Syntactic/Semantic Structures for Textual Entailment Recognition

Yashar Mehdad
FBK-IRST, DISI
University of Trento
Povo (TN) - Italy
mehdad@fbk.eu

Alessandro Moschitti
DISI
University of Trento
Povo (TN) - Italy
moschitti@disi.unitn.it

Fabio Massimo Zanzotto
DISP
University of Rome "Tor Vergata"
Roma - Italy
zanzotto@info.uniroma2.it

Abstract

In this paper, we describe an approach based on off-the-shelf parsers and semantic resources for the Recognizing Textual Entailment (RTE) challenge that can be generally applied to any domain. Syntax is exploited by means of tree kernels whereas lexical semantics is derived from heterogeneous resources, e.g. WordNet or distributional semantics through Wikipedia. The joint syntactic/semantic model is realized by means of tree kernels, which can exploit lexical relatedness to match syntactically similar structures, i.e. whose lexical compounds are related. The comparative experiments across different RTE challenges and traditional systems show that our approach consistently and meaningfully achieves high accuracy, without requiring any adaptation or tuning.

1 Introduction

Recognizing Textual Entailment (RTE) is rather challenging as effectively modeling syntactic and semantic for this task is difficult. Early deep semantic models (e.g., (Norvig, 1987)) as well as more recent ones (e.g., (Tatu and Moldovan, 2005; Bos and Markert, 2005; Roth and Sammons, 2007)) rely on specific world knowledge encoded in rules for drawing decisions. Shallower models exploit matching methods between syntactic/semantic graphs of texts and hypotheses (Haghighi et al., 2005). The matching step is carried out after the application of some lexical-syntactic rules that are used to transform the text T or the hypothesis H (Bar-Haim et al., 2009) at surface form level. For all these methods, the effective use of syntactic and semantic information depends on the coverage and the quality of the specific rules. Lexical-syntactic rules can be automatically extracted from plain corpora (e.g., (Lin and Pantel, 2001; Szpektor and Dagan, 2008)) but the quality (also in terms of little noise) and the coverage is low. In contrast, rules written at the semantic level are more accurate but their automatic design is difficult and so they are typically hand-coded for the specific phenomena.

In this paper, we propose models for effectively using syntactic and semantic information in RTE, without requiring either large automatic rule acquisition or hand-coding. These models exploit lexical similarities to generalize lexical-syntactic rules automatically derived by supervised learning methods. In more detail, syntax is encoded in the form of parse trees whereas similarities are defined by means of WordNet simlilarity measures or Latent Semantic Analysis (LSA) applied to Wikipedia or to the British National Corpus (BNC). The joint syntactic/semantic model is realized by means of novel tree kernels, which can match subtrees whose leaves are lexically similar (so not just identical).

To assess the benefit of our approach, we carried out comparative experiments with previous work: especially with the method described in (Zanzotto and Moschitti, 2006; Zanzotto et al., 2009). This constitutes our strong baseline as, although it can only exploit lexical-syntactic rules, it has achieved top accuracy in all RTE challenges. The results, across different RTE challenges, show that our approach constantly and significantly improves the

Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the ACL, pages 1020–1028,
Los Angeles, California, June 2010.

baseline model. Moreover, our approach does not require any adaptation or tuning and uses a computation for the similarity function based on Wikipedia which is faster than the computation of tools based on WordNet or other resources (Basili et al., 2006).

The remainder of the paper is organized as follows: Section 2 critically reviews the previous work by highlighting the need of generalizing lexico-syntactic rules. Section 3 describes lexical similarity approaches, which can serve the generalization purpose. Section 4 describes how to integrate lexical similarity in syntactic structures using syntactic/semantic tree kernels (SSTK) whereas Section 5 shows how to use SSTK in a kernel-based RTE system. Section 6 describes the experiments and results. Section 7 discusses the efficiency and accuracy of our system compared with other RTE systems. Finally, we draw the conclusions in Section 8.

2 Related work

Lexical-syntactic rules are largely used in textual entailment recognition systems (e.g., (Bar-Haim et al., 2007; Dinu and Wang, 2009)) as they conveniently encode world knowledge into linguistic structures. For example, to decide whether the simple sentences are in the entailment relation:

$T_2 \Rightarrow ? H_2$	
T_2	*"In 1980 Chapman killed Lennon."*
H_2	*"John Lennon died in 1980."*

we need a lexical-syntactic rule such as:

$$\rho_3 = \boxed{X} killed \boxed{Y} \rightarrow \boxed{Y} died$$

along with such rules, the temporal information should be taken into consideration.

