Learning textual entailment from examples

F.M. Zanzotto
DISCo
University of Milano-Bicocca
Milan, Italy
zanzotto@disco.unimib.it

A. Moschitti, M. Pennacchiotti, M.T. Pazienza
DISP
University of Rome "Tor Vergata"
Rome, Italy
{moschitti,pennacchiotti,
pazienza}@disp.uniroma2.it

Abstract

In this paper we present a novel approach for learning entailment relations from positive and negative examples. We define a similarity between two text-hypothesis pairs based on a syntactic and lexical information. We experimented our model within the RTE 2006 challenge obtaining the accuracy of 63.88% and 62.50% for the two submissions.

1 Introduction

Dermining whether or not a text T entails a hypothesis H is quite complex even when all information needed to derive such inference is explicitly asserted (Zaenen et al., 2005). For example, the sentence T_1: "*At the end of the year, all solid companies pay dividends.*" entails the hypothesis H_1: "*At the end of the year, all solid insurance companies pay dividends.*" but it does not entail the hypothesis H_2: "*At the end of the year, all solid companies pay cash dividends.*"

These implications are considered uncontroversial as they can be derived from the content of T_1, H_1, and H_2. Nevertheless, as H_1 and H_2 contain the same words, the entailment cannot be derived by simply relying on lexical distance (or similanrity). To carry out the correct inference we would need additional rules. For example, by studying the following implication:

$T_3 \Rightarrow H_3$?

T_3	*"All wild animals eat plants that have scientifically proven medicinal properties."*
H_3	*"All wild mountain animals eat plants that have scientifically proven medicinal properties."*

we note that T_3 is structurally (and somehow lexically) similar to T_1 and H_3 is more similar to H_1 than to H_2, thus from $T_1 \Rightarrow H_1$ we may extract rules to derive that $T_3 \Rightarrow H_3$.

The above example suggests that, we should rely on both (1) a *intra-pair* similarity between T and H and (2) a *cross-pair* similarity between two pairs (T', H') and (T'', H''). The latter similarity along with a set of annotated examples allow a learning algorithm to automatically derive syntactic and lexical rules that can solve complex entailment cases.

In this paper, we define a new cross-pair similarity measure based on syntactic interpretations of the sentences. Then, we use such similarity with traditional *intra-pair* similarities to define a novel semantic kernel function. We experimented such kernel with Support Vector Machines (Vapnik, 1995) on the Recognizing Textual Entailment (RTE) challenge test-beds. The comparative results show that we have designed an effective way to automatically learn entailments from examples.

The rest of the paper is organised as follows. Sec. 2 analyses the problems in learning entailment from examples an sketches the model. This latter is decribed in Sec. 3 and refined in Sec. 3.3. Its experimental results are described in Sec. 4.

2 Challenges in learning from examples

In the introductory section, we have shown that to carry out automatic learning from examples, we need to define a cross-pair similarity. Its definition is not straightforward as the measure has to detect whether two pairs (T', H') and (T'', H'') realize the same *rewrite rule*. Besides lexical and syntactic similarity between texts and hypotheses, we need to take into account the movements of *relevant constituents* from T' to H' and comparing with the corresponding movements in the other pair (T'', H''). Only those pairs showing compat-

