Recognizing Textual Entailment Using Sentence Similarity based on Dependency Tree Skeletons

Rui Wang and Günter Neumann
LT-lab, DFKI
Stuhlsatzenhausweg 3, 66123 Saarbrücken, Germany
{wang.rui,Neumann}@dfki.de

Abstract

We present a novel approach to RTE that exploits a structure-oriented sentence representation followed by a similarity function. The structural features are automatically acquired from tree skeletons that are extracted and generalized from dependency trees. Our method makes use of a limited size of training data without any external knowledge bases (e.g. WordNet) or handcrafted inference rules. We have achieved an accuracy of 71.1% on the RTE-3 development set performing a 10-fold cross validation and 66.9% on the RTE-3 test data.

1 Introduction

Textual entailment has been introduced as a relation between text expressions, capturing the fact that the meaning of one expression can be inferred from the other (Dagan and Glickman, 2004). More precisely, textual entailment is defined as "... *a relationship between a coherent text **T** and a language expression, which is considered as a hypothesis, **H**. We say that **T** entails **H** (**H** is a consequent of **T**), denoted by **T** $\Rightarrow$ **H**, if the meaning of **H**, as interpreted in the context of **T**, can be inferred from the meaning of **T**.*"

Table 1 displays several examples from the RTE-3 development set. For the third pair (id=410) the key knowledge needed to decide whether the entailment relation holds is that "[PN1]*'s wife,* [PN2]" entails "The name of [PN1]*'s wife is* [PN2]", although **T** contains much more (irrelevant) information. On the other hand, the first pair (id=1) requires an understanding of concepts with opposite meanings (i.e. "*buy*" and "*sell*"), which is a case of semantic entailment.

The different sources of possible entailments motivated us to consider the development of specialized entailment strategies for different NLP tasks. In particular, we want to find out the potential connections between entailment relations belonging to different linguistic layers for different applications.

In this paper, we propose a novel approach towards structure-oriented entailment based on our empirical discoveries from the RTE corpora: 1) **H** is usually textually shorter than **T**; 2) not all information in **T** is relevant to make decisions for the entailment; 3) the dissimilarity of relations among the same topics between **T** and **H** are of great importance.

Based on the observations, our primary method starts from **H** to **T** (i.e. in the opposite direction of the entailment relation) so as to exclude irrelevant information from **T**. Then corresponding key topics and predicates of both elements are extracted. We then represent the structural differences between **T** and **H** by means of a set of Closed-Class Symbols. Finally, these acquired representations (named Entailment Patterns - EPs) are classified by means of a subsequence kernel.

The Structure Similarity Function is combined with two robust backup strategies, which are responsible for cases that are not handled by the EPs. One is a Triple Similarity Function applied on top of the local dependency relations of **T** and **H**; the other is a simple Bag-of-Words (BoW) approach that calculates the overlapping ratio of **H** and **T**. Together, these three methods deal with different entailment cases in practice.

Proceedings of the Workshop on Textual Entailment and Paraphrasing, pages 36–41,

Id	Task	Text	Hypothesis	Entails?
1	IE	*The sale was made to pay Yukos' US$ 27.5 billion tax bill, Yuganskneftegaz was originally sold for US$ 9.4 billion to a little known company Baikalfinansgroup which was later* ***bought*** *by the Russian state-owned oil*	*Baikalfinansgroup was* ***sold*** *to Rosneft.*	YES
390	IR	***Typhoon*** *Xangsane* ***lashed*** *the* ***Philippine*** *capital on Thursday, grounding flights, halting vessels and closing schools and markets after triggering fatal flash floods in the centre of the country.*	*A* ***typhoon batters*** *the* ***Philippines***.	YES
410	QA	*(Sentence 1 ...). Along with the first lady's mother, Jenna Welch, the weekend gathering includes the president's parents, former President* ***George H.W. Bush and his wife, Barbara*** *; his sister Doro Koch and her husband, Bobby; and his brother, Marvin, and his wife, Margaret.*	*The name of George H.W. Bush's wife is Barbara.*	YES
739	SUM	*The* ***FDA*** *would not say in which* ***states*** *the* ***pills*** *had been sold, but instead recommended that customers determine whether products they bought are being recalled by checking the store* ***list*** *on the* ***FDA*** *Web site, and the batch* ***list***. *The batch numbers appear on the container's label.*	*The* ***FDA*** *provided a* ***list*** *of* ***states*** *in which the* ***pills*** *have been*	NO

