Convolution Kernels for Opinion Holder Extraction

Michael Wiegand and **Dietrich Klakow**
Spoken Language Systems
Saarland University
D-66123 Saarbrücken, Germany
{Michael.Wiegand|Dietrich.Klakow}@lsv.uni-saarland.de

Abstract

Opinion holder extraction is one of the important subtasks in sentiment analysis. The effective detection of an opinion holder depends on the consideration of various cues on various levels of representation, though they are hard to formulate explicitly as features. In this work, we propose to use convolution kernels for that task which identify meaningful fragments of sequences or trees by themselves. We not only investigate how different levels of information can be effectively combined in different kernels but also examine how the scope of these kernels should be chosen. In general relation extraction, the two candidate entities thought to be involved in a relation are commonly chosen to be the boundaries of sequences and trees. The definition of boundaries in opinion holder extraction, however, is less straightforward since there might be several expressions beside the candidate opinion holder to be eligible for being a boundary.

1 Introduction

In recent years, there has been a growing interest in the automatic detection of opinionated content in natural language text. One of the more important tasks in sentiment analysis is the extraction of opinion holders. Opinion holder extraction is one of the critical components of an opinion question-answering system (i.e. systems which automatically answer opinion questions, such as "What does [X] like about [Y]?"). Such systems need to be able to distinguish which entities in a candidate answer sentence are the sources of opinions (= opinion holder) and which are the targets.

On other NLP tasks, in particular, on relation extraction, there has been much work on *convolution kernels*, i.e. kernel functions exploiting huge amounts of features without an explicit feature representation. Previous research on that task has shown that convolution kernels, such as sequence and tree kernels, are quite effective when compared to manual feature engineering (Moschitti, 2008; Bunescu and Mooney, 2005; Nguyen et al., 2009). In order to effectively use convolution kernels, it is often necessary to choose appropriate substructures of a sentence rather than represent the sentence as a whole structure (Bunescu and Mooney, 2005; Zhang et al., 2006; Moschitti, 2008). As for tree kernels, for example, one typically chooses the syntactic subtree immediately enclosing two entities potentially expressing a specific relation in a given sentence. The opinion holder detection task is different from this scenario. There can be *several* cues within a sentence to indicate the presence of a genuine opinion holder and these cues need not be member of a particular word group, e.g. they can be opinion words (see Sentences 1-3), communication words, such as *maintained* in Sentence 2, or other lexical cues, such as *according* in Sentence 3.

1. <u>The U.S. commanders</u> consider$_{opinion}$ the prisoners to be unlawful_combatants$_{opinion}$ as opposed to prisoners of war.

2. During the summit, <u>Koizumi</u> maintained$_{communication}$ a clear-cut_collaborative_stance$_{opinion}$ towards the U.S. and emphasized that the President was objective$_{opinion}$ and circumspect.

3. According$_{cue}$ to <u>Fernandez</u>, it was the worst_mistake$_{opinion}$ in the history of the Argentine economy.

Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the ACL, pages 795–803,
Los Angeles, California, June 2010.

Thus, the definition of boundaries of the structures for the convolution kernels is less straightforward in opinion holder extraction.

The aim of this paper is to explore in how far convolution kernels can be beneficial for effective opinion holder detection. We are not only interested in how far different kernel types contribute to this extraction task but we also contrast the performance of these kernels with a manually designed feature set used as a standard vector kernel. Finally, we also examine the effectiveness of expanding word sequences or syntactic trees by additional prior knowledge.

2 Related Work

Choi et al. (2005) examine opinion holder extraction using CRFs with various manually defined linguistic features and patterns automatically learnt by the AutoSlog system (Riloff, 1996). The linguistic features focus on named-entity information and syntactic relations to opinion words. In this paper, we use very similar settings. The features presented in Kim and Hovy (2005) and Bloom et al. (2007) resemble very much Choi et al. (2005). Bloom et al. (2007) also consider communication words to be predictive cues for opinion holders.

Kim and Hovy (2006) and Bethard et al. (2005) explore the usefulness of semantic roles provided by FrameNet (Fillmore et al., 2003) for both opinion holder and opinion target extraction. Due to data sparseness, Kim and Hovy (2006) expand FrameNet data by using an unsupervised clustering algorithm.

Choi et al. (2006) is an extension of Choi et al. (2005) in that opinion holder extraction is learnt jointly with opinion detection. This requires that opinion expressions and their relations to opinion holders are annotated in the training data. Semantic roles are also taken as a potential source of information. In our work, we deliberately work with minimal annotation and, thus, do not consider any labeled opinion expressions and relations to opinion holders in the training data. We exclusively rely on entities marked as opinion holders. In many practical situations, the annotation beyond opinion holder labeling is too expensive.