Given the importance of lexical-syntactic rules in RTE, many methods have been proposed for their extraction from large corpora (e.g., (Lin and Pantel, 2001; Szpektor and Dagan, 2008)). Unfortunately, these unsupervised methods in general produce rules that can hardly be used: noise and coverage are the most critical issues.

Supervised approaches were experimented in (Zanzotto and Moschitti, 2006; Zanzotto et al., 2009), where lexical-syntactic rules were derived from examples in terms of complex relational features. This approach can easily miss some useful information and rules. For example, given the pair $\langle T_2, H_2 \rangle$, to derive the entailment value of the following case:

$T_4 \Rightarrow ? H_4$	
T_4	*"In 1963 Lee Harvey Oswald murdered JFK"*
H_4	*"JFK died in 1963"*

we can only rely on this relatively interesting lexical-syntactic rule (i.e. which is in common between the two examples):

$$\rho_5 = (VP(VBZ)(NP\boxed{X})) \rightarrow (S(NP\boxed{X})(VP(VBZ\ \ died)))$$

Unfortunately, this can be extremely misleading since it also derives similar decisions for the following example:

$T_6 \Rightarrow ? H_6$	
T_6	*"In 1956 JFK met Marilyn Monroe"*
H_6	*"Marilyn Monroe died in 1956"*

The problem is that the pairs $\langle T_2, H_2 \rangle$ and $\langle T_4, H_4 \rangle$ share more meaningful features than the rule ρ_5, which should make the difference with respect to the relation between the pairs $\langle T_2, H_2 \rangle$ and $\langle T_6, H_6 \rangle$. Indeed, the word "*kill*" is more semantically related to "*murdered*" than to "*meet*". Using this information, it is possible to derive more effective rules from training examples.

There are several solutions for taking this information into account, e.g. by using FrameNet semantics (e.g., like in (Burchardt et al., 2007)), it is possible to encode a lexical-syntactic rule using the KILLING and the DEATH frames, i.e.:

$$\rho_7 = \begin{array}{l} KILLING(Killer:\boxed{X}, \\ Victim:\boxed{Y}) \end{array} \rightarrow \begin{array}{l} DEATH(\\ Protagonist:\boxed{Y}) \end{array}$$

However, to use this model, specific rules and a semantic role labeler on the specific corpora are needed.

3 Lexical similarities

Previous research in computational linguistics has produced many effective lexical similarity measures based on many different resources or corpora. For example, WordNet similarities (Pedersen et al., 2004) or Latent Semantic Analysis over a large corpus are widely used in many applications and for

the definition of kernel functions, e.g. (Basili et al., 2006; Basili et al., 2005; Bloehdorn et al., 2006).

In this section we present the main component of our new kernel, i.e. a lexical similarity derived from different resources. This is used inside the syntactic/semantic tree kernel defined in (Bloehdorn and Moschitti, 2007a; Bloehdorn and Moschitti, 2007b) to enhance the basic tree kernel functions.

3.1 WordNet Similarities

WordNet similarities have been heavily used in previous NLP work (Chan and Ng, 2005; Agirre et al., 2009). All WordNet similarities apply to pairs of synonymy sets (synsets) and return a value indicating their semantic relatedness. For example, the following measures, that we use in this study, are based on path lengths between concepts in the Wordnet Hierarchy:

Path the measure is equal to the inverse of the shortest path length ($path\_length$) between two synsets c_1 and c_2 in WordNet

$$Sim_{Path} = \frac{1}{path\_length(c_1, c_2)} \quad (1)$$

WUP the Wu and Palmer (Wu and Palmer, 1994) similarity metric is based on the depth of two given synsets c_1 and c_2 in the WordNet taxonomy, and the depth of their least common subsumer (lcs). These are combined into a similarity score:

$$Sim_{WUP} = \frac{2 \times depth(lcs)}{depth(c_1) + depth(c_2)} \quad (2)$$

Wordnet similarity measures on synsets can be extended to similarity measures between words as follows:

$$\kappa_{\mathcal{S}}(w_1, w_2) = max_{(c_1,c_2)\in C_1\times C_2} Sim_{\mathcal{S}}(c_1, c_2) \quad (3)$$

where $\mathcal{S}$ is Path or WUP and C_i is the set of the synsets related to the word w_i.