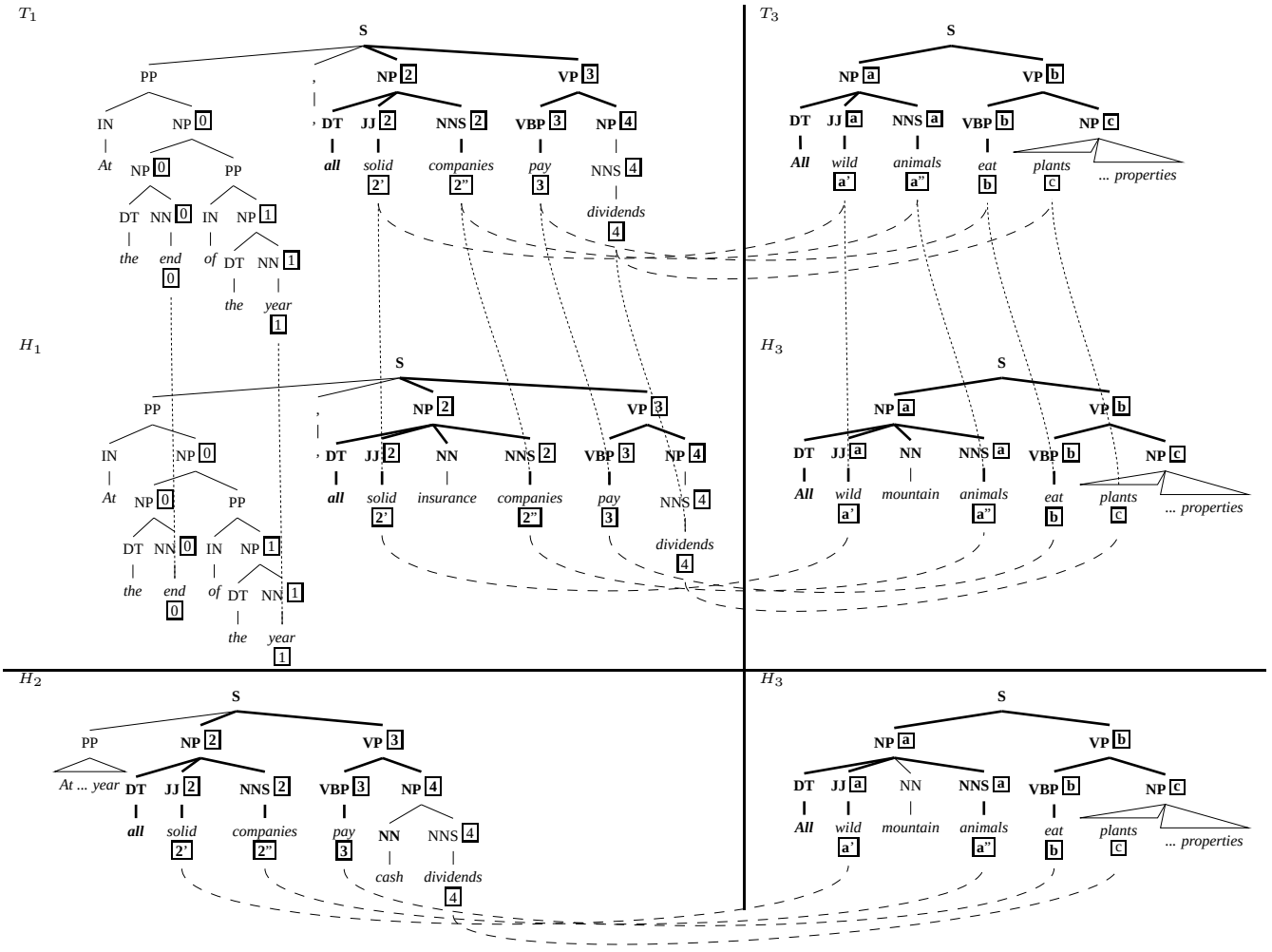

Figure 1: Two similar pairs of sentences both in entailment

ible constituent movements will realize the same rewrite rule. To detect compatible movements we need to carry out three main steps:

First, we detect links between words w_t in T that are equal, similar, or semantically dependent on words w_h in H. The pairs (w_t, w_h) are called *anchors* and we associate them with *placeholders*. For example, in Fig. 1, 2" indicates the *(companies,companies)* anchor between T_1 and H_1. Two pairs can be aligned if they show a similar anchoring. (T_1, H_1) and (T_3, H_3) show compatible constituent movements as the dashed lines that connect placeholders of the two pairs form *loops* with the anchors (dotted lines), e.g. 2' (T_1), 2' (H_1), a' (H_3) and a' (T_3).

