Determining *is-a* relationships for Textual Entailment

Vlad Niculae
Université de Franche-Comté, Besançon
Center for Computational Linguistics
University of Bucharest
vlad@vene.ro

Octavian Popescu
Fondazione Bruno Kessler
popescu@fbk.eu

Abstract

The Textual Entailment task has become influential in NLP and many researchers have become interested in applying it to other tasks. However, the two major issues emerging from this body of work are the fact that NLP applications need systems that (1) attain results which are not corpus dependent and (2) assume that the text for entailment cannot be incorrect or even contradictory. In this paper we propose a system which decomposes the text into chunks via a shallow text analysis, and determines the entailment relationship by matching the information contained in the $is - a$ pattern. The results show that the method is able to cope with the two requirements above.

1 Introduction

Given a pair of two text fragments, $\mathcal{T}$ and $\mathcal{H}$, the textual entailment task consists in deciding whether the information in $\mathcal{H}$ is entailed by the information in $\mathcal{T}$ (Dagan et al., 2006). Many and diverse systems participated in Recognizing Textual Entailment Challenges (RTE), which helped in pointing out interesting issues with an important impact in other NLP tasks. Under some assumptions, the papers published on this topic have proven that the TE methodology is useful for machine translation, text summarization, information retrieval, question answering, fact checking etc. (Padó et al., 2009; Lloret et al., 2008; Clinchant et al., 2006; Harabagiu and Hickl, 2006).

The two major issues emerging from this body of work are the fact that NLP applications need systems that (1) attain results which are not corpus dependent and (2) assume that the text for entailment may be incorrect or even contradictory. In this paper we propose a system which decomposes the text into chunks via a shallow text analysis, and determines the entailment relationship by matching the information contained in the $is - a$ pattern. The results show that the method is able to cope with the two requirements above.

Our system produces stable results on the RTE corpora and is little affected by the presupposed veridical value of the information in $\mathcal{T}$ and/or $\mathcal{H}$ and, therefore, it is instrumental in addressing the above enumerated issues. We focused on the pairs on which $\mathcal{H}$ has the form of an $is - a$ relation between an entity and a property. The method makes use of shallow text analysis, extracts the information contained in each chunk, and tries to find a match for the entity in $\mathcal{H}$ on the list of entities of $\mathcal{T}$. If the match is successful then the properties of the entities are compared in order to decide on the entailment.

In general, the information allowing the match is not found in a single chunk. The property of an entity expressed by the $is - a$ relation found in $\mathcal{H}$ may be not directly expressed in $\mathcal{T}$, the property and the entity being in separated chunks. The system resolves the coreference between the entities mentioned in each chunk by employing mostly techniques for inter-document coreference (Popescu et al., 2008; Ponzetto and Poesio, 2009).

To unify the information contained in each chunk we considered a set of heuristics which identifies syntactical fixed forms and expresses them as $is - a$ relations. For example, an apposition becomes a copula. We also recognized relations between entities which are typically expressed as a pattern, for example [[*e1 is known as e2*]], following the work of (Hearst, 1992; Pantel et al., 2004). The basic approach is extended by considering also synonyms/antonyms and negation mismatches. For comparison purposes we considered a set of features which extend the RTE feature set (MacCartney et al., 2006) and syntactic kernels (Moschitti, 2006) with SVM. The results we obtain support the statement that integration of syntactic and semantic information can yield better results over surface based features (Padó et al., 2009).

For a better understanding of the variance of the results according to the corpora, including robustness to noise and dependency of the veridical presupposition on the information in corpus, we used a technique of generating a scrambled corpus similar to the one described in (Yuret et al., 2010). The results we obtain confirm that the method is stable and overcome with a large margin other approaches. Unlike the methods based on logical forms and world knowledge, which many times are less efficient on noisy corpora, the proposed method maintains a shallow syntactic and semantic level while relevant information unification process takes part, a process which is mostly ignored by surface approaches.

The remainder of this paper is organized as follows: in Section 2 we review the relevant literature, in Section 3 we present the details of the methodology we employ and in Section 4 we present and discuss the experiments we carried on the RTE3, 4, and 5 corpora. The paper ends with the conclusions and further work section.

2 Relevant literature

Successful systems for recognizing textual entailment are usually complex and multi-tiered. The Stanford RTE system (MacCartney et al., 2006), for instance, has a linguistic analysis stage, an alignment stage and an entailment determination stage. The alignment stage, similar to (Haghighi et al., 2005), is based on dependency graph matching. The decision stage can be hand-tuned or learned, but the system did not perform significantly different in the two cases. In the RTE-5 competition, the best systems reach precisions up to 70% using rule-based methods (Iftene and Moruz, 2009) and distance-based approaches. Many systems are based on machine learning classifiers with lexical similarities (Castillo, 2010), non-symmetric causal metrics (Gaona et al., 2010) and syntactic features (Zanzotto et al., 2009). They attain competitive accuracy scores, but there is no report of precision.