Table 1 Examples from RTE-3

2 Related Work

Conventional methods for RTE define measures for the similarity between **T** and **H** either by assuming an independence between words (Corley and Mihalcea, 2005) in a BoW fashion or by exploiting syntactic interpretations. (Kouylekov and Magnini, 2006) explore a syntactic tree editing distance to detect entailment relations. Since they calculate the similarity between the two dependency trees of **T** and **H** directly, the noisy information may decrease accuracy. This observation actually motivated us to start from **H** towards the most relevant information in **T**.

Logic rules (as proposed by (Bos and Markert, 2005)) or sequences of allowed rewrite rules (as in (de Salvo Braz et al., 2005)) are another fashion of tackling RTE. One the best two teams in RTE-2 (Tatu et al., 2006) proposed a knowledge representation model which achieved about 10% better performance than the third (Zanzotto and Moschitti, 2006) based on their logic prover. The other best team in RTE-2 (Hickl et al., 2006) automatically acquired extra training data, enabling them to achieve about 10% better accuracy than the third as well. Consequently, obtaining more training data and embedding deeper knowledge were expected

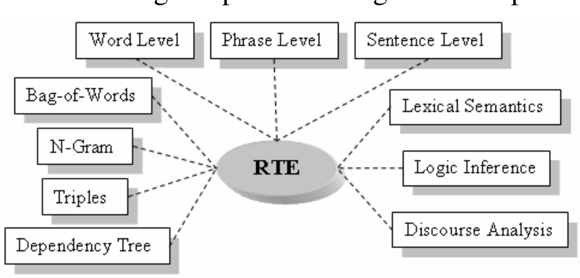

Figure 1 Overview of RTE

to be the two main directions pointed out for future research in the RTE-2 summary statement. However, except for the positive cases of SUM, **T**-**H** pairs are normally not very easy to collect automatically. Multi-annotator agreement is difficult to reach on most of the cases as well. The knowledge-based approach also has its caveats since logical rules are usually implemented manually and therefore require a high amount of specialized human expertise in different NLP areas.

Another group (Zanzotto and Moschitti, 2006) utilized a tree kernel method for cross-pair similarity, which showed an improvement, and this has motivated us to investigate kernel-based methods. The main difference in our method is that we apply subsequence kernels on patterns extracted from the dependency trees of **T** and **H**, instead of applying tree kernels on complete parsing trees. On the one hand, this allows us to discover essential parts indicating an entailment relationship, and on the other hand, computational complexity is reduced.

3 An Overview of RTE

Figure 1 shows the different processing techniques and depths applied to the RTE task. Our work focuses on constructing a similarity function operating between sentences. In detail, it consists of several similarity scores with different domains of locality on top of the dependency structure. Figure 2 gives out the workflow of our system. The main part of the sentence similarity function is the Structure Similarity Function; two other similarity scores are calculated by our backup strategies. The first backup strategy is a straightforward BoW method that we will not present in this paper (see more details in (Corley and Mihalcea, 2005));

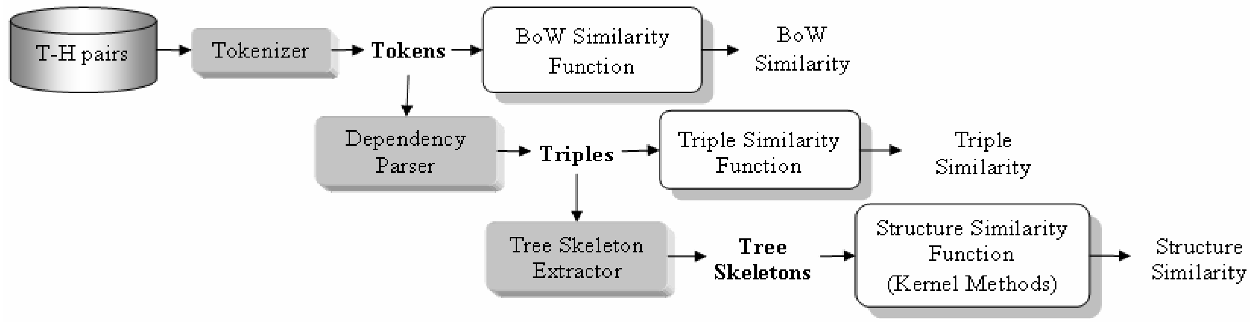

Figure 2 Workflow of the System

while the second one is based on a triple set representation of sentences that expresses the local dependency relations found by a parser[1].