Second, we use the anchors to extract the subparse trees that contain most of the information useful to derive the constituent movements (and consequently the entailment). Fig. 1 shows the three pairs (T_1, H_1), (T_1, H_2) and (T_3, H_3) already considered in the introduction. The interesting subtrees are those that exactly covers the anchors of the texts T_1 and T_3 as well as of the hypotheses, H_1, H_2 and H_3 (i.e. the bold subtrees). Although the lexicals in T_3 and H_3 are quite different from those in T_1 and H_1, their bold subtrees are more similar to those of T_1 and H_1 than to T_1 and H_2, respectively. Therefore, to make the entailment decision for (T_3,H_3), we should use the results available for (T_1, H_1).

Finally, we keep track of the constituent movements by finding the correct mapping between the anchor sets. Let A' and A'' be the placeholders of (T', H') and (T'', H''), respectively, without loss of generality we consider $|A'| \geq |A''|$ and we align a subset of A' to A''. The best alignment is the one that maximizes the syntactic and lexical overlapping of the two subtrees induced by the aligned set of anchors.

Let C be the set of all bijective mappings from

$a' \subseteq A' : |a'| = |A''|$ to A'', an element $c \in C$ is a substitution function. We define as the best alignment the one determined by $c_{max} =$

$$argmax_{c \in C}(K_T(t(H', c), t(H'', i)) + K_T(t(T', c), t(T'', i)) \quad (1)$$

where (1) $t(S, c)$ returns the syntactic tree of the hypothesis (text) S with placeholders replaced by means of the substitution c, (2) i is the identity substitution and (3) $K_T(t_1, t_2)$ is a function that measures the similarity between the two trees t_1 and t_2 (for more details see Sec. 3.2) . For example, the c_{max} between (T_1, H_1) and (T_3, H_3) is $\{(2', a'), (2'', a''), (3, b), (4, c)\}$.

3 Similarity Models

In this section we describe how anchors are found at the level of a single pair (T, H) (Sec. 3.1). The anchoring process gives the direct possibility of implementing a inter-pair similarity that can be used as a baseline approach or in combination with the cross-pair similarity. This latter will be implemented with tree kernel functions over syntactic structures (Sec. 3.2).

3.1 Anchoring and Lexical Similarity

The algorithm that we design to find the anchors is based on similarity functions between words or more complex expressions. Our approach is in line with many other researches (e.g. (Corley and Mihalcea, 2005; Glickman et al., 2005)). Hereafter, we describe the differences with the other methods.

Given the set of content words (verbs, nouns, adjectives, and adverbs) W_T and W_H of the two sentences T and H, respectively, the set of anchors $A \subset W_T \times W_H$ is built using a similarity measure between two words $sim_w(w_t, w_h)$. Each element $(w_t, w_h) \in A$ should have the property:

$$sim_w(w_t, w_h) = \max_{w'_t \in W_T} sim_w(w'_t, w_h)$$

Note that according to this property, elements in W_H can partecipate in more than one anchor and conversely more than one element in W_H can be linked to a single element $w \in W_T$.

Moreover, $sim_w(w_t, w_h)$ can be defined using different indicators and resources. First of all, two words are maximally similar if these have the same surface form $w_t = w_h$. Second, we can use one of the WordNet (Miller, 1995) similarities indicated with $d(l_w, l_{w'})$ (in line with what was done in (Corley and Mihalcea, 2005)) and different relation between words such as the lexical entailment between verbs (Ent) and derivationally relation between words (Der). Finally, we use the edit distance measure $lev(w_t, w_h)$ to capture the similarity between words that are missed by the previous analysis for misspelling errors or for the lack of derivationally forms not coded in WordNet.

As result, given the syntactic category $c_w \in \{noun, verb, adjective, adverb\}$ and the lemmatized form l_w of a word w, the similarity measure between two words w and w' is defined as follows:

$$sim_w(w, w') = \begin{cases} 1 & \text{if } w = w' \vee \\ & l_w = l_{w'} \wedge c_w = c_{w'} \vee \\ & ((l_w, c_w), (l_{w'}, c_{w'})) \in Ent \vee \\ & ((l_w, c_w), (l_{w'}, c_{w'})) \in Der \vee \\ & lev(w, w') = 1 \\ d(l_w, l_{w'}) & \text{if } c_w = c_{w'} \wedge d(l_w, l_{w'}) > 0.2 \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

It is worth noticing that, the above measure is not a *pure* similarity measure as it includes the entailment relation that does not represent synonymy or similarity between verbs. To emphasize the contribution of each used resource, in the experimental section, we will compare Eq. 2 with some versions that exclude some word relations.

