An Enhanced Semantic Tree Kernel for Sentiment Polarity Classification

Luis A. Trindade, Hui Wang, William Blackburn, and Niall Rooney

University of Ulster, School of Computing and Mathematics,
Faculty of Computing and Engineering
Trindade-L@email.ulster.ac.uk,
{H.wang,Wt.blackburn,Nf.rooney}@ulster.ac.uk

Abstract. Sentiment analysis has gained a lot of attention in recent years, mainly due to the many practical applications it supports and a growing demand for such applications. This growing demand is supported by an increasing amount and availability of opinionated online information, mainly due to the proliferation and popularity of social media. The majority of work in sentiment analysis considers the polarity of word terms rather than the polarity of specific senses of the word in context. However there has been an increased effort in distinguishing between different senses of a word as well as their different opinion-related properties. Syntactic parse trees are a widely used natural language processing construct that has been effectively employed for text classification tasks. This paper proposes a novel methodology for extending syntactic parse trees, based on word sense disambiguation and context specific opinion-related features. We evaluate the methodology on three publicly available corpuses, by employing the sub-set tree kernel as a similarity function in a support vector machine. We also evaluate the effectiveness of several publicly available sense specific sentiment lexicons. Experimental results show that all our extended parse tree representations surpass the baseline performance for every measure and across all corpuses, and compared well to other state-of-the-art techniques.

Keywords: Information Retrieval, Social Media, Sentiment Analysis, Opinion Mining, Polarity Classification, Kernel Methods, Word Sense Disambiguation.

1 Introduction

Text consists of either facts or opinions. Facts are objective descriptions of entities, events and their properties; opinions are subjective expressions of people's sentiments, appraisals or feelings toward entities, events and their properties [11]. Determining the opinion contained within a piece of text is the aim of *sentiment analysis* (or *opinion mining*), which is assisted by techniques drawn from *natural language processing* (NLP), *information retrieval* (IR) and *computational linguistics* (CL).

Sentiment analysis has gained a lot of attention in recent years. This is mainly due to the many practical applications it supports. Examples include: helping companies and organizations find customer opinions of commercial products or services; tracking opinions in online forums, blogs and social networks; and helping individuals decide on which product to buy or which movie to watch.

A. Gelbukh (Ed.): CICLing 2013, Part II, LNCS 7817, pp. 50–62, 2013.
© Springer-Verlag Berlin Heidelberg 2013

This growing demand for automated sentiment analysis is supported by an increasing amount and availability of opinionated information online, mainly due to the proliferation of social media websites [11], [18]. Some of the most common tasks in sentiment analysis include: *subjectivity classification* [16]; *polarity classification* [16]; *polarity intensity classification* [17]; *feature/aspect-based sentiment analysis* [10]. These tasks can also be performed in combination, for example, one can start by classifying expressions as being either objective or subjective in nature; expressions classified as subjective can then be further classified as neutral or polar; and finally polar expressions can be classified as either positive, negative or both. Moreover polarity classification can be performed at various levels, for example: *word-level*, *phrase-level*, *sentence-level* and *document-level*. Note that classifying the sentiment of documents is a very different task from recognizing the *contextual polarity* of words and phrases, for instance, when working at the sentence level (or sub-sentence level) there is very little contextual information.

Polarity classification is commonly considered a binary text classification task, amounting to the classification of the polarity of a given piece of text as either positive or negative. *Support vector machine* (SVM) is a popular kernel method for text classification tasks [22]. Kernel methods are based on the use of a kernel function, which allows the mapping of data from the original data space into a higher dimensional feature space. The comparison of data can be done by computing the inner product in the high dimensional feature space, albeit implicitly through the so-called *kernel trick*. The choice of kernel function depends on the application and since this mapping (from data space to high dimensional feature space) is very general, kernel methods can be applied to complex structured objects such as sequences, images, graphs and textual documents [23]. This makes them well suited for structured NLP [25] and they have been applied to various tasks such as Question Answering, Summarization and Recognizing Textual Entailment. This paper focuses on *tree kernels* (TK) and explores their use for sentence (and phrase) level sentiment classification tasks. TK measure the similarity between two parse trees by aggregating the frequency of their matching sub-structures (for example in terms of subset trees or subtrees). A common approach is to consider the syntactic or dependency parse trees of two pieces of text. Advantages in the use of kernel approaches to natural language based classification, include the avoidance of complex feature engineering.