3.2 Distributional Semantic Similarity

Latent Semantic Analysis (LSA) is one of the corpus-based measure of distributional semantic similarity, proposed by (Landauer et al., 1998). Words $\vec{w_i}$ are represented in a document space. Each feauture is a document and its value is the frequency of the word in the document. The similarity is generally computed as a cosine similarity:

$$\kappa_{LSI}(w_1, w_2) = \frac{\vec{w_1}\vec{w_2}}{||\vec{w_1}|| \times ||\vec{w_2}||} \quad (4)$$

In our approach we define a proximity matrix P where $p_{i,j}$ represents $\kappa_{LSI}(w_i, w_j)$ The core of our approach lies on LSI (Latent Semantic Indexing) over a large corpus. We used singular value decomposition (SVD) to build the proximity matrix $P = DD^T$ from a large corpus, represented by its word-by-document matrix D.

SVD decomposes D (weighted matrix of term frequencies in a collection of text) into three matrices $U\Sigma V^T$, where U (matrix of term vectors) and V (matrix of document vectors) are orthogonal matrices whose columns are the eigenvectors of DD^T and D^TD respectively, and Σ is the diagonal matrix containing the singular value of D.

Given such decomposition, P can be obtained as $U_k\Sigma_k^2U_k^T$, where U_k is the matrix containing the first k columns of U and k is the dimensionality of the latent semantic space. This is efficiently used to reduce the memory requirements while retaining the information. Finally we computed the term similarity using the cosine measure in the vector space model (VSM).

Generally, LSA can be observed as a way to overcome some of the drawbacks of the standard vector space model, such as sparseness and dimensionality. In other words, the LSA similarity is computed in a lower dimensional space, in which second-order relations among words and documents are exploited (Mihalcea et al., 2006).

It is worth mentioning that the LSA similarity measure depends on the selected corpus but it benefits from a higher computation speed in comparison to the construction of the similarity matrix based on the WordNet Similarity package (Pedersen et al., 2004).

4 Lexical similarity in Syntactic Tree Kernels

Section 2 has shown that the role of the syntax is important in extracting generalized rules for RTE but it is not enough. Therefore, the lexical similarity described in the previous section should be taken into

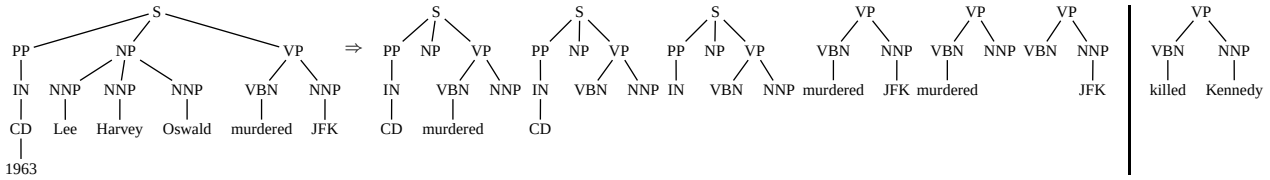

Figure 1: A syntactic parse tree (on the left) along with some of its fragments. After the bar there is an important fragment from a semantically similar sentence, which cannot be matched by STK but it is matched by SSTK.

account in the model definition. Since tree kernels have been shown to be very effective for exploiting syntactic information in natural language tasks, a promising idea is to merge together the two different approaches, i.e. tree kernels and semantic similarities.

4.1 Syntactic Tree Kernel (STK)

Tree kernels compute the number of common substructures between two trees T_1 and T_2 without explicitly considering the whole fragment space. The standard definition of the STK, given in (Collins and Duffy, 2002), allows for any set of nodes linked by one or more entire production rules to be valid substructures. The formal characterization is given in (Collins and Duffy, 2002) and is reported hereafter:

Let $\mathcal{F} = \{f_1, f_2, \ldots, f_{|\mathcal{F}|}\}$ be the set of tree fragments and $\chi_i(n)$ be an indicator function, equal to 1 if the target f_i is rooted at node n and equal to 0 otherwise. A tree kernel function over T_1 and T_2 is defined as $TK(T_1, T_2) = \sum_{n_1 \in N_{T_1}} \sum_{n_2 \in N_{T_2}} \Delta(n_1, n_2)$, where N_{T_1} and N_{T_2} are the sets of nodes in T_1 and T_2, respectively and $\Delta(n_1, n_2) = \sum_{i=1}^{|\mathcal{F}|} \chi_i(n_1)\chi_i(n_2)$.