A dependency structure consists of a set of triple relations (TRs). A TR is of the form <*node1, relation, node2*>, where *node1* represents the head, *node2* the modifier and *relation* the dependency relation. Chief requirements for the backup system are robustness and simplicity. Accordingly, we construct a similarity function, the Triple Similarity Function (TSF), which operates on two triple sets and determines how many triples of **H**[2] are contained in **T**. The core assumption here is that *the higher the number of matching triple elements, the more similar both sets are, and the more likely it is that* **T** *entails* **H**.

TSF uses an approximate matching function. Different cases (i.e. ignoring either the parent node or the child node, or the relation between nodes) might provide different indications for the similarity of **T** and **H**. In all cases, a successful match between two nodes means that they have the same lemma and POS. We then sum them up using different weights and divide the result by the cardinality of **H** for normalization. The different weights learned from the corpus indicate that the "amount of missing linguistic information" affect entailment decisions differently.

4 Workflow of the Main Approach

Our Structure Similarity Function is based on the hypothesis that *some particular differences between* ***T*** *and* ***H*** *will block or change the entailment relationship*. Initially we assume when judging the entailment relation that it holds for each **T-H** pair (using the default value "YES"). The major steps are as follows (see also Figure 2):

4.1 Tree Skeleton Extractor

Since we assume that **H** indicates how to extract relevant parts in **T** for the entailment relation, we start from the Tree Skeleton of **H** (TS_H). First, we construct a set of keyword pairs using all the nouns that appear in both **T** and **H**. In order to increase the hits of keyword pairs, we have applied a partial search using stemming and some word variation techniques on the substring level. For instance, the pair (id=390) in Table 1 has the following list of keyword pairs,

```
<Typhoon_Xangsane ## typhoon,
Philippine ## Philippines>
```

Then we mark the keywords in the dependency trees of **T** and **H** and extract the sub-trees by ignoring the inner yields. Usually, the Root Node of **H** (RN_H) is the main verb; all the keywords are contained in the two spines of TS_H (see Figure 3). Note that in the Tree Skeleton of **T** (TS_T), 1) the Root Node (RN_T) can either be a verb, a noun or even a dependency relation, and 2) if the two Foot Nodes (FNs) belong to two sentences, a dummy node is created that connects the two spines.

Thus, the prerequisite for this algorithm is that TS_H has two spines containing all keywords in **H**, and **T** satisfies this as well. For the RTE-3 development set, we successfully extracted tree skele-

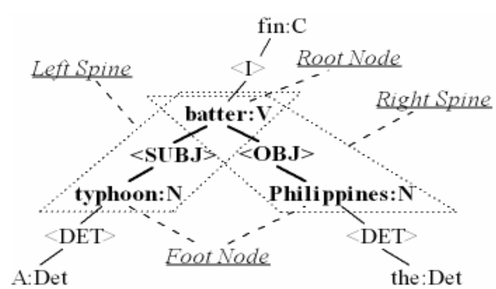

Figure 3 Example of a Tree Skeleton

[1] We are using Minipar (Lin, 1998) and Stanford Parser (Klein and Manning, 2003) as preprocessors, see also sec. 5.2.
[2] Note that henceforth T and H will represent either the original texts or the dependency structures.

tons from 254 pairs, i.e., 32% of the data is covered by this step, see also sec. 5.2.

Next, we collapse some of the dependency relation names from the parsers to more generalized tag names, e.g., collapsing <OBJ2> and <DESC> to <OBJ>. We group together all nodes that have relation labels like <CONJ> or <NN>, since they are assumed to refer to the same entity or belong to one class of entities sharing some common characteristics. Lemmas are removed except for the keywords. Finally, we add all the tags to the CCS set.