Despite recent efforts [2], [4], [5], [6], [12], [21] the majority of work in sentiment analysis still considers the polarity of word terms rather than the polarity of specific senses of the word. It is clear that different senses of a word can have different opinion-related properties, for example, the verb "kill" can mean a source of pain (e.g. these new shoes are killing me) but it can also mean overwhelm with hilarity, pleasure, or admiration (e.g. "*the comedian was so funny, he was killing me*"). This paper explores a range of features based on *word sense disambiguation* (WSD) and *sentiment lexicons* with sense specific opinion-related properties. We make use of those features to augment the syntactic parse trees used by the TKs and make them more efficient for sentiment polarity classification tasks. The features we consider are the WordNet [13] senses (defined as a concatenation of the word's lemma, its reduced part of speech (POS) tag and its sense number, see section 3.3) and their contextual polarity (processed for negation). We evaluate our extended parse tree representations on a binary text classification task, the determination of sentence level polarity for various corpuses. Our methodology surpasses the baseline performance for every measure and across all corpuses. To the best of our knowledge no previous study has considered the extensions to parse trees in the way that we do.

The rest of this paper is structured as follows. Section 2 gives a brief introduction to tree kernels and the trees and substructures they make use of. Section 3 describes the methodology as well as the text classification task and experimental setting considered for evaluation. Section 4 reports the experimental results. Section 5 concludes this paper with a discussion of the results and possible future work.

2 Tree Kernels

The main underlying idea of tree kernels is to compute the number of common substructures (fragments) between two trees, for example parse trees. These are usually constructed according to either the constituency parse tree or a dependency parse tree or graph. For the purposes of this paper we consider constituent syntactic parse trees. In constituent syntactic parse trees each non leaf node and its children are associated with a grammar production rule, where the symbol on the left-hand side corresponds to the parent node and the symbols on right-hand side are associated with its children (e.g. NP => DT JJ NN). These trees make the distinction between terminal and non-terminal nodes. The interior nodes are labelled by non-terminal categories of the grammar, while the leaf nodes are labelled by terminal categories. For example, Figure 1 illustrates the syntactic parse tree of an example sentence "This is not a bad movie ".

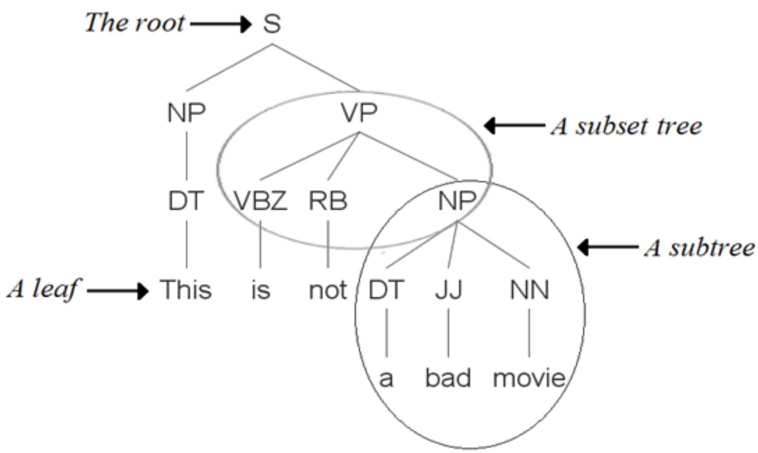

Fig. 1. Syntactic parse tree of an example sentence (“This is not a bad movie”)

2.1 Substructures

This paper considers two types of parse tree substructures, the subtrees (STs) and the subset trees (SSTs). A ST is defined as any node of a tree along with all its descendants. For example, the ST rooted in the NP node, which is circled in Figure 1. A SST is a more general structure where the leaves can be associated with non-terminal symbols. The SSTs satisfy the constraint that they follow the same grammatical rules set which generated the original tree. For example, [VP [VBZ RB NP]] is a SST of the tree in Figure 1 which has three non-terminal symbols, VBZ, RB and NP, as leaves.

Given a syntactic tree we can use the set of all its STs or SSTs as a feature representation. For instance, in the example sentence ("This is not a bad movie") there are ten STs but there are hundreds of SSTs. This substantial difference in the number of substructures between the two tree-based representations, indicates a difference in the level of information these substructures convey.