Δ function counts the number of subtrees rooted in n_1 and n_2 and can be evaluated 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 (i.e. they are pre-terminal symbols) then $\Delta(n_1, n_2) = \lambda$;

3. if the productions at n_1 and n_2 are the same, and n_1 and n_2 are not pre-terminals then $\Delta(n_1, n_2) = \lambda \prod_{j=1}^{l(n_1)} (1 + \Delta(c_{n_1}(j), c_{n_2}(j)))$, where $l(n_1)$ is the number of children of n_1, $c_n(j)$ is the j-th child of node n and λ is a decay factor penalizing larger structures.

Figure 1 shows some fragments (out of the overall 472) of the syntactic parse tree on the left, which is derived from the text T4. These fragments satisfy the constraint that grammatical rules cannot be broken. For example, *(VP (VBN (murdered) NNP (JFK)))* is a valid fragment whereas *(VP (VBN (murdered))* is not. One drawback of such kernel is that two sentences expressing similar semantics but with different lexicals produce structures which will not be matched. For example, after the vertical bar there is a fragment, extracted from the parse tree of a semantically identical sentences: `In 1963 Oswald killed Kennedy`. In this case, much less matches will be counted by the kernel function applied to such parse trees and the one of T4. In particular, the complete VP subtree will not be matched.

To tackle this problem the Syntactic Semantic Tree Kernel (SSTK) was defined in (Bloehdorn and Moschitti, 2007a); hereafter, we report its definition.

4.2 Syntactic Semantic Tree kernels (SSTK)

An SSTK produces all the matches of STK. Moreover, the fragments, which are identical but for their lexical nodes, produce a match proportional to the product of the similarity between their corresponding words. This is a sound definition. Indeed, since the structures are the same, each word in position i of the first fragment can be associated with a word located in the same position i of the second fragment. More formally, the fast evaluation of Δ for STK can be used for computing the semantic Δ for SSTK by simply adding the following step

0. if n_1 and n_2 are pre-terminals and $label(n_1) = label(n_2)$ then $\Delta(n_1, n_2) = \lambda \kappa_{\mathcal{S}}(ch^1_{n_1}, ch^1_{n_2})$,

where $label(n_i)$ is the label of node n_i and $\kappa_{\mathcal{S}}$ is a term similarity kernel, e.g. based on Wikipedia, Wordnet or BNC, defined in Section 3. Note that: (a) since n_1 and n_2 are pre-terminals of a parse tree

they can have only one child (i.e. $ch^1_{n_1}$ and $ch^1_{n_2}$) and such children are words and (b) Step 2 of the original Δ evaluation is no longer necessary.

For example, the fragments: *(VP (VBN (murdered) NNP (JFK)))* has a match with *(VP (VBN (killed) NNP (Kennedy)))* equal to $\kappa_{\mathcal{S}}(murdered, kill) \times \kappa_{\mathcal{S}}(JFK, Kennedy)$.

Beside the novelty of taking into account tree fragments that are not identical it should be noted that the lexical semantic similarity is constrained in syntactic structures, which limit errors/noise due to incorrect (or, as in our case, not provided) word sense disambiguation.

Finally, it should be noted that when a valid kernel is used in place of $\kappa_{\mathcal{S}}$, SSTK is a valid kernel for definition of convolution kernels (Haussler, 1999). Since the matrix P derived by applying LSA produces a semi-definite matrix (see (Cristianini and Holloway, 2001)) we can always use the similarity matrix derived by LSA in SSTK. In case of Wordnet, the validity of the kernel will depend of the kind of similarity used. In our experiments, we have carried out single value decomposition and we have verified that our Wordenet matrices, Path and WUP, are indeed positive semi-definite.

5 Kernels for Textual Entailment Recognition

In this section, we describe how we use the syntactic tree kernel (STK) and the semantic/syntactic tree kernel (SSTK) for modeling lexical-syntactic kernels for textual entailment recognition. We build on the kernel described in (Zanzotto and Moschitti, 2006; Zanzotto et al., 2009) that can model lexical-syntactic rules with variables (i.e., first-order rules).