3 Methodology

In this section we describe the main components of the strategy of finding a match between the information in $\mathcal{H}$ and $\mathcal{T}$. Usually the relevant information in $\mathcal{T}$ is not in a single chunk and it does not have a form directly comparable with the information in $\mathcal{H}$. Let us see an example:

$\mathcal{T}$: *Pop star Madonna has suffered "minor injuries" and bruises after falling from a startled horse on New York's Long Island on Saturday. According to her spokeswoman, the 50-year-old singer fell when her horse ...*

$\mathcal{H}$: *Madonna is 50 years old.*

The information in $\mathcal{H}$ assigns to the entity *Madonna* a certain attribute. In order to match this information in $\mathcal{T}$ we have to find the same entity and all its mentions and join the attributes of each mentions together in order to see if the attribute occurring in $\mathcal{H}$ is within all these. The general strategy of resolving the entailment is:

1. Match the [[X **be** α]] pattern in $\mathcal{H}$
2. Identify all entities $X_1, ..., X_n$ in $\mathcal{T}$
3. corefer X_i and join the attributes α_i in X_e and α_e
4. match X against each X_e and check the attribute α_e.

We use a parser to obtain the heads of all NPs. Most of the dependency parsers normalize the syntactic variant like passive, apposition, time expressions (De Marneffe et al., 2006; Meyers et al., 2009). Each head represents a possible entity and we extract as attributes all the heads of adjectival and nominal phrases which are under the respective head. For example in Figure 1, the entity *Bob Iger* has the attribute *CEO of Disney* in both cases. Notice that the dependency structures are very different and a direct comparison is likely to be of little help.

The coreference of heads is carried out using a local coreference engine based on multi-pass sieve coreference resolution (MacCartney et al., 2006). For attribute matching we also considered synonyms (Roget, 1911). For example, the system catches correctly the entailment relation in the example below:

$\mathcal{T}$: *The home at 7244 S. Prairie Ave. once owned by mobster Al Capone and his family has hit the market for $450,000.*

$\mathcal{H}$: *Al Capone was a ganster.*

because *gangster* and *mobster* are synonyms.

# +/-	RTE-3′ 104 / 91				RTE-4′ 90 / 102				RTE-5′ 134 / 131			
	A	P	R	F	A	P	R	F	A	P	R	F
BL1	.69	.71	.69	.70	.51	.49	.35	.40	.57	.56	.84	.67
BL2	.74	.71	.87	.78	.62	.62	.53	.57	.59	.59	.82	.68
BL3	.56	.61	.45	.52	.57	.53	.53	.53	.59	.59	.65	.61
NB	.57	.96	.21	.35	.62	.90	.21	.34	.44	.96	.18	.30
NA	.56	.92	.21	.34	.62	.88	.23	.37	.45	.96	.20	.34
SB	.56	.88	.22	.35	.62	.82	.26	.39	.44	.82	.21	.34
SA	.57	.86	.24	.38	.64	.81	.29	.43	.45	.78	.26	.39

Table 1: Results on RTE′ corpora.

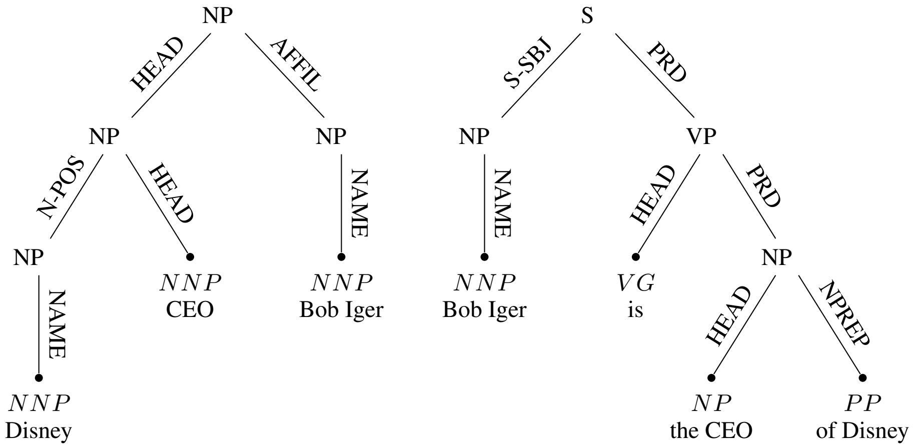

Figure 1: [[X is α]] pattern extraction.