2.2 The Tree Kernel Function

The main idea of tree kernels is to compute the number of the common substructures between two trees T_1 and T_2 without explicitly considering the whole fragment space. For this purpose, Moschitti [15], slightly modified the kernel function proposed by Collins & Duffy [8] by introducing a parameter σ which enables the evaluation of the subtree kernel (STK) or the subset tree kernel (SSTK). Given the set of fragments $F = \{f_1, f_2, \dots, f_{|F|}\}$, the indicator function $x_i(n)$ is equal 1 if the target f_i is rooted at node n and 0 otherwise. Let the tree kernel function TK be defined as:

$$TK(T_1, T_2) = \sum_{n_1 \in N_{T_1}} \sum_{n_2 \in N_{T_2}} \Delta(n_1, n_2) \tag{1}$$

where N_{T_1} and N_{T_2} are the sets of the T_1's and T_2's nodes, respectively and $\Delta(n_1, n_2) = \sum_{i=1}^{|F|} x_i(n_1)\, x_i(n_2)$. This latter is equal to the number of common fragments rooted in the n_1 and n_2 nodes. Δ can be computed as follows:

1. If the productions at n_1 and n_2 are different then $\Delta(n_1, n_2) = 0$;
2. If the productions at n_1 and n_2 are the same, and n_1 and n_2 have only leaf children (meaning they are pre-terminals symbols) then $\Delta(n_1, n_2) = 1$;
3. If the productions at n_1 and n_2 are the same, and n_1 and n_2 are not pre-terminals then:

$$\Delta(n_1, n_2) = \prod_{j=1}^{nc(n_1)} (\sigma + \Delta(c_{n_1}^j, c_{n_2}^j)) \tag{2}$$

where $\sigma \in \{0,1\}$, $nc(n_1)$ is the number of the children of n_1 and c_n^j is the j-th child of the node n. Note that, since the productions are the same, $nc(n_1) = nc(n_2)$.

When σ is equal to 0, $\Delta(n_1, n_2)$ is equal to 1 only if $\forall j\, \Delta(c_{n_1}^j, c_{n_2}^j) = 1$, meaning that all the productions associated with the children are identical. From the recursive application of this property, it follows that the subtrees in n_1 and n_2 are identical. Thus, equation 1 evaluates the STK when $\sigma = 0$. When σ is equal to 1, $\Delta(n_1, n_2)$ evaluates the number of SSTs common to n_1 and n_2 as proved in Collins and Duffy [8].

The computational complexity of $TK(T_1, T_2)$ is $O(|N_{T1}| \times |N_{T2}|)$. Although this basic implementation has quadratic complexity, this scenario is quite unlikely for the syntactic trees of natural language sentences, as Collins & Duffy [8] noted. In practice it is possible to design algorithms that run in linear time on average [15]. Moschitti has implemented these algorithms, effectively encoding the STK and SSTK in a popular SVM (SVM-light), and made them freely available online (http://disi.unitn.it/moschitti/Tree-Kernel.htm).

3 Methodology

This paper presents a novel methodology for enriching (syntactic) parse trees with WSD and sense specific opinion-related properties, in order to improve the effectiveness of TKs for polarity classification tasks. It explores a range of features, used separately or in combination, to extend the leaf nodes (words) of syntactic parse trees with the corresponding WordNet senses and/or their contextual polarity (processed for negation). Note that this makes the extended features the new leaf nodes in the parse tree. For a visual interpretation see Figure 2 below.

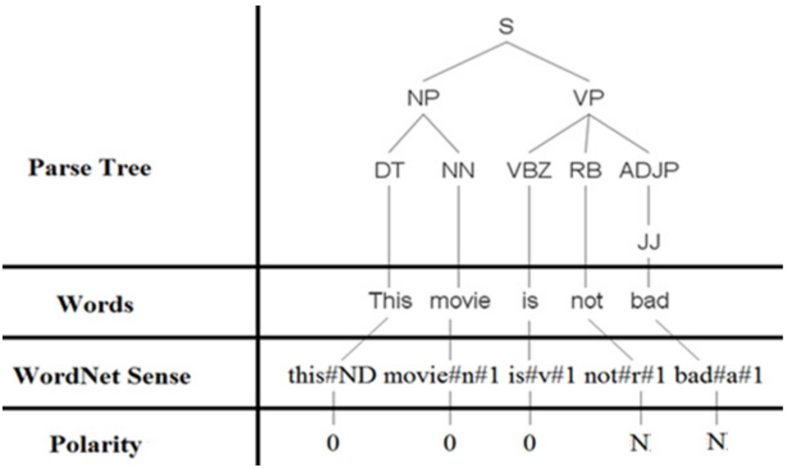

Fig. 2. Example of a parse tree extended with WordNet senses and polarity

This allows the kernel to not only match the surface words (at the leaf node) but also senses of the words as well as the polarity of the word senses. The idea is that having a set of features that is tailored for the task will increase the overall performance.