Since a tree skeleton TS consists of spines connected via the same root node, TS can be transformed into a sequence. Figure 4 displays an example corresponding to the second pair (id=390) of Table 1. Thus, the general form of a sequential representation of a tree skeleton is:

`LSP #RN# RSP`

where LSP represents the Left Spine, RSP represents the Right Spine, and RN is the Root Node. On basis of this representation, a comparison of the two tree skeletons is straightforward: 1) merge the two LSPs by excluding the longest common prefix, and 2) merge the two RSPs by excluding the longest common suffix. Then the Spine Difference (SD) is defined as the remaining infixes, which consists of two parts, SD_T and SD_H. Each part can be either empty (i.e. ε) or a CCS sequence. For instance, the two SDs of the example in Figure 4 (id=390) are (LSD – Left SD; RSD – Right SD; ## is a separator sign):

LSD_T(N) ## LSD_H(ε)

RSD_T(ε) ## RSD_H(ε)

We have observed that two neighboring dependency relations of the root node of a tree skeleton (<SUBJ> or <OBJ>) can play important roles in predicting the entailment relation as well. Therefore, we assign them two extra features named Verb Consistence (VC) and Verb Relation Consistence (VRC). The former indicates whether two root nodes have a similar meaning, and the latter indicates whether the relations are contradictive (e.g. <SUBJ> and <OBJ> are contradictive).

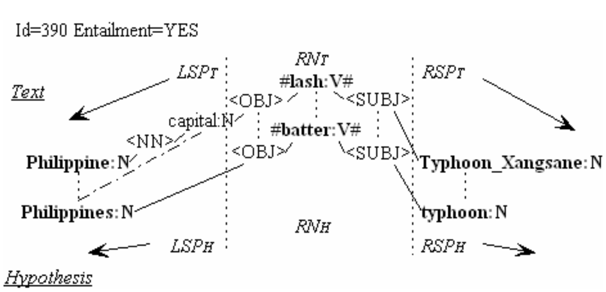

Figure 4 Spine Merging

We represent the differences between TS_T and TS_H by means of an Entailment Pattern (EP), which is a quadruple <LSD, RSD, VC, VRC>. VC is either true or false, meaning that the two RNs are either consistent or not. VRC has ternary value, whereby 1 means that both relations are consistent, -1 means at least one pair of corresponding relations is inconsistent, and 0 means RN_T is not a verb.[3] The set of EPs defines the feature space for the subsequence kernels in our Structure Similarity Function.

4.2 Structure Similarity Function

We define the function by constructing two basic kernels to process the LSD and RSD part of an EP, and two trivial kernels for VC and VRC. The four kernels are combined linearly by a composite kernel that performs binary classification on them.

Since all spine differences SDs are either empty or CCS sequences, we can utilize subsequence kernel methods to represent features implicitly, cf. (Bunescu and Mooney, 2006). Our subsequence kernel function is:

$$K_{subsequence}(<T,H>,<T',H'>) = \sum_{i=1}^{|T|}\sum_{i'=1}^{|T'|} K_{CCS}(CCS_i, CCS_{i'}) + \sum_{j=1}^{|H|}\sum_{j'=1}^{|H'|} K_{CCS}(CCS_j, CCS_{j'})$$

whereby *T* and *H* refers to all spine differences SDs from **T** and **H**, and $|T|$ and $|H|$ represent the cardinalities of SDs. The function $K_{CCS}(CCS, CCS')$ checks whether its arguments are equal.

Since the RTE task checks the relationship between **T** and **H**, we need to consider collocations of some CCS subsequences between *T* and *H* as well. Essentially, this kernel evaluates the similarity of **T** and **H** by means of those CCS subsequences appearing in both elements. The kernel function is as follows:

$$K_{collocation}(<T,H>,<T',H'>) = \sum_{i=1}^{|T|}\sum_{i'=1}^{|T'|}\sum_{j=1}^{|H|}\sum_{j'=1}^{|H'|} K_{CCS}(CCS_i, CCS_{i'}) \cdot K_{CCS}(CCS_j, CCS_{j'})$$

On top of the two simple kernels, K_{VC}, and K_{VRC}, we use a composite kernel to combine them linearly with different weights:

$$K_{composite} = \alpha K_{subsequence} + \beta K_{collocation} + \gamma K_{VC} + \delta K_{VRC} ,$$

[3] Note that RN_H is guaranteed to be a verb, because otherwise the pair would have been delegated to the backup strategies.

where γ and δ are learned from the training corpus; α=β=1.