An important improvement of the performances is obtained if before a transformation to $is - a$ relation is carried out for some fixed, strictly defined patterns. A very common pattern involving *as* phrases is:

$\mathcal{T}$: *During Reinsdorf's 24 seasons as chairman of the White Sox, the team ....*

$\mathcal{H}$: *Reinsdorf was the chairman of the White Sox for 24 seasons*

The pattern *[Person] as* α is equivalent with *[Person]* $is - a$ α. The following patterns are prototypical *as* usage as copula alternative: [[NP known as α]], [[NP served as α]], [[NP formed as α]], and [[NP work as α]].

Another common pattern is used for part of a whole or location: [[NP found in α]], [[NP located in α]], and [[NP in α]].

An example instantiation of such a pattern is:

$\mathcal{T}$: *The Gaspe is a North American peninsula (...) in Quebec.*

$\mathcal{H}$: *The Gaspe Peninsula is located in Quebec*

While the main strategy remains the same, using the transformation of these types of patterns increases the recall of the system significantly.

4 Experiments

We based our experiments on the freely available corpora from the Recognizing Textual Entailment competitions RTE-3, 4 and 5. All of the entailment pairs were parsed with the BLLIP parser (Charniak and Johnson, 2005) and subsequently processed with GLARF (Meyers et al., 2009). The copula pattern [[X **be** α]] was matched in all hypotheses, and only instances where the match was positive were kept, see Table 2. The method presented in the previous section does not require training. However, in order to have a direct comparison with other methods, we report only the results obtained on the gold corpus.

We employed three progressively complex baselines:

- **BL1**: Lexical overlap baseline with threshold determined by a linear SVM (Mehdad and Magnini, 2009)

	RTE3	*RTE4*	*RTE5*
copula gold	202	102	269
copula dev	204	101	246

Table 2: RTE corpora, only copula examples

- **BL2**: Linear SVM, features: number of common words, number of words exclusively in $\mathcal{H}$, number of common named entities, number of named entities exclusive to $\mathcal{H}$, number of negative words in $\mathcal{T}$ and respectively $\mathcal{H}$, and number of common parse subtrees
- **BL3**: Tree kernel SVM (Moschitti, 2006), each pair being encoded as the set of common parse subtrees between $\mathcal{T}$ and $\mathcal{H}$.

BL1 and **BL2** were trained using the *scikit-learn* machine learning library version 0.12 (Pedregosa et al., 2011), with the feature extraction from NLTK (Bird et al., 2009). **BL3** was trained using *svmlight-tk* (Moschitti, 2006; Joachims, 1999). In the case of RTE-3$'$ and RTE-5$'$ the provided train-test split was used, whereas for RTE-4$'$ we made a 50-50 split. The regularization parameters and the tree kernel parameters were optimized using grid-search with cross validation.

Four configurations of our system were evaluated, and were labelled with two-letter names. The first letter signifies whether synonym matching is used (**S**) or not (**N**). The second letter marks whether matching is performed at word boundaries (**B**) or anywhere (**A**).

Hypothesis scrambling. A cursory look at Table 1 shows that the baseline approaches vary significantly from corpus to corpus, while the attribute extraction is relatively invariable. Also, apparently, the BL3 using a tree kernel does not perform as well as BL1 or BL2. The difference may come from the typology of entailment pairs in RTE corpora. It seems that matching one entity from $\mathcal{H}$ with one entity from $\mathcal{T}$ is correlated with the entailment. However, this is not the case in general. On the one hand, this observation suggests that on a corpus with a lower degree of correlation, the results may be different. On the other hand, many NLP applications must make decisions when the relationship between $\mathcal{T}$ and $\mathcal{H}$ is more ambiguous than in RTE corpora. That is why we decided to apply the scrambling technique on RTE corpora for evaluation (Yuret et al., 2010).

	RTE3	RTE4	RTE5
#	673	315	418
BL1	.50	.75	.20
BL2	.15	.32	.18
BL3	.65	.60	.54
NB	.90	.95	.95
NA	.90	.94	.94
SB	.84	.91	.87
SA	.79	.85	.80

Table 3: Results on scrambled corpora

For each pair in an entailment relationship we replaced the name of the entities in $\mathcal{H}$ with entities from $\mathcal{T}$. For example the sentence *Bob Iger is Disney CEO* which originally was in entailmentship with *The puzzlement comes from video players who don't work at NBC, Fox or Hulu, and who can't see the upside in Disney CEO Bob Iger throwing in his lot with Hulu* was replaced with *Fox is Disney CEO*, *Hulu is Disney CEO*. On the corpus obtained in this way we run all thesystems obtaining the results in Table 3.

In absolute values, the performance of attribute extraction systems does not change too much, but the baseline systems have registered a serious drop in accuracy. Also the BL3 system was much better than the other baselines. This shows that the use of structural information pays off.