3.1 Sentence/Phrase Level Sentiment Polarity Corpuses

To evaluate our approach we compare our extended trees with the plain syntactic parse trees on a binary text classification task, the determination of whether a sentence/phrase expresses a positive or negative sentiment. We conduct a series of

10-fold cross-validation tests on three publicly available corpuses from different domains.,namely:

- *Movie Reviews corpus* (sentence polarity dataset v1.0) [17] – This corpus contains 5331 positive and 5331 negative processed sentences/snippets taken from several movie reviews.
- *SemEval-2007 Affective Task corpus* [24] – This corpus contains 1000 positive and 1000 negative news headlines, extracted from news web sites (such as Google news and CNN) and/or newspapers.
- *Mixed Product Reviews* [26] – This corpus contains 923 positive and 1320 negative sentences. These sentences are extracted from 294 product reviews from various online sources, manually annotated with sentence level sentiment.

3.2 Word Sense Disambiguation

We start by obtaining the syntactic parse trees for each sentence/phrase in the corpuses using the Stanford CoreNLP package (nlp.stanford.edu/software/corenlp.shtml). We then perform WSD with a WordNet-based method (WordNet::SenseRelate::AllWords [20]) in order to obtain the WordNet sense corresponding to the words in the corpuses. We choose the same combination of parameters that achieved the best result reported in [20], using the Lesk measure [19] as the similarity function, which tends to result in much higher recall, (since it is able to measure the similarity between words with any POS); and a window size of 15 (the number of words, to be taken into consideration when performing the WSD). In order to increase the compatibility of the sentences in the corpuses with WordNet::SenseRelate::AllWords, we replace contracted expressions with their full version (e.g. “won’t” replaced with “will not”).

3.3 Sentiment Lexicons

Despite recent efforts, most work still makes use of the words’ prior polarity in order to classify the polarity of sentences or documents. Often overlooking the fact that the polarity of a word depends on the context in which it is expressed [28]. In order to address this issue this paper makes use of several WordNet-based sentiment lexicons that take into account the polarity of particular senses of the words. The lexicons in question are Micro-WNOp [7], Q-WordNet [1] and SentiWordNet [3], [9].

In SentiWordNet and Micro-WNOp each WordNet synset is associated polarity scores (ranging from 0 to 1) that describe how positive and negative the senses are. This paper instead assigns each WordNet sense a value based on an aggregated score (A-score = P-score – N-score) similar to the approach taken by Agerri et al. [1]. Namely assigning a:

- **P** to positive senses (A-score > 0) – e.g. true#a#2 which has a P-score of 1 and a N-score of 0;
- **N** to negative senses (A-score < 0) – e.g. cynical#a#1 which has a P-score of 0 and a N-score of 1; and
- **O** to objective and neutral senses (A-score = 0) – e.g. real#a#7 which has a P-score of 0 and a N-score of 0.

We also consider an alternative representation by assigning a **B** for senses that can have both polarities (A-score = 0, P-score ≠ 0, N-score ≠ 0, and P-score = N-score) – e.g. literal#a#1 which has a P-score of 0.25 and a N-score of 0.25. This alternative representation seems to have little to no effect in preliminary experiments, as such it is not considered for the final experiments.

We analyse the effectiveness and coverage of the polarities obtained from the different sentiment lexicons, by themselves and in combination as depicted in Table 1.

Table 1. Sentiment Lexicons Considered

Lexicon ID	Lexicon	Senses
L1	Micro-WNOp (MWN)	2800
L2	Q-WordNet (QWN)	15511
L3	SentiWordNet (SWN)	49447
CL1	Micro-WNOp + Q-WordNet (MWN + QWN)	18062
CL2	Micro-WNOp + SentiWordNet (MWN + SWN)	51001
CL3	Q-WordNet + SentiWordNet (QWN+SWN)	60738
CL4	Micro-WNOp + Q-WordNet + SentiWordNet (MWN+QWN+SWN)	62194