5.1 Anchoring and pruning

Kernels for modeling lexical-syntactic rules with variables presuppose that words in texts T are explicitly related to words in hypotheses H. This correlation is generally called anchoring and it is implemented with placeholders that co-index the syntactic trees derived from T and H. Words and intermediate nodes are co-indexed when they are equal or similar. For example, in the pair:

$T_8 \Rightarrow ? H_8$

T_8	*"Lee Harvey Oswald was born in New Orleans, Louisiana, and was of English, German, French and Irish ancestry. In 1963 1 Oswald murdered JFK 2"*
H_8	*"JFK 2 died in 1963 1"*

Moreover, the set of anchors also allows us to prune fragments of the text T that are irrelevant for the final decision: we can discard sentences or phrases uncovered by placeholders. For example, in the pair $\langle T_8, H_8 \rangle$, we can infer that "*Lee H... ancestry*" is not a relevant fragment and remove it. This allows us to focus on the critical part for determining the entailment value.

5.2 Kernels for capturing lexical-syntactic rules

Once placeholders are available in the entailment pairs, we can apply the model proposed in (Zanzotto et al., 2009). This derives the maximal similarity between pairs of T and H based on the lexico-syntactic information encoded by the syntactic parse trees of T and H enriched with placeholders. More formally, the original kernel is based on the following equation:

$$maxSTK(\langle T, H \rangle, \langle T', H' \rangle) = max_{c \in C} \quad (5)$$
$$(STK(t(T, c), t(T', i)) + STK(t(H, c), t(H', i)),$$

where: (i) C is the set of all bijective mappings between the placeholders (i.e., the possible variables) from $\langle T, H \rangle$ into $\langle T', H' \rangle$; (ii) $c \in C$ is a substitution function, which implements such mapping; (iii) $t(\cdot, c)$ returns the syntactic tree enriched with placeholders replaced by means of the substitution c; and (iv) $STK(\tau_1, \tau_2)$ is a tree kernel function.

The new semantic-syntactic kernel for lexical-syntactic rules, maxSSTK, substitutes STK with SSTK in Eq. 5 thus enlarging the coverage of the matching between the pairs of texts and the pairs of hypotheses.

6 Experiments

The aim of the experiments is to investigate if our RTE system exploiting syntactic semantic kernels (SSTK) can effectively derive generalized lexico-syntactic rules. In more detail, first, we determine the best lexical similarity suitable for the task, i.e.

		No Semantic	Wiki	BNC	Path	WUP
RTE2	**j = 1**	63.12	63.5	62.75	62.88	**63.88**
	j = 0.9	63.38	**64.75**	62.26	63.88	64.25
RTE3	**j = 1**	66.88	**67.25**	**67.25**	66.88	66.5
	j = 0.9	67.25	**67.75**	67.5	67.12	67.38
RTE5	**j = 1**	65.5	**66.5**	65.83	66	66
	j = 0.9	65.5	**66.83**	65.67	66	66.33

Table 1: Accuracy of plain (WOK+STK+maxSTK) and Semantic Lexico-Syntactic (WOK+SSTK+maxSSTK) Kernels. The latter according to different similarities

distributional vs. Wordnet-based approaches. Second, we derive qualitative and quantitative properties, which justify the selection of one with respect to the other.

For this purpose, we tested four different version of SSTK, i.e. using Path, WUP, BNC and WIKI lexical similarities on three different RTE datasets. These correspond to the three different challenges in which the development set was provided.

6.1 Experimental Setup

We used the data from three recognizing textual entailment challenge: RTE2 (Bar-Haim et al., 2006), RTE3 (Giampiccolo et al., 2007), and RTE5, along with the standard split between training and test sets. We did not use RTE1 as it was differently built from the others and RTE4 as it does not contain the development set.

We used the following publicly available tools: the Charniak Parser (Charniak, 2000) for parsing sentences and SVM-light-TK (Moschitti, 2006; Joachims, 1999), in which we coded our new kernels for RTE. Additionally, we used the Jiang&Conrath (J&C) distance (Jiang and Conrath, 1997) computed with `wn::similarity` package (Pedersen et al., 2004) to measure the similarity between T and H. This similarity is also used to define the text-hypothesis word overlap kernel (WOK).

The distributional semantics is captured by means of LSA: we used the java Latent Semantic Indexing (jLSI) tool (Giuliano, 2007). In particular, we precomputed the word-pair matrices for RTE2, RTE3, and RTE5. We built different LSA matrices from the British National Corpus (BNC) and Wikipedia (Wiki). The British National Corpus (BNC) is a balanced synchronic text corpus containing 100 million words with morpho-syntactic annotation. For Wikipedia, we created a model from the 200,000 most visited Wikipedia articles, after cleaning the unnecessary markup tags. Articles are our documents for creating the term-by-document matrix. Wikipedia provides the largest coverage knowledge resource developed by a community, besides the noticeable coverage of named entities. This further motivates the design of a similarity measure. We also consider two typical WordNet similarities (i.e., Path and WUP, respectively) as described in Sec. 3.1.