Complex convolution kernels have been successfully applied to various NLP tasks, such as relation extraction (Bunescu and Mooney, 2005; Zhang et al., 2006; Nguyen et al., 2009), question answering (Zhang and Lee, 2003; Moschitti, 2008), and semantic role labeling (Moschitti et al., 2008). In all these tasks, they offer competitive performance to manually designed feature sets. Bunescu and Mooney (2005) combine different sequence kernels encoding different contexts of candidate entities in a sentence. They argue that several kernels encoding different contexts are more effective than just using one kernel with one specific context. We build on that idea and compare various scopes eligible for opinion holder extraction. Moschitti (2008) and Nguyen et al. (2009) suggest that different kinds of information, such as word sequences, part-of-speech tags, syntactic and semantic information should be contained in separate convolution kernels. We also adhere to this notion.

3 Data

As labeled data, we use the sentiment annotation of the *MPQA 2.0 corpus*[1]. Opinion holders are not explicitly labeled as such. However sources of *private states* and *subjective speech events* (Wiebe et al., 2003) are a fairly good approximation of the task. Previous work (Choi et al., 2005; Kim and Hovy, 2005; Choi et al., 2006) uses similar approximations.

4 Method

In this work, we consider all noun phrases (NPs) as possible candidate opinion holders. Therefore, the set of all data instances is the set of the NPs within the MPQA 2.0 corpus. Each NP is labeled as to whether it is a genuine opinion holder or not. Throughout this section, we will use Sentence 2 from Section 1 as an example.

4.1 The Different Levels of Representation

Several levels of representation are important for opinion holder extraction. Table 1 lists all the different levels that are used in this work. Generalized sequences employ *named-entity tags*, an OPINION tag for *opinion words* and a COMM tag for *communication words*[2]. Thus, in a generalized word se-

[1] `www.cs.pitt.edu/mpqa/databaserelease`

[2] Note that all candidate tokens are reduced to one generic CAND token. Thus, we hope to account for data sparseness in

quence (WRD_{GN}) a word is replaced by a generalized token whereas in a generalized part-of-speech sequence (POS_{GN}) a part-of-speech tag is replaced. For augmented constituent trees ($CONST_{AUG}$), the same sources of information are used. The difference to generalizing sequences is that instead of replacing words by generalized tokens, we add a node in the syntax tree with a generalized token so that it dominates the pertaining leaf node (see also nodes marked with $_{AUG}$ in Figure 2). All sources used for this type of generalization are known to be predictive for opinion holder classification (Choi et al., 2005; Kim and Hovy, 2005; Choi et al., 2006; Kim and Hovy, 2006; Bloom et al., 2007).

Note that the grammatical relation paths, i.e. $GRAM_{WRD}$ and $GRAM_{POS}$, can only be applied in case there is another expression in the focus in addition to the candidate of the data instance itself, e.g. the nearest opinion expression to the candidate. Section 4.4 explains in detail how this is done.

Predicate-argument structures (PAS) are represented by PropBank trees (Kingsbury and Palmer, 2002).

4.2 Support Vector Machines and Kernel Methods

Support Vector Machines (SVMs) are one of the most robust supervised machine learning techniques in which training data instances $\vec{x}$ are separated by a hyperplane $H(\vec{x}) = \vec{w} \cdot \vec{x} + b = 0$ where $w \in \mathbb{R}^n$ and $b \in \mathbb{R}$. One advantage of SVMs is that kernel methods can be applied which map the data to other feature spaces in which they can be separated more easily. Given a feature function $\phi : \mathbb{O} \rightarrow \mathbb{R}$, where $\mathbb{O}$ is the set of the objects, the kernel trick allows the decision hyperplane to be rewritten as:

$$H(\vec{x}) = \left(\sum_{i=1 \dots l} y_i \alpha_i \vec{x}_i \right) \cdot \vec{x} + b = \sum_{i=1 \dots l} y_i \alpha_i \vec{x}_i \cdot \vec{x} + b = \sum_{i=1 \dots l} y_i \alpha_i \phi(o_i) \cdot \phi(o) + b$$

where y_i is equal to 1 for positive and -1 for negative examples, $\alpha_i \in \mathbb{R}$ with $\alpha_i \geq 0, o_i \forall i \in \{1, \dots, l\}$ are the training instances and the product $K(o_i, o) = \langle \phi(o_i) \cdot \phi(o) \rangle$ is the kernel function associated with the mapping ϕ.

case there are several tokens making up the candidate.

4.3 Sequence and Tree Kernels

A sequence kernel (SK) measures the similarity of two sequences by counting the number of common subsequences. We use the kernel by Taylor and Christianini (2004) which has the advantage that it also considers subsequences of the original sequence with some elements missing. The extent of these *gaps* in a sequence is suitably reflected by a weighting function incorporated into the kernel.