The polarity lexicons are in the format Lemma#ReducedPart-of-SpeechTag#SenseNumber Polarity {P or N or O (or B)}. Note that the combined lexicon QWN+SWN (CL3), for example, does not have the same meaning as SWN+QWN. QWN+SWN is generated by using the polarities in Q-WordNet as a starting point and then adding to it the polarities extracted from SentiWordNet for words that are present in SentiWordNet and not in Q-WordNet. This means that there are other possible combinations that are not featured in this table, since they proved to be less efficient. The most efficient combinations are those that give priority to the most fine-grained and smallest lexicons especially when considering SWN, for example QWN (15511) + SWN (49447) results in 60738 total unique WordNet sense polarities. This might be due to the fact that SWN was not manually annotated and some senses are misclassified, so by giving priority to the senses in MWN and QWN we reduce this negative influence.

To examine the quality and coverage of the polarities obtained from the different sentiment lexicons, prior to the final experiments, we consider a simple measure based on Turney's [27]. The total percentage of sentences in the corpuses that are positive and whose sum of polarities (of the individual WordNet senses of terms in the sentence) is more than 0, in combination with those that are negative and whose sum of polarities is less than 0 relative to the total number of examples. The lexicon that scores best using this measure is CL4 (MWN+QWN+SWN) which also offers the most coverage of the data, as broken down in Table 2.

Table 2. Polarity lexicon quality and coverage in term of the percentage of correctly classified positive (Pos), negative (Neg), overall neutral (Neu) and total examples (Tot)

Lexicon	Movie Reviews				SemEval				Mixed Reviews			
	Pos	**Neg**	**Neu**	**Tot**	**Pos**	**Neg**	**Neu**	**Tot**	**Pos**	**Neg**	**Neu**	**Tot**
L1	19.95	0.30	40.48	10.11	5.98	0.00	45.77	2.82	20.15	0.08	33.44	8.34
L2	48.09	4.78	26.47	26.4	23.93	3.23	41.45	12.98	48.00	3.18	25.32	21.62
L3	61.53	15.37	16.91	38.42	36.32	5.89	35.61	20.22	64.90	10.23	16.54	32.72
CL1	50.83	5.34	24.78	28.05	26.5	3.80	41.45	14.49	49.19	3.79	24.83	22.47
CL2	61.80	15.39	16.77	38.56	36.54	6.27	35.81	20.52	65.11	10.15	16.63	32.77
CL3	**65.69**	16.71	14.14	**41.16**	42.52	12.17	**29.48**	26.46	**67.71**	9.47	15.87	33.44
CL4	65.44	**16.82**	**14.11**	41.09	**42.52**	**12.36**	30.18	**26.56**	67.61	**9.70**	**15.74**	**33.53**

3.4 Negation Processing

It should be clear from the breakdown presented in Table 2 that even with CL4 a greater percentage of the positive examples (42-67%) are correctly classified, as opposed to a very small percentage of negative examples (9-16%). In an effort to address this issue and balance these measures, we make use of the dependencies generated by the Stanford CoreNLP, in order to process each sentence for negation, namely the dependency modifier "neg", which allows us to easily determine the presence of several simple types of negation. We found that the average number of negations per sentence greatly varies with the domain of the corpus. While the Movie Reviews and Mixed Reviews corpuses have around 1 negation every 5 sentences, the SemEval News corpus has only 1 negation every 50 sentences.

We tested different negation schemas in preliminary tests and found that the most efficient schema is when we emphasize the negation. When the negated word is positive (e.g. good) or neutral, the resulting polarity for the negating word (e.g. not) and negated word will both be negative; and when the negated word is negative, the resulting polarity for the negating word (e.g. not) and negated word (e.g. bad) will be positive. This is illustrated in the following examples:

Table 3. Feature breakdown of two example sentences, higlighting negation

Features	Sentence				
Word	This	movie	is	not	good
Word Sense	this#ND	movie#n#1	is#v#1	not#r#1	good#a#1
Polarity	O	O	O	N	P
Polarity with Negation	O	O	O	**N**	**N**
Word	This	movie	is	not	bad
Word Sense	this#ND	movie#n#1	is#v#1	not#r#1	bad#a#1
Polarity	O	O	O	N	N
Polarity with Negation	O	O	O	**P**	**P**

Processing negation offers significant improvement when the lexicon considered has a low coverage for the data, but gradually decreases in influence as the lexicon considered grows in size. This is illustrated in Table 4.