The main RTE model that we consider is constituted by three main kernels:

- WOK, i.e. the kernel based on only the text-hypothesis lexical overlapping words (this is an intra-pair similarity);
- STK, i.e. the sum of the standard tree kernel (see Section 4.1) applied to the two text parse-trees and the two hypothesis parse trees;
- SSTK, i.e. the same as STK with the use of lexical similarities as explained in Section 4.2;
- maxSTK and maxSSTK, i.e. the kernel for RTE, illustrated in Section 5.2, where the latter exploits similarity since it uses SSTK in Eq. 5.

Note that the model presented in (Zanzotto et al., 2009), our baseline, corresponds to the combination kernel: WOK+maxSTK. In this paper, in addition to the role of lexical similarities, we also study several combinations (we just need to sum the separated kernels), i.e. WOK+STK+maxSTK, SSTK+maxSSTK, WOK+SSTK+maxSSTK and WOK+maxSSTK.

Finally, we measure the performance of our system with the standard accuracy and then we determine the statistical significance by using the model

		STK	SSTK	maxSTK	maxSSTK	STK+maxSTK	SSTK+maxSSTK	$\emptyset$
RTE2	+WOK	61.5	61.12	63.88	64.12	63.12	63.50	60.62
		52.62	52.75	61.25	59.38	61.25	58.75	-
RTE3	+WOK	66.38	66.5	66.5	67.0	66.88	67.25	66.75
		53.25	54.5	62.25	64.38	63.12	63.62	-
RTE5	+WOK	62.0	62.0	64.83	64.83	65.5	66.5	60.67
		54.33	57.33	63.33	62.67	61.83	62.67	-

Table 2: Comparing different lexico-syntactic kernels with Wiki-based semantic kernels

described in (Yeh, 2000) and implemented in (Padó, 2006).

6.2 Distributional vs. WordNet-based Semantics

The first experiment compares the basic kernel, i.e. WOK+STK+maxSTK, with the new semantic kernel, i.e. WOK+SSTK+maxSSTK, where SSTK and maxSSTK encode four different kinds of similarities, BNC, WIKI, WUP and Path. The aim is twofold: understanding if semantic similarities can be effectively used to derive generalized lexico-syntactic rules and to determine the best similarity model.

Table 1 shows the results according to No Semantics, Wiki, BNC, Path and WUP. The three pairs of rows represent the results over the three different datasets, i.e., RTE2, RTE3, and RTE5. For each pair, we have two rows representing a different j parameter of SVM. An increase of j augments the weight of positive with respect to negative examples and during learning it tunes-up the Recall/Precision rate. We use two values $j = 1$ (the default value) and $j = 0.9$ (selected during a preliminary experiment on a validation set on RTE2). $j = 0.9$ was used to minimally increase the Precision, considering that the semantic model tends to improve the Recall.

The results show that:

- WIKI semantics constantly improves the basic kernel (no Semantics) for any datasets or parameter.
- The distributional semantics is almost always better than the WordNet-based one.
- In one case WUP improves WIKI, i.e. 63.88 vs 63.5 and in another case BNC reaches WIKI, i.e. 67.25 but this happens for the default values of the j parameters, i.e. $j = 1$, which was not selected by our limited parameter validation.

Finally, the difference between the accuracy of the best WIKI kernels and the No Semantic kernels are statistically significant ($p << 0.05$).

6.3 Kernel Comparisons

The previous experiments (Sec. 6.2) show that Wikipedia-based distributional semantics provides an effective similarity to generalize lexico-syntactic rules (features). As our RTE kernel is a composition of other basic kernels, we experimented with different combinations to understand the role of each component. Moreover, to obtain results independent of parameterization we used the default parameter j.

Table 2 reports the accuracy of different kernels and their combinations on different RTE datasets. Each row describes the results for each dataset and it is split in two according to the use of WOK or not in the RTE model. In the each column, the different kernels are reported. For example, the entry in the 4th column and the 2nd row refers to the accuracy of SSTK in combination with WOK, i.e. WOK+SSTK for the RTE2.

We observe that: first WOK produces a very high accuracy in RTE challenges, i.e. 60.62, 66.75 and 60.67 and it is an essential component of RTE systems since its ablation always causes a large accuracy decrease. This is reasonable as the major source of information to establish entailment between sentences is their word overlap.