Tree kernels (TKs) represent trees by their substructures. The feature space of these substructures, or fragments, is mapped onto a vector space. The kernel function computes the similarity of pairs of trees by counting the number of common fragments. In this work, we evaluate two tree kernels: Subset Tree Kernel (STK) (Collins and Duffy, 2002) and Partial Tree Kernel (PTK_{basic}) (Moschitti, 2006).

In STK, a tree fragment can be any set of nodes and edges of the original tree provided that every node has either all or none of its children. This constraint makes that kind of kernel well-suited for constituency trees which have been generated by context free grammars since the constraint corresponds to the restriction that no grammatical rule must be broken. For example, STK enforces that a subtree, such as *[VP [VBZ, NP]]*, cannot be matched with *[VP [VBZ]]* since the latter *VP* node only possesses one of the children of the former.

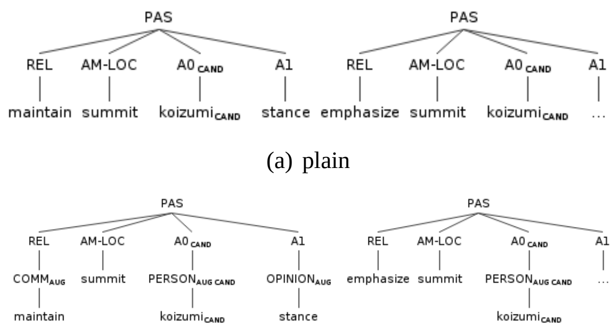
PTK_{basic} is more flexible since the constraint of STK on nodes is relaxed. This makes this type of tree kernel less suitable for constituency trees. We, therefore, apply it only to trees representing predicate-argument structures (PAS) (see Figure 1). Note that a data instance is represented by a set of those structures[3] rather than a single structure. Thus, the actual partial tree kernel function we use for this task, PTK, sums over all possible pairs PAS_l and PAS_m of two data instances x_i and x_j: $PTK(x_i, x_j) = \sum_{PAS_l \in x_i} \sum_{PAS_m \in x_j} PTK_{basic}(PAS_l, PAS_m)$.

To summarize, Table 2 lists the different kernel types we use coupled with the suitable levels of representation. This choice of pairing has already been motivated and empirically proven suitable on other

[3] i.e. all predicate-argument structures of a sentence in which the head of the candidate opinion holder occurs

Type	Description	Example
WRD	sequence of words	During the summit , Koizumi$_{CAND}$ maintained a clear-cut collaborative stance . . .
WRD_{GN}	sequence of generalized words	During the summit , CAND COMM OPINION . . .
POS	part-of-speech sequence	IN DET NN PUNC CAND VBD DET JJ JJ NN . . .
POS_{GN}	generalized part-of-speech sequence	IN DET NN PUNC CAND COMM OPINION . . .
$CONST$	constituency tree	*see Figure 2 without nodes marked* $_{AUG}$
$CONST_{AUG}$	augmented constituency tree	*see Figure 2*
$GRAM_{WRD}$	grammatical relation path labels with words	Koizumi$_{CAND}$ NSUBJ↑ maintained DOBJ↓ stance
$GRAM_{POS}$	grammatical relation path labels with part-of-speech tags	CAND NSUBJ↑ VBD DOBJ↓ NN
PAS	predicate argument structures	*see Figure 1(a)*
PAS_{AUG}	augmented predicate argument structures	*see Figure 1(b)*

Table 1: The different levels of representation.



(a) plain

(b) augmented

Figure 1: Predicate-argument structures (PAS).

tasks (Moschitti, 2008; Nguyen et al., 2009).

Type	Description	Levels of Representation
SK	Sequential Kernel	$WRD_{(GN)}$, $POS_{(GN)}$, $GRAM_{WRD}$, $GRAM_{POS}$
STK	Subset Tree Kernel	$CONST_{(AUG)}$
PTK	Partial Tree Kernel	PAS
VK	Vector Kernel	*not restricted*

Table 2: The different types of kernels.

4.4 The Different Scopes

We argue that using the entire word sequence or syntax tree of the sentence in which a candidate opinion holder is situated to represent a data instance produces too large structures for a convolution kernel. Since a classifier based on convolution kernels has to derive meaningful features by itself, the larger these structures are, the more likely noise is included in the model. Previous work in relation extraction has also shown that the usage of more focused substructures, e.g. the smallest subtree containing the two candidate entities of a relation, is more effective (Zhang et al., 2006). Unfortunately, in our task there is only one explicit entity we know of for each data instance which is the candidate opinion holder. However, there are several indicative cues within the context of the candidate which might be considered important. We identify three different cues being the nearest *predicate*, i.e. full verb or nominalization, *opinion word* and *communication word*[4]. For each of these expressions, we define a scope where the boundaries are the candidate opinion holder and the pertaining cue. Given these scopes, we can define resulting subsequences/subtrees and combine them. We further add two *background scopes*, one being the semantic scope of the candidate opinion holder and the entire sentence. As semantic scope we consider the subclause in which a candidate opinion holder is situated[5].