5 Evaluation

We have evaluated four methods: the two backup systems as baselines (BoW and TSM, the Triple Set Matcher) and the kernel method combined with the backup strategies using different parsers, Minipar (Mi+SK+BS) and the Stanford Parser (SP+SK+BS). The experiments are based on RTE-3 Data[4]. For the kernel-based classification, we used the classifier SMO from the WEKA toolkit (Witten and Frank, 1999).

5.1 Experiment Results

RTE-3 data include the Dev Data (800 T-H pairs, each task has 200 pairs) and the Test Data (same size). Experiment A performs a 10-fold cross-validation on Dev Data; Experiment B uses Dev Data for training and Test Data for testing cf. Table 2 (the numbers denote accuracies):

Systems\Tasks	**IE**	**IR**	**QA**	**SUM**	**All**
Exp A: 10-fold Cross Validation on RTE-3 Dev Data					
BoW	54.5	70	76.5	**68.5**	67.4
TSM	53.5	60	68	62.5	61.0
Mi+SK+BS	**63**	74	79	68.5	**71.1**
SP+SK+BS	**60.5**	70	81.5	68.5	**70.1**
Exp B: Train: Dev Data; Test: Test Data					
BoW	54.5	66.5	76.5	56	63.4
TSM	54.5	62.5	66	54.5	59.4
Mi+SP+SK+BS	58.5	70.5	79.5	59	**66.9**

Table 2 Results on RTE-3 Data

For the IE task, Mi+SK+BS obtained the highest improvement over the baseline systems, suggesting that the kernel method seems to be more appropriate if the underlying task conveys a more "relational nature." Improvements in the other tasks are less convincing as compared to the baselines. Nevertheless, the overall result obtained in experiment B would have been among the top 3 of the RTE-2 challenge. We utilize the system description table of (Bar-Haim et al., 2006) to compare our system with the best two systems of RTE-2 in Table 3[5]:

Systems	Lx	Ng	Sy	Se	**LI**	C	M	B	**L**
Hickl et al.	X	X	X	X		X	X		**X**
Tatu et al.	X				**X**			X	
Ours		X	X				X		

Table 3 Comparison with the top 2 systems in RTE-2.

Note that the best system (Hickl et al., 2006) applies both shallow and deep techniques, especially in acquiring extra entailment corpora. The second best system (Tatu et al., 2006) contains many manually designed logical inference rules and background knowledge. On the contrary, we exploit no additional knowledge sources besides the dependency trees computed by the parsers, nor any extra training corpora.

5.2 Discussions

Table 4 shows how our method performs for the task-specific pairs matched by our patterns:

Tasks	IE	IR	QA	SUM	ALL
ExpA:Matched	**53**	19	23.5	31.5	**31.8**
ExpA:Accuracy	67.9	78.9	91.5	71.4	**74.8**
ExpB:Matched	**58.5**	16	27.5	42	**36**
ExpB:Accuracy	57.2	81.5	90.9	65.5	**68.8**

Table 4 Performances of our method

For IE pairs, we find good coverage, whereas for IR and QA pairs the coverage is low, though it achieves good accuracy. According to the experiments, BoW has already achieved the best performance for SUM pairs cf. Table 2.

As a whole, developing task specific entailment operators is a promising direction. As we mentioned in the first section, the RTE task is neither a one-level nor a one-case task. The experimental results uncovered differences among pairs of different tasks with respect to accuracy and coverage.

On the one hand, our method works successfully on structure-oriented **T**-**H** pairs, most of which are from IE. If both TS_T and TS_H can be transformed into CCS sequences, the comparison performs well, as in the case of the last example (id=410) in Table 1. Here, the relation between "*wife*", "*name*", and "*Barbara*" is conveyed by the punctuation ",", the verb "*is*", and the preposition "*of*". Other cases like the "*work for*" relation of a person and a company or the "*is located in*" relation between two location names are normally conveyed by the preposition "*of*". Based on these findings, taking into account more carefully the lexical semantics based on inference rules of functional words might be helpful in improving RTE.

[4] See (Wang and Neumann, 2007) for details concerning the experiments of our method on RTE-2 data.

[5] Following the notation in (Bar-Haim et al., 2006): Lx: Lexical Relation DB; Ng: N-Gram / Subsequence overlap; Sy: Syntactic Matching / Alignment; Se: Semantic Role Labeling; LI: Logical Inference; C: Corpus/Web; M: ML Classification; B: Paraphrase Technology / Background Knowledge; L: Acquisition of Entailment Corpora.

