			TKAug + VK			TKAug + VK[heur]		
	Prec	Rec	F1	Prec	Rec	F1	Prec	Rec	F1	Prec	Rec	F1
600 (1%)	52.14	31.49	38.63	54.18	34.44	41.52	49.60	46.74	47.38	51.47	46.63	48.20
3000 (5%)	51.69	43.80	47.39	53.17	45.92	49.27	50.68	54.48	52.50	51.40	56.84	53.97
6000 (10%)	53.31	50.39	51.78	54.22	51.91	52.99	51.13	58.33	54.46	52.14	59.55	55.57
15000 (25%)	54.75	57.96	56.31	55.52	59.08	57.24	52.96	63.76	57.86	53.02	64.46	58.18
30000 (50%)	55.14	62.69	58.66	55.82	64.06	59.65	53.40	66.89	59.38	53.02	67.75	59.91
60000 (100%)	55.94	66.80	60.88	56.68	68.56	62.05	54.60	70.30	61.46	54.92	71.30	62.04

Table 6: Performance of supervised classifiers incorporating the prediction of the rule-based classifier.

6 Conclusion

We proposed to harness contextual information from prototypical opinion holders for opinion holder extraction. We showed that mentions of such nouns when they are agents of a predicate are a useful source for automatically building a rule-based classifier. The resulting classifier performs at least as well as classifiers depending on task-specific lexical resources and can also be extended by self-training. We also demonstrated that this knowledge can be incorporated into supervised classifiers and thus improve performance, in particular, if only few labeled training data are used.

Acknowledgements

This work was funded by the German Federal Ministry of Education and Research (Software-Cluster) under grant no. "01IC10S01" and the Cluster of Excellence for Multimodal Computing and Interaction. The authors thank Alessandro Moschitti, Josef Ruppenhofer, Ines Rehbein and Yi Zhang for their technical support and interesting discussions.

References

- A. Andreevskaia and S. Bergler. 2008. When Specialists and Generalists Work Together: Overcoming Domain Dependence in Sentiment Tagging. In *ACL/HLT*.
- S. Bethard, H. Yu, A. Thornton, V. Hatzivassiloglou, and D. Jurafsky. 2004. Extracting Opinion Propositions and Opinion Holders using Syntactic and Lexical Cues. In *Computing Attitude and Affect in Text: Theory and Applications*.
- K. Bloom, S. Stein, and S. Argamon. 2007. Appraisal Extraction for News Opinion Analysis at NTCIR-6. In *NTCIR-6*.
- P. Brown, P. deSouza, R. Mercer, V. Della Pietra, and J. Lai. 1992. Class-based n-gram models of natural language. *Computational Linguistics*, 18.

- Y. Choi, C. Cardie, E. Riloff, and S. Patwardhan. 2005. Identifying Sources of Opinions with Conditional Random Fields and Extraction Patterns. In *HLT/EMNLP*.
- M. Collins and N. Duffy. 2001. Convolution Kernels for Natural Language. In *NIPS*.
- R. Johansson and A. Moschitti. 2010. Reranking Models in Fine-grained Opinion Analysis. In *COLING*.
- S. Kim and E. Hovy. 2006. Extracting Opinions, Opinion Holders, and Topics Expressed in Online News Media Text. In *ACL Workshop on Sentiment and Subjectivity in Text*.
- D. Lambov, G. Dias, and V. Noncheva. 2009. Sentiment Classification across Domains. In *EPIA*.
- A. Moschitti. 2006. Making Tree Kernels Practical for Natural Language Learning. In *EACL*.
- D. Pighin and A. Moschitti. 2009. Reverse Engineering of Tree Kernel Feature Spaces. In *EMNLP*.
- S. Tan, Y. Wang, and X. Cheng. 2008. Combining Learn-based and Lexicon-based Techniques for Sentiment Detection without using Labeled Examples. In *SIGIR*.
- J. Taylor and N. Christianini. 2004. *Kernel Methods for Pattern Analysis*. Cambridge University Press.
- J. Turian, L. Ratinov, and Y. Bengio. 2010. Word Representations: A Simple and General Method for Semi-supervised Learning. In *ACL*.
- J. Wiebe and E. Riloff. 2005. Creating Subjective and Objective Sentence Classifiers from Unannotated Texts. In *CICLing*.
- M. Wiegand and D. Klakow. 2010. Convolution Kernels for Opinion Holder Extraction. In *HLT/NAACL*.
- T. Wilson, J. Wiebe, and P. Hoffmann. 2005. Recognizing Contextual Polarity in Phrase-level Sentiment Analysis. In *HLT/EMNLP*.
- Y. Zhang, R. Wang, and H. Uszkoreit. 2008. Hybrid Learning of Dependency Structures from Heterogeneous Linguistic Resources. In *CoNLL*.

wise t-test, where $p < 0.05$.