Figure 2 illustrates the different scopes. Abbreviations are explained in Table 3. As already mentioned in Section 4.1 for grammatical relation paths, a second expression in addition to the candidate opinion holder is required. These expressions can be derived from the different scopes, i.e. for $PRED$ it

[4]These three expressions may coincide but do not have to.

[5]Typically, the subtree representing a subclause has the closest S node dominating the candidate opinion holder as the root node and it contains only those nodes from the original sentence parse which are also dominated by that S node and whose path to that node does not contain another S node.

is the nearest predicate to the candidate, for OP it is the nearest opinion word and for $COMM$ it is the nearest communication word. For the background scopes SEM and $SENT$, however, there is no second expression in focus. Therefore, grammatical relation paths cannot be defined for these scopes.

Type	**Description**
$PRED$	scope with the boundaries being the candidate opinion holder and the nearest predicate
OP	scope with the boundaries being the candidate opinion holder and nearest opinion word
$COMM$	scope with the boundaries being the candidate opinion holder and the nearest communication word
SEM	semantic scope of the candidate opinion holder, i.e. subclause containing the candidate
$SENT$	entire sentence in which in the opinion holder occurs

Table 3: The different types of scope.

headword/governing category of CAND
is CAND capitalized/a person?
is CAND *subj*|*dobj*|*iobj*|*pobj* of OPINION/COMM?
is CAND preceded by *according to*? (Choi et al., 2005)
does CAND contain possessive and is followed by OPINION/COMM? (Choi et al., 2005)
is CAND preceded by *by* which is attached to OPINION/COMM? (Choi et al., 2005)
predicate-argument pairs in which CAND occurs
lemma/part-of-speech tag/subcategorization frame/voice of nearest predicate
is nearest predicate OPINION/COMM?
does CAND precede/follow nearest predicate?
words between nearest predicate and CAND (bag of words)
part-of-speech sequence between nearest predicate and CAND
constituency path/grammatical relation path from predicate to CAND

Table 4: Manually designed feature set.

4.5 Manually Designed Feature Set for a Standard Vector Kernel

In addition to the different types of convolution kernels, we also define an explicit feature set for a vector kernel (VK). Many of these features mainly describe properties of the relation between the candidate and the nearest predicate[6] since in our initial experiments the nearest predicate has always been the strongest cue. Adding these types of features for other cues, e.g. the nearest opinion or communication word, only resulted in a decrease in performance. Table 4 lists all the features we use. Note that this manual feature set employs all those sources of information which are also exploited by the convolution kernels. Some of the information contained in the convolution kernels can, however, only be represented in a more simplified fashion when using a manual feature set. For example, the first PAS in Figure 1(a) is converted to just the pair of predicate and argument representing the candidate (i.e. *REL:maintain_A0:Koizumi*). The entire PAS is not used since it would create too sparse features. Convolution kernels can cope with fairly complex structures as input since they internally match substructures. Manual features are less flexible since they do not account for partial matches.

5 Experiments

We used 400 documents of the MPQA corpus for five-fold crossvalidation and 133 documents as a development set. We report statistical significance on the basis of a paired t-test using 0.05 as the significance level. All experiments were done with the *SVM-Light-TK* toolkit[7]. We evaluated on the basis of exact phrase matching. We set the trade-off parameter $j = 5$ for all feature sets. For the manual feature set we used a polynomial kernel of third degree. These two critical parameters were tuned on the development set. As far as the sequence and tree kernels are concerned, we used the parameter settings from Moschitti (2008), i.e. $\lambda = 0.4$ and $\mu = 0.4$. Kernels were combined using plain summation. The documents were parsed using the Stanford Parser (Klein and Manning, 2003). Named-entity information was obtained by the Stanford tagger (Finkel et al., 2005). Semantic roles were obtained by using the parser by Zhang et al. (2008). Opinion expressions were identified using the Subjectivity Lexicon from the MPQA project (Wilson et al., 2005). Communication words were obtained by using the Appraisal Lexicon (Bloom et al., 2007). Nominalizations were recognized by looking

[6] We select the nearest predicate by using the syntactic parse tree. Thus, we hope to select the predicate which syntactically relates to the candidate opinion holder.