Second, STK and SSTK, when added to WOK, improve it on RTE2 and RTE5 but do not improve it on RTE3. This suggests a difficulty of exploiting syntactic information for RTE3.

Third, maxSTK+WOK relevantly improves WOK on RTE2 and RTE5 but fails in RTE3. Again, the syntactic rules (with variables) which this kernel

	BNC	WN	WIKI
RTE2	0.55	0.42	0.83
RTE3	0.54	0.41	0.83
RTE5	0.45	0.34	0.82

Table 3: Coverage of the different resources for the words of the three datasets

	Average Acc.	Our rank	# participants
RTE2	59.8	3rd	23
RTE3	64.5	4th	26
RTE5	61.5	4th	20

Table 5: Comparison with other approaches to RTE

can provide are not enough general for RTE3. In contrast, maxSSTK+WOK improves WOK on all datasets thanks to its generalization ability.

Finally, STK and SSTK added to maxSTK+WOK or to maxSSTK+WOK tend to produce an accuracy increase, although not in every condition.

7 Discussion

7.1 Coverage and efficiency

As already mentioned, the practical use of Wikipedia to design lexical similarities is motivated by a large coverage. Deriving similarities from other resources such as WordNet is more time-consuming. To prove our claim, we performed an analysis on the coverage and efficiency in computing the pair term similarity.

Table 3 shows the coverage of the content words of the three datasets. The coverage of Wikipedia is about two times more than the other resources in all experimented datasets.

Speed	Milliseconds
LSA	0.54
WN with POS	5.3
WN without POS	15.2

Table 4: The comparison in terms of speed calculated over 10000 pairs after loading the model.

Moreover, Table 4 shows that the computation of the LSA matrix on Wikipedia is faster than using the WordNet similarity software (Pedersen et al., 2004). Even if the accuracy of some WordNet models can reach the one based on Wikipedia, the latter is preferable for the smaller computational cost.

7.2 Comparison with previous work

The results of our models show that lexical semantics for building more effective lexical-syntactic rules is promising. Here, we compare our approaches with other RTE systems to show that our results are indeed state-of-the-art. Unfortunately, deriving a reasonable accuracy value to represent the state-of-the-art is extremely difficult as many factors can determine the final score. For example, the best systems in RTE2 and RTE3 (Giampiccolo et al., 2007) have an accuracy 10% higher than the others but they generally use resources that are not publicly available.

Table 5 shows the average accuracy of the participant systems, the rank of our system that we propose in this paper and the number of participants. Our model accuracy is absolutely above the average and it is ranked at the top positions. We can also carry out a finer comparison with respect to RTE2 (Bar-Haim et al., 2006). Our system results are the best when compared with systems using semantic models based on FrameNet, indeed the best ranked system in this class, i.e., (Burchardt et al., 2007), scores only 62.5. Among systems using logical inference, our model is instead the 3rd out of 8 systems using logical inference that perform worse than ours. Finally, it is the 2nd among systems using supervised machine learning models.

8 Conclusion

In this paper we presented a model to effectively include semantics in lexical-syntactic features for textual entailment recognition. We have experimentally shown that LSA-derived lexical semantics embedded in syntactic structures is a promising approach. The model that we have presented is one of the best system in the RTE challenges. Additionally, in contrast to many other methods it does not require large sets of handcrafted or corpus extracted lexical-syntactic rules.

Acknowledgements

The research of Alessandro Moschitti has been partially supported by Trustworthy Eternal Systems via Evolving Software, Data and Knowledge (EternalS, project number FP7 247758).

References

E. Agirre, E. Alfonseca, K. Hall, J. Kravalova, M. Paşca, and A. Soroa. 2009. A study on similarity and relatedness using distributional and wordnet-based approaches. In *NAACL '09: Proc. HLT/NAACL*.

R. Bar-Haim, I. Dagan, B. Dolan, L. Ferro, D. Giampiccolo, and I. Magnini, B. Szpektor. 2006. The ii PASCAL recognising textual entailment challenge. In *Proc. of the II PASCAL Challenges Workshop*.

R. Bar-Haim, I. Dagan, I. Greental, and E. Shnarch. 2007. Semantic inference at the lexical-syntactic level. In *AAAI'07: Proc. of the 22nd national conference on Artificial intelligence*.

R. Bar-Haim, J. Berant, and I. Dagan. 2009. A compact forest for scalable inference over entailment and paraphrase rules. In *Proc. of EMNLP*.