The above word similarity measure can be used to compute the similarity between T and H. In line with (Corley and Mihalcea, 2005), we define it as:

$$s_1(T, H) = \frac{\sum_{(w_t, w_h) \in A} sim_w(w_t, w_h) \times idf(w_h)}{\sum_{(w_t, w_h) \in A} idf(w_h)} \quad (3)$$

where $idf(w)$ is the inverse document frequency of the word w. For sake of comparison, we consider also the corresponding more classical version that do not apply the inverse document frequency

$$s_2(T, H) = \sum_{(w_t, w_h) \in A} sim_w(w_t, w_h)/|A| \quad (4)$$

From the above intra-pair similarity s_1 and s_2, we can obtain the baseline *cross-pair* similarity based on only lexical information:

$$K_i((T', H'), (T'', H'')) = s_i(T', H') \times s_i(T'', H''), \quad (5)$$

where $i \in \{1, 2\}$. In the next section we define a novel cross-pair similarity that takes into account syntactic evidence by means of tree kernel functions.

3.2 Cross-pair syntactic kernels

Section 2 has shown that to measure the syntactic similarity between two pairs, (T', H') and (T'', H''), we should capture the number of common subtrees between texts and hypotheses that share the same anchoring scheme. The best alignment between anchor sets, i.e. the best substitution c_{max}, can be found with Eq. 1. As the corresponding maximum quantifies the *alignment degree*, we could define a cross-pair similarity as:

$$K_s((T', H'), (T'', H'')) = \max_{c \in C} \Big(K_T(t(H', c), t(H'', i)) + K_T(t(T', c), t(T'', i)) \Big), \quad (6)$$

where as $K_T(t_1, t_2)$ we use the tree kernel function defined in (Collins and Duffy, 2002). This evaluates the number of subtrees shared by t_1 and t_2, thus defining an implicit substructure space.

Formally, given a subtree space $\mathcal{F} = \{f_1, f_2, \ldots, f_{|\mathcal{F}|}\}$, the indicator function $I_i(n)$ is defined. It is 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 $K_T(t_1, t_2) = \sum_{n_1 \in N_{t_1}} \sum_{n_2 \in N_{t_2}} \Delta(n_1, n_2)$, where N_{t_1} and N_{t_2} are the sets of the t_1's and t_2's nodes, respectively. In turn $\Delta(n_1, n_2) = \sum_{i=1}^{|\mathcal{F}|} \lambda^{l(f_i)} I_i(n_1) I_i(n_2)$, where $0 \leq \lambda \leq 1$ and $l(f_i)$ is the number of levels of the subtree f_i. Thus $\lambda^{l(f_i)}$ assigns a lower weight to larger fragments. When $\lambda = 1$, Δ is equal to the number of common fragments rooted at nodes n_1 and n_2. As shown in (Collins and Duffy, 2002), Δ can be computed in $O(|N_{t_1}| \times |N_{t_2}|)$.

The K_T function has been proven to be a valid kernel, i.e. its associated *Gram* matrix is positive-semidefinite. Some basic operations on kernel functions, e.g. the sum, are closed with respect to the set of valid kernels. Thus, if the maximum held such property, Eq. 6 would be a valid kernel and we could use it in kernel based machines like SVMs. Unfortunately, a counterexample illustrated in (Boughorbel et al., 2004) shows that the *max* function does not produce valid kernels in general.

However, we observe that (1) $K_s((T', H'), (T'', H''))$ is a symmetric function since the set of transformation C are always computed with respect to the pair that has the largest anchor set. (2) In (Haasdonk, 2005), it is shown that when kernel functions are not positive semidefinite, SVMs still solve a data separation problem in pseudo Euclidean spaces. The drawback is that the solution may be only a local optimum. Therefore, we can experiment Eq. 6 with SVMs and observe if the empirical results are satisfactory. Section 4 shows that the solutions found by Eq. 6 produce accuracy higher than those evaluated on previous approaches.