[7] available at `disi.unitn.it/moschitti`

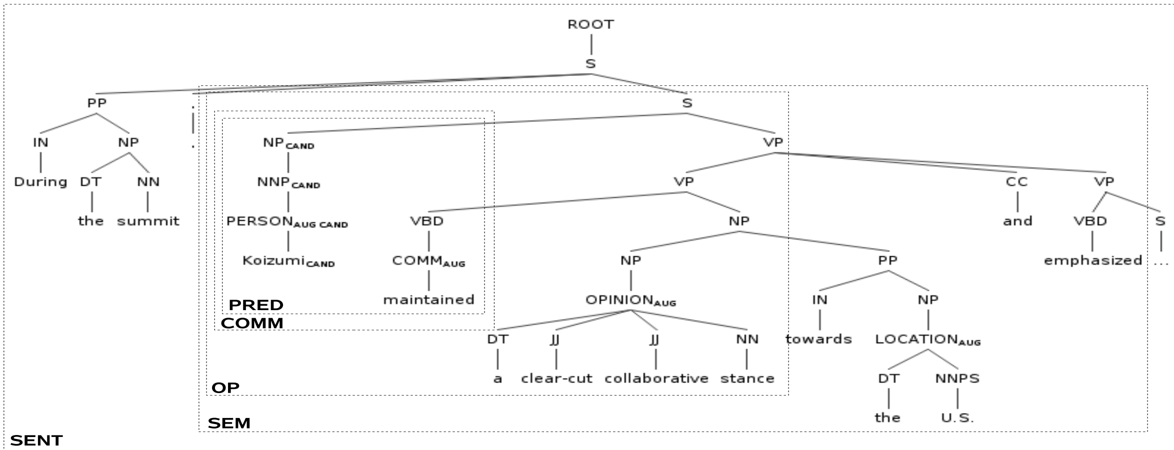


Figure 2: Illustration of the different scopes on a $CONST_{AUG}$; nodes belonging to the candidate opinion holder are marked with $_{CAND}$.

up nouns in NOMLEX (Macleod et al., 1998).

5.1 Notation

Each kernel is represented as a triple ⟨*levelOfRepresentation* (Table 1), *Scope* (Table 3), *typeOfKernel* (Table 2)⟩, e.g. $\langle CONST, SENT, STK\rangle$ is a Subset Tree Kernel of a constituency parse having the scope of the entire sentence. Note that not all combinations of these three parameters are meaningful. In the following, we will just focus on important and effective combinations. The kernel composed of manually designed features is denoted by just VK. The kernel composed of predicate-argument structures is denoted by $\langle PAS, SENT, PTK\rangle$.

5.2 Vector Kernel (VK)

The first line in Table 7 displays the result of the vector kernel using a manually designed feature set. It should be interpreted as a baseline. Due to the high class imbalance we will focus on the comparison of F(1)-Score throughout this paper rather than accuracy which is fairly biased on this data set. The F-Score of this classifier is at 56.16%.

5.3 Sequence Kernels (SKs)

For both sequence and tree kernels we need to find out what the best scope is, whether it is worthwhile to combine different scopes and what different layers of representation can be usefully combined.

The upper part of Table 5 lists the results of simple word kernels using the different scopes. The performance of the kernels using individual scopes varies greatly. The best scope is $PRED$ (1), the second best is SEM (2). The good performance of $PRED$ does not come as a surprise since the sequence is the smallest among the different scopes, so this scope is least affected by data sparseness. Moreover, this result is consistent with our initial experiments on the manual feature set (see Section 4.5).

Using different combinations of the word sequence kernels shows that $PRED$ and SEM (6) are a good combination, whereas OP, $COMM$, and $SENT$ (7;8;9) do not positively contribute to the overall performance which is consistent which the individual scope evaluation. Apparently, these scopes capture less linguistically relevant structure.

The next part of Table 5 shows the contribution of POS kernels when added to WRD kernels. Adding the corresponding POS kernel to the WRD kernel with $PRED$ scope (10) results in an improvement by more than 5% in F-Score. We get another improvement by approx. 3% when the corresponding SEM kernels (11) are added. This suggests that POS is an effective generalization and that the two scopes $PRED$ and SEM are complementary.

For the $GRAM_{WRD}$ kernel, the $PRED$ scope (12) is again most effective. We assume that this kernel most likely expresses meaningful syntactic relationships for our task. Adding the $GRAM_{POS}$ kernel (14) gives another boost by almost 4%.

Generalized sequence kernels are important.

Adding the corresponding WRD_{GN} kernels to the WRD kernel with $PRED$ and SEM scope results in an improvement from 47.77% (1) to 53.00% (15) which is a bit less than the combination of WRD and $POS_{(GN)}$ kernels (16). However, these types of kernels seem to be complementary since their combination provides an F-Score of 56.06% (17). This kernel combination already performs on a par with the manually designed vector kernel though less information is taken into consideration.