R. Basili, M. Cammisa, and A. Moschitti. 2005. Effective use of wordnet semantics via kernel-based learning. In *CoNLL*.

R. Basili, M. Cammisa, and A. Moschitti. 2006. A semantic kernel to classify texts with very few training examples. In *Informatica*.

S. Bloehdorn and A. Moschitti. 2007a. Combined syntactic and semantic kernels for text classification. In *ECIR*.

S. Bloehdorn and A. Moschitti. 2007b. Structure and semantics for expressive text kernels. In *Proc. of CIKM '07*.

S. Bloehdorn, R. Basili, M. Cammisa, and A. Moschitti. 2006. Semantic kernels for text classification based on topological measures of feature similarity. In *Proc. of ICDM 06, Hong Kong, 2006*.

J. Bos and K. Markert. 2005. Recognising textual entailment with logical inference. In *HLT '05: Proc. of the conference on HLT and EMNLP*.

A. Burchardt, N. Reiter, S. Thater, and A. Frank. 2007. Semantic Approach to Textual Entailment: System Evaluation and Task Analysis. In *Proc. of the 3rd-PASCAL Workshop on Textual Entailment*, Prague.

Y. S. Chan and H. T. Ng. 2005. Word sense disambiguation with distribution estimation. In *Proc. of IJCAI'05*.

Eugene Charniak. 2000. A maximum-entropy-inspired parser. In *Proc. of the 1st NAACL conference*.

M. Collins and N. Duffy. 2002. New ranking algorithms for parsing and tagging: kernels over discrete structures, and the voted perceptron. In *Proc. of ACL '02*.

N. Cristianini and R. Holloway. 2001. Latent semantic kernels.

G. Dinu and R. Wang. 2009. Inference rules and their application to recognizing textual entailment. In *Proc. of the EACL '09*.

D. Giampiccolo, B. Magnini, Ido Dagan, and B. Dolan. 2007. The third pascal recognizing textual entailment challenge. In *Proc. of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*.

Claudio Giuliano. 2007. jLSI a for latent semantic indexing. *http://tcc.itc.it/research/textec/tools-resources/jLSI.html*.

A. D. Haghighi, A. Y. Ng, and C. D. Manning. 2005. Robust textual inference via graph matching. In *HLT '05: Proc. of the conference on HLT and EMNLP*.

David Haussler. 1999. Convolution kernels on discrete structures. Technical report.

J. J. Jiang and D. W. Conrath. 1997. Semantic similarity based on corpus statistics and lexical taxonomy. In *Proc. of the 10th ROCLING*.

Thorsten Joachims. 1999. Making large-scale support vector machine learning practical.

Landauer, Foltz, and Laham. 1998. Introduction to latent semantic analysis. In *Discourse Processes 25*.

D. Lin and P. Pantel. 2001. DIRT-discovery of inference rules from text. In *Proc. of the ACM KDD-01*.

R. Mihalcea, C. Corley, and C. Strapparava. 2006. Corpus-based and knowledge-based measures of text semantic similarity. In *Proc. of AAAI06*.

Alessandro Moschitti. 2006. Making tree kernels practical for natural language learning. In *Proc. of EACL*.

Peter Norvig. 1987. A unified theory of inference for text understanding. Technical report, USA.

Sebastian Padó, 2006. *User's guide to* `sigf`*: Significance testing by approximate randomisation*.

T. Pedersen, S. Patwardhan, and J. Michelizzi. 2004. Wordnet::similarity - measuring the relatedness of concepts. In *Proc. of 5th NAACL*.

D. Roth and M. Sammons. 2007. Semantic and logical inference model for textual entailment. In *Proc. of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*.

I. Szpektor and I. Dagan. 2008. Learning entailment rules for unary templates. In *Proc. of COLING '08*.

M. Tatu and D. Moldovan. 2005. A semantic approach to recognizing textual entailment. In *HLT '05: Proc. of HLT/EMNLP*.

Z. Wu and M. Palmer. 1994. Verb semantics and lexical selection. In *Proc. of ACL*.

Alexander Yeh. 2000. More accurate tests for the statistical significance of result differences. In *Proc. of ACL 2000*, Morristown, NJ, USA.

F. M. Zanzotto and A. Moschitti. 2006. Automatic learning of textual entailments with cross-pair similarities. In *Proc. of ACL '06*.

F. M. Zanzotto, M. Pennacchiotti, and A. Moschitti. 2009. A machine learning approach to textual entailment recognition. *NATURAL LANGUAGE ENGINEERING*.