3.3 Enhancing cross-pair syntactic similarity

As the computational cost of the similarity measure depends on the number of the possible set of correspondences C and this depends on the size of the anchor sets, we reduce the number of *placeholders* used to represent the anchors. The idea is the following. If a set of placeholders is contained in both a chunk of the hypothesis and a chunk of the text we replace it with a single label. Moreover, placeholders are propagated in the nodes of the syntactic trees following the head of the constituents (see Fig. 1). This enriches the trees with explicit dependecies between anchors. Finally, since, in general, texts implying a hypothesis contain more information than what is actually needed to support the entailment, we also reduce the syntactic trees of the texts. Here, we keep only that part of the tree playing an active role in supporting the entailment with respect to the hypothesis.

4 Experimental investigation

The aim of the experiments is twofold: (1) showing that entailments can be learned from examples and that our kernel functions over syntactic structures are effective to derive syntactic properties; (2) choosing the systems to be submitted to the challenge. The above goals can be achieved by experimenting the different intra and cross pair similarity measures.

4.1 Experimental settings

For the experiments, we used the RTE Challenge data sets, which we name as follows:
- $D1$, $T1$ and $D2$, $T2$ are, respectively, the development and the test sets of the first and second RTE challenges.
- ALL is the union of $D1$, $D2$, and $T1$, which we also split in 70%/30% (training/testing). This set is useful to test if we can learn entailment from the data prepared in the two different challenges.
- $D2(50\%)'$ and $D2(50\%)''$ is a random split of $D2$. As the data sets of the two competitions may

Experiment Settings								
$w = w' \vee l_w = l_{w'} \wedge c_w = c_{w'}$	√	√	√	√	√	√	√	√
$c_w = c_{w'} \wedge d(l_w, l_{w'}) > 0.2$			√	√	√	√	√	√
$((l_w, c_w), (l_{w'}, c_{w'})) \in Der$					√	√	√	√
$((l_w, c_w), (l_{w'}, c_{w'})) \in Ent$					√	√	√	√
$lev(w, w') = 1$						√	√	√
idf		√		√	√	√	√	√
Only Synt Trees							√	
Synt Trees with placeholders								√
Datasets								
Train:$D1$ Test:$T1$	0.5388	0.5813	0.5500	0.5788	0.5900	0.5888	0.6213	**0.6300**
Train:$T1$ Test:$D1$	0.5714	0.5538	0.5767	0.5450	0.5591	0.5644	0.5732	**0.5838**
Train:$D2(50\%)'$ Test:$D2(50\%)''$	0.6034	0.5961	0.6083	0.6010	0.6083	0.6083	0.6156	**0.6350**
Train:$D2(50\%)''$ Test:$D2(50\%)'$	0.6452	0.6375	0.6427	0.6350	0.6324	0.6272	0.5861	**0.6607**
Train:$D2$ Test:$T2$	0.6000	0.5950	0.6025	0.6050	0.6050	0.6038	0.6238	**0.6388**
Mean	0.5918 (± 0.0396)	0.5927 (± 0.0303)	0.5960 (± 0.0349)	0.5930 (± 0.0335)	0.5990 (± 0.0270)	0.5985 (± 0.0235)	0.6040 (± 0.0229)	**0.6297** (± 0.0282)
Train:$ALL(70\%)$ Test:$ALL(30\%)$	0.5902	0.6024	0.6009	-	0.6131	0.6193	0.6086	**0.6376**
Train:ALL Test:$T2$	0.5863	0.5975	0.5975	0.6038	-	-	0.6213	**0.6250**

Table 1: Experimental results of the different methods over different test settings

differ we created this *homogeneous* split.