Finally, the best combination of sequence kernels (18) comprises WRD, WRD_{GN}, POS, and POS_{GN} kernels with $PRED$ and SEM scope combined with a $GRAM_{WRD}$ and a $GRAM_{POS}$ kernel with $PRED$ scope. The performance of 58.70% significantly outperforms the vector kernel.

5.4 Tree Kernels (TKs)

Table 6 shows the results of the different tree kernels. The table is divided into two halves. The left half (A) are plain tree kernels, whereas the right half (B) are the augmented tree kernels. As far as $CONST$ kernels are concerned, there is a systematic improvement by approximately 2% using tree augmentation. This proves that further non-syntactic knowledge added to the tree itself results in an improved F-Score. However, tree augmentation does not have any impact on the PAS kernels.

The overall performance of the tree kernels shows that they are much more expressive than sequence kernels. For instance, in order to obtain the same performance as of $\langle CONST_{AUG}, PRED, STK \rangle$ (19B), i.e. a single kernel with an F-Score 56.52, it requires several sequence kernels, hence much more effort. The performance of the different $CONST$ kernels relative to each other resembles the results of the WRD kernels. The best scope is $PRED$ (19). By far the worst performance is obtained by the $SENT$ scope (23). The combination of $PRED$ and SEM scope achieves an F-Score of 59.67% (25B) which is already slightly better than the best configuration of sequence kernels (18).

The performance of the PAS kernel (28A) with an F-Score of 53.51% is slightly worse than the best single plain $CONST$ kernel (19A). The PAS kernel and the $CONST$ kernels are complementary, since their best combination (29B) achieves an F-Score of 61.67% which is significantly better than the best combination of $CONST$ kernels (25B) or sequence kernels (18).

Combination	**Acc.**	**Prec.**	**Rec.**	**F1**
VK	93.63	53.28	59.37	56.16
best SKs	94.21	57.64	59.81	58.70
best TKs	94.16	56.18	**68.36**	61.67*
VK + best SKs	94.34	58.44	61.27	59.82*
VK + best TKs	94.33	57.41	68.03	62.27*
best SKs + best TKs	94.49	**59.22**	63.96	61.49*
VK + best SKs + best TKs	**94.53**	59.10	66.57	**62.61***†

Table 7: Results of kernel combinations (*: significantly better than best SKs; †: significantly better than best TKs; all convolution kernels are significantly better than VK).

5.5 Combinations

Table 7 lists the results of the different kernel type combinations. If VK is added to the best TKs, the best SKs, or both, a slight increase in F-Score is achieved. The best performance with an F-Score of 62.61% is obtained by combining all kernels.

6 Conclusion

In this paper, we compared convolution kernels for opinion holder extraction. We showed that, in general, a combination of two scopes, namely the scope immediately encompassing the candidate opinion holder and its nearest predicate and the subclause containing the candidate opinion holder provide best performance. Tree kernels containing constituency parse information and semantic roles achieve better performance than sequence kernels or vector kernels using a manually designed feature set. Best performance is achieved if all kernels are combined.

Acknowledgements

Michael Wiegand was funded by the German research council DFG through the International Research Training Group "IRTG" between Saarland University and University of Edinburgh.

The authors would like to thank Yi Zhang for processing the MPQA corpus with his semantic-role labeling system, the researchers from the MPQA project for helping to create an opinion holder corpus, and, in particular, Alessandro Moschitti for insightful comments and suggestions.