Table 4. Lexicon polarity quality and coverage with and without negation processing, in terms of the percentage of correctly classified examples

Lexicon	Movie Reviews		SemEval News		Mixed Reviews	
	Plain Polarities	With Negation	Plain Polarities	With Negation	Plain Polarities	With Negation
L1	10.11	16.06	8.34	20.91	2.80	4.30
L2	26.40	28.71	21.62	28.85	12.90	13.70
L3	38.42	39.12	32.72	35.00	20.10	20.70
CL1	28.05	29.94	22.47	29.78	14.40	15.20
CL2	38.56	39.22	32.77	34.95	20.40	21.00
CL3	41.16	**41.54**	33.44	35.13	26.30	26.60
CL4	41.09	41.43	33.52	**35.27**	26.40	**26.70**

Again the lexicon that scores best across most corpuses is CL4 (MWN+QWN+SWN), which also offers the most coverage of the data and thus is the lexicon chosen for the actual parse tree extension experiments.

3.5 Support Vector Machine

The SVM implementation chosen to run the classification tasks is SVMlight-TK 1.2 [14]. This SVM package contains the implementations of the STK and SSTK as part of it. Since we are mostly interested in comparing the performance of our extended parse trees against the plain parse trees, we leave the parameters in both the SVM and the kernels as default.

4 Experimental Evaluation

We evaluate the impact of the proposed methodology, for extending syntactic parse trees with WSD and polarity features, for polarity classification tasks. We start by evaluating the performance of the different sentiment lexicons. We also evaluate the impact of the features in separate and combination as well as the impact of negation processing. Finally we compare the performance of TKs for sentiment polarity classification compared to the other kernel based approaches. We use 10-fold cross-validation classification accuracy (%) as a measure of performance throughout our experimental evaluations. Note that early experiments revealed that the SSTK is much more accurate than the STK (by about 10%) so we decided to use only the SSTK in our final experiments. This is not surprising since the SSTK is a specialized kernel which is more appropriate to explore constituent syntactic parse trees [14].

The sentiment lexicon evaluation confirmed our initial analysis of the quality and coverage of the lexicons we consider. However, this is true only when the polarity is used in combination with the word senses.

Table 5. Sentiment lexicon evaluation - parse trees extended with WSD and polarity with and without negation processing

Lexicon	Movie Reviews		SemEval News		Mixed Reviews	
	WSD+Pol	WSD+Pol-N	WSD+Pol	WSD+Pol-N	WSD+Pol	WSD+Pol-N
L1	74.18	74.23	65.00	64.80	72.40	72.45
L2	74.18	74.24	65.00	64.90	72.40	72.36
L3	74.18	74.28	65.00	64.80	72.40	72.62
CL1	74.18	74.27	65.00	64.90	72.40	72.36
CL2	74.18	74.28	64.20	64.80	72.40	72.62
CL3	74.18	74.26	65.00	65.00	72.13	72.62
CL4	**74.19**	**74.29**	**65.00**	**65.00**	**72.40**	**72.63**

Table 6. Evaluation of our parse tree extensions

Features	Movie Reviews	SemEval News	Mixed Reviews
Tree Kernel Baseline	71.70	62.60	71.29
WSD	73.27	63.90	71.24
Polarity	73.35	64.30	72.13
Polarity with Neg	73.44	64.10	72.00
WSD + Polarity	74.19	65.00	72.40
WSD + Pol with Neg	**74.29**	**65.00**	**72.63**

Table 7. Comparison of our approach and other popular kernels for polarity classification tasks

Methodology	Movie Reviews	SemEval	Mixed Reviews
Linear / Bag of Words	50.47	54.10	59.20
TK Syntactic Parse tree	71.70	62.60	71.29
TK Extended Parse tree (WSD + Pol with Neg)	74.29	65.00	72.63
Sequence Kernel / bigrams	76.21	67.60	74.45

As we can see our parse tree extensions provide an improvement over the baseline (the syntactic parse tree with no augmentation) results across all corpuses. The results also seem to indicate that the WordNet senses and polarities are complementary features, since the improvement provided by extending the parse trees with both WordNet senses and polarities, is always larger than when these features are used to extend the parse trees separately. Furthermore negation seems to offer some benefits in most cases, especially when combined with the WSD features. Note that early experiments with the STK still show the same (or higher) improvement but the results were much lower in general. This can be attributed to the different substructures that each kernel considers.