We also used the following resources:
- The Charniak parser (Charniak, 2000) and the `morpha` lemmatiser (Minnen et al., 2001) to carry out the syntactic and morphological analysis.
- WordNet 2.0 (Miller, 1995) to extract both the verbs in entailment, Ent set, and the derivationally related words, Der set.
- The `wn::similarity` package (Pedersen et al., 2004) to compute the J&C distance (Jiang and Conrath, 1997) used as $d(l_w, l_{w'})$.
- A selected portion of the British National Corpus[1] to compute the idf. We assigned the maximum idf to unknown words.
- SVM-light-TK[2] (Moschitti, 2004) which encodes the basic tree kernel function, K_T, in SVM-light (Joachims, 1999). We used such software to implement K_s (Eq. 6), K_1, K_2 (Eq. 5) and $K_s + K_i$ kernels. The latter combines traditional approaches ($i \in \{1, 2\}$) with our new kernel.

4.2 Results and analysis

Table 1 reports the results of different similarity kernels on the different training and test split described in the previous section. The table is organized as follows:

The first 5 rows (*Experiment settings*) reports the intra-pair similarity measures defined in Section 3.1, the 6th row refers to only the idf similarity metric whereas the following two rows report the cross-pair similarity carried out with Eq. 6 with (*Synt Trees with placeholders*) and without (*Only Synt Trees*) augmenting the trees with placeholders, respectively. Each column in the *Experiment settings* indicates a different intra-pair similarity measure build by means of a combination of basic similarity approaches. These are specified with the check sign √. For example, Column 5 refers to a model using: the surface word form similarity, the $d(l_w, l_{w'})$ similarity and the idf.

The next 5 rows show the accuracy on the data sets and splits used for the experiments and the next row reports the average and Std. Dev. over the previous 5 results. Finally, the last two rows report the accuracy on ALL dataset split in 70%/30% and on the whole ALL dataset used for training and T2 for testing.

From the table we note the following aspects:

First, the lexical-based distance kernels K_1 and K_2 (Eq. 5) show accuracy significantly higher than the random baseline, i.e. 50%. In all the datasets (except for the first one), the $sim_w(T, H)$ similarity, based on the lexical overlap (first column), provides an accuracy essentially similar to the best lexical-based distance method. The dataset "Train:$D1$-Test:$T1$" shows that the accuracy reported for the best systems, i.e. 58.6% (Glickman et al., 2005; Bayer et al., 2005), is not significantly far from the result obtained with K_1 that uses the idf.

Second, our approach (last column) is significantly better than all the other methods as it provides the best result for each combination of training and test sets. On the "Train:$D1$-Test:$T1$" testbed, it overcomes the accuracy of the current state-of-the-art models (Glickman et al., 2005; Bayer et al., 2005) of about 4.4 absolute percent points (63% vs. 58.6%) and 4% over our best lexical similarity measure. By comparing the average on all datasets, our system improves of at least 3 absolute percent points all the methods.

Finally, the accuracy produced by *Synt Trees with placeholders* is quite higher than the one ob-

[1] http://www.natcorp.ox.ac.uk/

[2] SVM-light-TK is available at `http://ai-nlp.info.uniroma2.it/moschitti/`

tained with *Only Synt Trees*. Thus, the use of placeholders is fundamental to automatically learn entailments from examples.

We submitted two systems derived using two different learning sets: $D2$ and ALL. As expected, the accuracy (63.88%) obtained with the first system (trained on $D2$) is higher than the one (62.50%) obtained with the second system (trained on ALL). Homogeneity between training and testing is fairly relevant.

4.2.1 Qualitative analysis

Hereafter we show some instances selected from the first experiment "Train:$T1$-Test:$D1$". They were correctly classified by our overall model (last column) and miss-classified by the models in the seventh and in the eighth columns. The first is an example in entailment:

$T \Rightarrow H$	(id: 35)
T	*"Saudi Arabia, the biggest oil producer in the world, was once a supporter of Osama bin Laden and his associates who led attacks against the United States."*
H	*"Saudi Arabia is the world's biggest oil exporter."*

It was correctly classified by probably exploiting examples like these two:

$T \Rightarrow H$	(id: 929)
T	*"Ron Gainsford, chief executive of the TSI, said: ..."*
H	*"Ron Gainsford is the chief executive of the TSI."*