ID	Kernel	Acc.	Prec.	Rec.	F1
1	$\langle WRD, PRED, SK \rangle$	93.25	51.08	42.29	46.26
2	$\langle WRD, OP, SK \rangle$	92.77	46.38	32.52	38.21
3	$\langle WRD, COMM, SK \rangle$	92.42	43.70	35.99	39.46
4	$\langle WRD, SEM, SK \rangle$	93.16	50.32	34.65	41.04
5	$\langle WRD, SENT, SK \rangle$	90.60	29.90	27.29	28.53
6	$\langle WRD, PRED, SK \rangle + \langle WRD, SEM, SK \rangle$	93.78	56.55	41.36	47.77
7	$\sum_{j \in \{PRED,OP,COMM\}} \langle WRD, j, SK \rangle$	93.55	54.26	39.50	45.71
8	$\sum_{j \in Scopes \setminus SENT} \langle WRD, j, SK \rangle$	93.82	57.21	40.28	47.26
9	$\sum_{j \in Scopes} \langle WRD, j, SK \rangle$	93.63	55.15	39.52	46.03
10	$\langle WRD, PRED, SK \rangle + \langle POS, PRED, SK \rangle$	93.03	49.39	53.53	51.37
11	$\sum_{i \in \{PRED,SEM\}} (\langle WRD, i, SK \rangle + \langle POS, i, SK \rangle)$	93.86	55.60	53.22	54.38
12	$\sum_{i \in \{PRED,SEM\}} \langle WRD, i, SK \rangle + \langle GRAM_{WRD}, PRED, SK \rangle$	94.01	58.19	45.88	51.29
13	$\sum_{i \in \{PRED,SEM\}} \langle WRD, i, SK \rangle + \sum_{j \in \{PRED,OP,COMM\}} \langle GRAM_{WRD}, j, SK \rangle$	93.83	56.28	45.64	50.40
14	$\sum_{i \in \{PRED,SEM\}} \langle WRD, i, SK \rangle + \langle GRAM_{WRD}, PRED, SK \rangle + \langle GRAM_{POS}, PRED, SK \rangle$	93.98	56.59	53.92	55.21
15	$\sum_{i \in \{PRED,SEM\}} (\langle WRD, i, SK \rangle + \langle WRD_{GN}, i, SK \rangle)$	93.97	57.08	49.46	53.00
16	$\sum_{i \in \{PRED,SEM\}} (\langle WRD, i, SK \rangle + \langle POS_{GN}, i, SK \rangle)$	93.97	56.60	52.42	54.42
17	$\sum_{i \in \{PRED,SEM\}} (\langle WRD, i, SK \rangle + \langle WRD_{GN}, i, SK \rangle + \langle POS, i, SK \rangle + \langle POS_{GN}, i, SK \rangle)$	93.85	55.16	57.00	56.06
18	$\sum_{i \in \{PRED,SEM\}} (\langle WRD, i, SK \rangle + \langle WRD_{GN}, i, SK \rangle + \langle POS, i, SK \rangle + \langle POS_{GN}, i, SK \rangle)$ $+ \langle GRAM_{WRD}, PRED, SK \rangle + \langle GRAM_{POS}, PRED, SK \rangle$	**94.21**	**57.64**	**59.81**	**58.70**

Table 5: Results of the different sequence kernels.

		A				B			
		$i = CONST, j = PAS$				$i = CONST_{AUG}, j = PAS_{AUG}$			
ID	**Kernel**	**Acc.**	**Prec.**	**Rec.**	**F1**	**Acc.**	**Prec.**	**Rec.**	**F1**
19	$\langle i, PRED, STK \rangle$	92.89	48.68	62.34	54.67	93.12	49.99	65.04	56.52
20	$\langle i, OP, STK \rangle$	93.04	49.49	54.71	51.96	93.27	50.93	59.06	54.68
21	$\langle i, COMM, STK \rangle$	92.76	47.79	55.89	51.50	92.96	49.03	58.85	53.47
22	$\langle i, SEM, STK \rangle$	93.70	54.40	52.13	53.23	93.90	55.47	56.59	56.03
23	$\langle i, SENT, STK \rangle$	92.42	44.34	39.92	41.99	92.50	45.20	42.40	43.74
24	$\sum_{k \in \{PRED,OP,COMM\}} \langle i, k, STK \rangle$	93.62	53.26	60.05	56.44	93.77	54.06	63.21	58.26
25	$\sum_{k \in \{PRED,SEM\}} \langle i, k, STK \rangle$	93.90	55.26	59.50	57.30	94.13	56.57	63.12	59.67
26	$\sum_{k \in Scopes \setminus SENT} \langle i, k, STK \rangle$	94.09	56.65	59.68	58.11	94.21	57.21	62.61	59.80
27	$\sum_{k \in Scopes} \langle i, k, STK \rangle$	94.14	57.41	57.88	57.63	94.29	**58.11**	61.10	59.56
28	$\langle j, SENT, PTK \rangle$	92.11	45.02	**69.96**	53.51	91.92	44.27	67.39	53.43
29	$\sum_{k \in \{PRED,SEM\}} \langle i, k, STK \rangle + \langle PAS, SENT, PTK \rangle$	94.05	55.68	66.01	**60.40**	94.16	56.18	**68.36**	**61.67**
30	$\sum_{k \in Scopes \setminus SENT} \langle i, k, STK \rangle + \langle PAS, SENT, PTK \rangle$	**94.30**	**57.95**	62.62	60.19	**94.36**	58.07	64.94	61.31

Table 6: Results of the different tree kernels.

References

Steven Bethard, Hong Yu, Ashley Thornton, Vasileios Hatzivassiloglou, and Dan Jurafsky. 2005. Extracting Opinion Propositions and Opinion Holders using Syntactic and Lexical Cues. In *Computing Attitude and Affect in Text: Theory and Applications*. Springer.