5 Discussion

Document level and sentence level polarity classification are two very different tasks. When working at the sentence level (and sub-sentence) there is very little contextual information, leading in most cases to lower results. Furthermore the majority of the work in sentiment analysis considers the polarity of word terms rather than the polarity of specific senses of the word. It should be clear that different senses of a word can have different opinion-related properties. This paper addressed the issues of word sense and contextual polarity by making use of a novel combination of features drawn from external knowledge sources.

We evaluated three sentiment lexicons and four combinations of these. We found that the combined lexicon CL4 comprising Micro-WNop, Q-WordNet and SentiWordNet, achieves the best performance. Prior to the final experiments, we used a simple measure to analyse the quality and coverage of the polarities obtained from the different sentiment lexicons. We noticed that a great percentage of the positive examples are correctly classified (42-67%), as opposed to a very small percentage of negative examples (9-16%). We addressed this issue and managed to balance these measures, by processing each sentence for negation with the use of the dependencies generated by the Stanford CoreNLP. We also tested different negation schemas in preliminary tests and found that the most efficient schema is when we emphasize the negation. As such when the negated word is positive (e.g. good) or neutral, the resulting polarity for the negating word (e.g. not) and negated word will both be negative; and when the negated word is negative, the resulting polarity for the negating word (e.g. not) and negated word (e.g. bad) will be positive.

Note that despite WSD being reportedly only about 50-70% accurate [5], [20], [21] the experimental evaluation shows that our parse tree extensions provide an improvement over the baseline results (for all measures) across all corpuses. The improvement provided by extending the parse trees with both WordNet senses and polarities is always larger than when these features are used to extend the parse trees separately, suggesting that the features we selected are complementary. This confirms that WSD offers improvements for polarity classification tasks, however since the WSD is an intermediate task, disambiguation errors can affect the quality of the corresponding sense specific opinion-related properties and thus the classification quality. Furthermore the results indicate that our local negation processing offers some benefits, especially when combined with the WSD features. Particularly in the Movie Reviews and Mixed Reviews corpuses where there was a significant improvement in performance. This appears to relate with the number of negations in the corpuses, while the Movie Reviews and Mixed Reviews corpuses have around 1 negation every 5 sentences; the SemEval News corpus has only 1 negation every 50 sentences.

Finally, our methodology has the added benefit of working with most TKs, so advances in TKs that make use of syntactic parse trees, might be further enhanced by our extended parse trees.

Possible work for the future includes: developing different extension representations; enhancing dependency trees; developing our own unique tree representations, rather than extending parse trees; including more features (e.g. Named Entities); applying the methodology for multi-class polarity classification tasks; and adapting the methodology to document-level polarity classification.

References

1. Agerri, R., Garc, A.: Q-WordNet: Extracting polarity from WordNet senses. In: Seventh Conference on International Language Resources and Evaluation Malta (2009)
2. Akkaya, C., Wiebe, J., Conrad, A., Mihalcea, R.: Improving the impact of subjectivity word sense disambiguation on contextual opinion analysis. In: Proceedings of the Fifteenth Conference on Computational Natural Language Learning, pp. 87–96. Association for Computational Linguistics (2011)
3. Baccianella, S., Esuli, A., Sebastiani, F.: SentiWordNet 3.0: An Enhanced Lexical Resource for Sentiment Analysis and Opinion Mining. In: Proceedings of the Seventh conference on International Language Resources and Evaluation LREC 2010 European Language Resources Association (ELRA), pp. 2200–2204 (2008)
4. Balamurali, A., Joshi, A., Bhattacharyya, P.: Robust Sense-Based Sentiment Classification. In: ACL HLT 2011, p. 132 (2011)
5. Balamurali, A.R., Joshi, A., Bhattacharyya, P.: Harnessing WordNet senses for supervised sentiment classification. In: Conference on Empirical Methods in Natural Language Processing, EMNLP 2011, July 27-31, pp. 1081–1091. Association for Computational Linguistics (ACL), Edinburgh (2011)
6. Carrillo de Albornoz, J., Plaza, L., Gervás, P.: A hybrid approach to emotional sentence polarity and intensity classification. In: Proceedings of the Fourteenth Conference on Computational Natural Language Learning, pp. 153–161. Association for Computational Linguistics (2010)
7. Cerini, S., Compagnoni, V., Demontis, A., Formentelli, M., Gandini, G.: Micro-WNOp: A gold standard for the evaluation of automatically compiled lexical resources for opinion mining. In: Language Resources and Linguistic Theory: Typology, Second Language Acquisition, English linguistics, Franco Angeli Editore, Milano, IT (2007)
8. Collins, M., Duffy, N.: New ranking algorithms for parsing and tagging: Kernels over discrete structures, and the voted perceptron. In: Proceedings of the 40th Annual Meeting on Association for Computational Linguistics, pp. 263–270. Association for Computational Linguistics (2002)
9. Esuli, A., Sebastiani, F.: Sentiwordnet: A publicly available lexical resource for opinion mining. In: Proceedings of LREC, pp. 417–422. Citeseer (2006)
10. Hu, M., Liu, B.: Mining and summarizing customer reviews. In: Proceedings of the tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 168–177. ACM (2004)
11. Liu, B.: Sentiment analysis and subjectivity. In: Handbook of Natural Language Processing (2010)
12. Martın-Wanton, T., Balahur-Dobrescu, A., Montoyo-Guijarro, A., Pons-Porrata, A.: Word sense disambiguation in opinion mining: Pros and cons. In: Special Issue: Natural Language Processing and its Applications, pp. 119–130 (2010)
13. Miller, G.A.: WordNet: a lexical database for English. Communications of the ACM 38(11), 39–41 (1995)
14. Moschitti, A.: Efficient convolution kernels for dependency and constituent syntactic trees. In: Fürnkranz, J., Scheffer, T., Spiliopoulou, M. (eds.) ECML 2006. LNCS (LNAI), vol. 4212, pp. 318–329. Springer, Heidelberg (2006)
15. Moschitti, A.: Making tree kernels practical for natural language learning. In: Proceedings of EACL, pp. 113–120 (2006)