On the other hand, accuracy varies with **T**-**H** pairs from different tasks. Since our method is mainly structure-oriented, differences in modifiers may change the results and would not be caught under the current version of our tree skeleton. For instance, "*a commercial company*" will not entail "*a military company*", even though they are structurally equivalent.

Most IE pairs are constructed from a binary relation, and so meet the prerequisite of our algorithm (see sec. 4.1). However, our method still has rather low coverage. **T**-**H** pairs from other tasks, for example like IR and SUM, usually contain more information, i.e. more nouns, the dependency trees of which are more complex. For instance, the pair (id=739) in Table 1 contains four keyword pairs which we cannot handle by our current method. This is one reason why we have constructed extra **T**-**H** pairs from MUC, TREC, and news articles following the methods of (Bar-Haim et al., 2006). Still, the overall performance does not improve. All extra training data only serves to improve the matched pairs (about 32% of the data set) for which we already have high accuracy (see Table 4). Thus, extending coverage by machine learning methods for lexical semantics will be the main focus of our future work.

6 Conclusions and Future Work

Applying different RTE strategies for different NLP tasks is a reasonable solution. We have utilized a structure similarity function to deal with the structure-oriented pairs, and applied backup strategies for the rest. The results show the advantage of our method and direct our future work as well. In particular, we will extend the tree skeleton extraction by integrating lexical semantics based on inference rules for functional words in order to get larger domains of locality.

Acknowledgements

The work presented here was partially supported by a research grant from BMBF to the DFKI project HyLaP (FKZ: 01 IW F02) and the EC-funded project QALL-ME.

References

Bar-Haim, R., Dagan, I., Dolan, B., Ferro, L., Giampiccolo, D., Magnini, B. and Szpektor, I. 2006. *The Second PASCAL Recognising Textual Entailment Challenge*. In Proc. of the PASCAL RTE-2 Challenge.

Bos, J. and Markert, K. 2005. *Combining Shallow and Deep NLP Methods for Recognizing Textual Entailment*. In Proc. of the PASCAL RTE Challenge.

Bunescu, R. and Mooney, R. 2006. *Subsequence Kernels for Relation Extraction*. In Advances in Neural Information Processing Systems 18. MIT Press.

Corley, C. and Mihalcea, R. 2005. *Measuring the Semantic Similarity of Texts*. In Proc. of the ACL Workshop on Empirical Modeling of Semantic Equivalence and Entailment.

Dagan, R., Glickman, O. 2004. *Probabilistic textual entailment: Generic applied modelling of language variability*. In PASCAL Workshop on Text Understanding and Mining.

de Salvo Braz, R., Girju, R., Punyaka-nok, V., Roth, D., and Sammons, M. 2005. *An Inference Model for Semantic Entailment in Natural Language*. In Proc. of the PASCAL RTE Challenge.

Hickl, A., Williams, J., Bensley, J., Roberts, K., Rink, B. and Shi, Y. 2006. *Recognizing Textual Entailment with LCC's GROUNDHOG System*. In Proc. of the PASCAL RTE-2 Challenge.

Klein, D. and Manning, C. 2003. *Accurate Unlexicalized Parsing*. In Proc. of ACL 2003.

Kouylekov, M. and Magnini, B. 2006. *Tree Edit Distance for Recognizing Textual Entailment: Estimating the Cost of Insertion*. In Proc. of the PASCAL RTE-2 Challenge.

Lin, D. 1998. *Dependency-based Evaluation of MINIPAR*. In Workshop on the Evaluation of Parsing Systems.

Tatu, M., Iles, B., Slavik, J., Novischi, A. and Moldovan, D. 2006. *COGEX at the Second Recognizing Textual Entailment Challenge*. In Proc. of the PASCAL RTE-2 Challenge.

Wang, R. and Neumann, G. 2007. *Recognizing Textual Entailment Using a Subsequence Kernel Method.* In Proc. of AAAI 2007.

Witten, I. H. and Frank, E. *Weka: Practical Machine Learning Tools and Techniques with Java Implementations*. Morgan Kaufmann, 1999.

Zanzotto, F.M. and Moschitti, A. 2006. Automatic Learning of Textual Entailments with Cross-pair Similarities. In Proc. of ACL 2006.