Kenneth Bloom, Sterling Stein, and Shlomo Argamon. 2007. Appraisal Extraction for News Opinion Analysis at NTCIR-6. In *Proceedings of NTCIR-6 Workshop Meeting*, Tokyo, Japan.

Razvan C. Bunescu and Raymond J. Mooney. 2005. Subsequence Kernels for Relation Extraction. In *Proceedings of the Conference on Neural Information Processing Systems (NIPS)*, Vancouver, Canada.

Yejin Choi, Claire Cardie, Ellen Riloff, and Siddharth Patwardhan. 2005. Identifying Sources of Opinions with Conditional Random Fields and Extraction Patterns. In *Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing (HLT/EMNLP)*, Vancouver, Canada.

Yejin Choi, Eric Breck, and Claire Cardie. 2006. Joint Extraction of Entities and Relations for Opinion Recognition. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Sydney, Australia.

Michael Collins and Nigel Duffy. 2002. New Ranking Algorithms for Parsing and Tagging. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, Philadelphia, USA.

Charles. J. Fillmore, Christopher R. Johnson, and Miriam R. Petruck. 2003. Background to FrameNet. *International Journal of Lexicography*, 16:235 – 250.

Jenny Rose Finkel, Trond Grenager, and Christopher Manning. 2005. Incorporating Non-local Information into Information Extraction Systems by Gibbs Sampling. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, Ann Arbor, USA.

Soo-Min Kim and Eduard Hovy. 2005. Identifying Opinion Holders for Question Answering in Opinion Texts. In *Proceedings of AAAI-05 Workshop on Question Answering in Restricted Domains*, Pittsburgh, USA.

Soo-Min Kim and Eduard Hovy. 2006. Extracting Opinions, Opinion Holders, and Topics Expressed in Online News Media Text. In *Proceedings of the ACL Workshop on Sentiment and Subjectivity in Text*, Sydney, Australia.

Paul Kingsbury and Martha Palmer. 2002. From TreeBank to PropBank. In *Proceedings of the 3rd Conference on Language Resources and Evaluation (LREC)*, Las Palmas, Spain.

Dan Klein and Christopher D. Manning. 2003. Accurate Unlexicalized Parsing. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, Sapporo, Japan.

Catherine Macleod, Ralph Grishman, Adam Meyers, Leslie Barrett, and Ruth Reeves. 1998. NOMLEX: A Lexicon of Nominalizations. In *Proceedings of EURALEX*, Liège, Belgium.

Alessandro Moschitti, Daniele Pighin, and Roberto Basili. 2008. Tree Kernels for Semantic Role Labeling. *Computational Linguistics*, 34(2):193 – 224.

Alessandro Moschitti. 2006. Efficient Convolution Kernels for Dependency and Constituent Syntactic Trees. In *Proceedings of the 17th European Conference on Machine Learning (ECML)*, Berlin, Germany.

Alessandro Moschitti. 2008. Kernel Methods, Syntax and Semantics for Relational Text Categorization. In *Proceedings of the Conference on Information and Knowledge Management (CIKM)*, Napa Valley, USA.

Truc-Vien T. Nguyen, Alessandro Moschitti, and Giuseppe Riccardi. 2009. Convolution Kernels on Constituent, Dependency and Sequential Structures for Relation Extraction. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Singapore.

Ellen Riloff. 1996. An Empirical Study of Automated Dictionary Construction for Information Extraction. *Artificial Intelligence*, 85.

John Taylor and Nello Christianini. 2004. *Kernel Methods for Pattern Analysis*. Cambridge University Press.

Janyce Wiebe, Theresa Wilson, and Claire Cardie. 2003. Annotating Expressions of Opinions and Emotions in Language. *Language Resources and Evaluation*, 1:2.

Theresa Wilson, Janyce Wiebe, and Paul Hoffmann. 2005. Recognizing Contextual Polarity in Phrase-level Sentiment Analysis. In *Proceedings of Human Language Technologies Conference/Conference on Empirical Methods in Natural Language Processing (HLT/EMNLP)*, Vancouver, Canada.

Dell Zhang and Wee Sun Lee. 2003. Question Classification using Support Vector Machines. In *Proceedings of the ACM Special Interest Group on Information Retrieval (SIGIR)*, Toronto, Canada.

Min Zhang, Jie Zhang, and Jian Su. 2006. Exploring Syntactic Features for Relation Extraction using a Convolution Tree Kernel. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the ACL (HLT/NAACL)*, New York City, USA.

Yi Zhang, Rui Wang, and Hans Uszkoreit. 2008. Hybrid Learning of Dependency Structures from Heterogeneous Linguistic Resources. In *Proceedings of the Conference on Computational Natural Language Learning (CoNLL)*, Manchester, United Kingdom.