16. Pang, B., Lee, L.: A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. In: Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics, pp. 271–278. Association for Computational Linguistics (2004)
17. Pang, B., Lee, L.: Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In: Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics, pp. 115–124. Association for Computational Linguistics (2005)
18. Pang, B., Lee, L.: Opinion mining and sentiment analysis. Foundations and Trends in Information Retrieval 2(1-2), 1–135 (2008)
19. Patwardhan, S., Banerjee, S., Pedersen, T.: Using measures of semantic relatedness for word sense disambiguation. In: Gelbukh, A. (ed.) CICLing 2003. LNCS, vol. 2588, pp. 241–257. Springer, Heidelberg (2003)
20. Pedersen, T., Kolhatkar, V.: WordNet: SenseRelate: AllWords: a broad coverage word sense tagger that maximizes semantic relatedness. In: Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics, Companion Volume: Demonstration Session, pp. 17–20. Association for Computational Linguistics (2009)
21. Rentoumi, V., Giannakopoulos, G., Karkaletsis, V., Vouros, G.A.: Sentiment analysis of figurative language using a word sense disambiguation approach. In: Proc. of the International Conference RANLP, pp. 370–375 (2009)
22. Sebastiani, F.: Machine learning in automated text categorization. ACM Computing Surveys (CSUR) 34(1), 1–47 (2002)
23. Shawe-Taylor, J., Cristianini, N.: Kernel methods for pattern analysis. Cambridge University Press (2004)
24. Strapparava, C., Mihalcea, R.: Semeval-2007 task 14: Affective text. In: Proceedings of SemEval, vol. 7 (2007)
25. Suzuki, J., Hirao, T., Sasaki, Y., Maeda, E.: Hierarchical directed acyclic graph kernel: Methods for structured natural language data. In: Proceedings of the 41st Annual Meeting on Association for Computational Linguistics, vol. 1, pp. 32–39. Association for Computational Linguistics (2003)
26. Täckström, O., McDonald, R.: Discovering fine-grained sentiment with latent variable structured prediction models. In: Clough, P., Foley, C., Gurrin, C., Jones, G.J.F., Kraaij, W., Lee, H., Mudoch, V. (eds.) ECIR 2011. LNCS, vol. 6611, pp. 368–374. Springer, Heidelberg (2011)
27. Turney, P.D.: Thumbs up or thumbs down?: semantic orientation applied to unsupervised classification of reviews. In: Proceedings of the 40th Annual Meeting on Association for Computational Linguistics, pp. 417–424. Association for Computational Linguistics (2002)
28. Wiegand, M., Klakow, D.: Convolution kernels for opinion holder extraction. In: Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics, pp. 795–803. Association for Computational Linguistics (2010)