$T \Rightarrow H$	(id: 976)
T	*"Harvey Weinstein, the co-chairman of Miramax, who was instrumental in popularizing both independent and foreign films with broad audiences, agrees."*
H	*"Harvey Weinstein is the co-chairman of Miramax."*

A more interesting rule links the following two sentences which are not in entailment:

$T \not\Rightarrow H$	(id: 2045)
T	*"Mrs. Lane, who has been a Director since 1989, is Special Assistant to the Board of Trustees and to the President of Stanford University."*
H	*"Mrs. Lane is the president of Stanford University."*

It was correctly classified using instances like:

$T \not\Rightarrow H$	(id: 2044)
T	*"Jacqueline B. Wender is Assistant to the President of Stanford University."*
H	*"Jacqueline B. Wender is the President of Stanford University."*

$T \not\Rightarrow H$	(id: 2069)
T	*"Grieving father Christopher Yavelow hopes to deliver one million letters to the queen of Holland to bring his children home."*
H	*"Christopher Yavelow is the queen of Holland."*

Here, the implicit rule is: *"X (VP (V ...) (NP (to Y) ...)"* does not implies *"X is Y"*.

References

Samuel Bayer, John Burger, Lisa Ferro, John Henderson, and Alexander Yeh. 2005. MITRE's Submissions to the EU Pascal RTE Challenge. In *Proceedings of the 1st Pascal Challenge Workshop*, Southampton, UK.

S. Boughorbel, J-P. Tarel, and F. Fleuret. 2004. Non-mercer kernel for svm object recognition. In *Proceedings of BMVC 2004*, pages 137–146.

Eugene Charniak. 2000. A maximum-entropy-inspired parser. In *Proc. of the 1st NAACL*, pages 132–139, Seattle, Washington.

Michael Collins and Nigel Duffy. 2002. New ranking algorithms for parsing and tagging: Kernels over discrete structures, and the voted perceptron. In *Proceedings of ACL02*.

Courtney Corley and Rada Mihalcea. 2005. Measuring the semantic similarity of texts. In *Proc. of the ACL Workshop on Empirical Modeling of Semantic Equivalence and Entailment*, Ann Arbor, Michigan, June.

Oren Glickman, Ido Dagan, and Moshe Koppel. 2005. Web based probabilistic textual entailment. In *Proceedings of the 1st Pascal Challenge Workshop*, Southampton, UK.

Bernard Haasdonk. 2005. Feature space interpretation of SVMs with indefinite kernels. *IEEE Trans Pattern Anal Mach Intell*, 27(4):482–92, Apr.

Jay J. Jiang and David W. Conrath. 1997. Semantic similarity based on corpus statistics and lexical taxonomy. In *Proc. of the 10th ROCLING*, pages 132–139, Tapei, Taiwan.

Thorsten Joachims. 1999. Making large-scale svm learning practical. In B. Schlkopf, C. Burges, and A. Smola, editors, *Advances in Kernel Methods-Support Vector Learning*. MIT Press.

George A. Miller. 1995. WordNet: A lexical database for English. *Communications of the ACM*, 38(11):39–41, November.

Guido Minnen, John Carroll, and Darren Pearce. 2001. Applied morphological processing of english. *Natural Language Engineering*, 7(3):207–223.

Alessandro Moschitti. 2004. A study on convolution kernels for shallow semantic parsing. In *proceedings of the ACL*, Barcelona, Spain.

Ted Pedersen, Siddharth Patwardhan, and Jason Michelizzi. 2004. Wordnet::similarity - measuring the relatedness of concepts. In *Proc. of 5th NAACL*, Boston, MA.

Vladimir Vapnik. 1995. *The Nature of Statistical Learning Theory*. Springer.

Annie Zaenen, Lauri Karttunen, and Richard Crouch. 2005. Local textual inference: Can it be defined or circumscribed? In *Proc. of the ACL Workshop on Empirical Modeling of Semantic Equivalence and Entailment*, Ann Arbor, Michigan, June.