5 Conclusion and further research

In this paper we introduced a system for TE which identifies the entities in both $\mathcal{T}$ and $\mathcal{H}$ and determines the attributes which may match in order to infer the entailment relationship. The system uses a shallow text analysis. While the precision of this type of approach is very high, the experiments show that without the help of modules that cope with grammatical variance and synonymy correspondence, the recall remains very low. However, the method is stable and the scrambling experiment suggests that the presented approach is competitive for applications requiring unbiased results on heterogeneous corpora.

We think that pattern matching is a good solution to increase the recall. The mapping from fixed syntactic structures to an $is-a$ relation seems possible.

References

Steven Bird, Ewan Klein, and Edward Loper. 2009. *Natural language processing with Python*. O'Reilly Media.

Julio J Castillo. 2010. Using machine translation systems to expand a corpus in textual entailment. In *Advances in Natural Language Processing*, pages 97–102. Springer.

Eugene Charniak and Mark Johnson. 2005. Coarse-to-fine n-best parsing and maxent discriminative reranking. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pages 173–180. Association for Computational Linguistics.

Stéphane Clinchant, Cyril Goutte, and Eric Gaussier. 2006. Lexical entailment for information retrieval. In *Advances in Information Retrieval*, pages 217–228. Springer.

Ido Dagan, Oren Glickman, and Bernardo Magnini. 2006. The pascal recognising textual entailment challenge. In *Machine Learning Challenges. Evaluating Predictive Uncertainty, Visual Object Classification, and Recognising Tectual Entailment*, pages 177–190. Springer.

Marie-Catherine De Marneffe, Bill MacCartney, and Christopher D Manning. 2006. Generating typed dependency parses from phrase structure parses. In *Proceedings of LREC*, volume 6, pages 449–454.

Miguel Angel Ríos Gaona, Alexander Gelbukh, and Sivaji Bandyopadhyay. 2010. Recognizing textual entailment using a machine learning approach. In *Advances in Soft Computing*, pages 177–185. Springer.

Aria D Haghighi, Andrew Y Ng, and Christopher D Manning. 2005. Robust textual inference via graph matching. In *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, pages 387–394. Association for Computational Linguistics.

Sanda Harabagiu and Andrew Hickl. 2006. Methods for using textual entailment in open-domain question answering. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, pages 905–912. Association for Computational Linguistics.

Marti A Hearst. 1992. Automatic acquisition of hyponyms from large text corpora. In *Proceedings of the 14th conference on Computational linguistics-Volume 2*, pages 539–545. Association for Computational Linguistics.

Adrian Iftene and Mihai-Alex Moruz. 2009. UAIC participation at RTE5. *Proceedings of TAC, Gaithersburg, Maryland*.

Thorsten Joachims. 1999. Advances in kernel methods. chapter Making large-scale support vector machine learning practical, pages 169–184. MIT Press, Cambridge, MA, USA.

Elena Lloret, Óscar Ferrández, Rafael Muñoz, and Manuel Palomar. 2008. A text summarization approach under the influence of textual entailment. In Bernadette Sharp and Michael Zock, editors, *NLPCS*, pages 22–31. INSTICC Press.

Bill MacCartney, Trond Grenager, Marie-Catherine de Marneffe, Daniel Cer, and Christopher D Manning. 2006. Learning to recognize features of valid textual entailments. In *Proceedings of the main conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics*, pages 41–48. Association for Computational Linguistics.

Yashar Mehdad and Bernardo Magnini. 2009. A word overlap baseline for the recognizing textual entailment task.

Adam Meyers, Michiko Kosaka, Nianwen Xue, Heng Ji, Ang Sun, Shasha Liao, and Wei Xu. 2009. Automatic recognition of logical relations for english, chinese and japanese in the glarf framework. In *Proceedings of the Workshop on Semantic Evaluations: Recent Achievements and Future Directions*, pages 146–154. Association for Computational Linguistics.

Alessandro Moschitti. 2006. Making tree kernels practical for natural language learning. In *Proceedings of EACL*, volume 6, pages 113–120.

Sebastian Padó, Michel Galley, Dan Jurafsky, and Chris Manning. 2009. Robust machine translation evaluation with entailment features. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 1-Volume 1*, pages 297–305. Association for Computational Linguistics.

Patrick Pantel, Deepak Ravichandran, and Eduard Hovy. 2004. Towards terascale knowledge acquisition. In *Proceedings of the 20th international conference on Computational Linguistics*, page 771. Association for Computational Linguistics.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Simone Paolo Ponzetto and Massimo Poesio. 2009. State-of-the-art nlp approaches to coreference resolution: theory and practical recipes. In *Tutorial Abstracts of ACL-IJCNLP 2009*, pages 6–6. Association for Computational Linguistics.

Octavian Popescu, Christian Girardi, Emanuele Pianta, and Bernardo Magnini. 2008. Improving cross document coreference. In *Proceedings of JADT*.

Peter Mark Roget. 1911. *Roget's Thesaurus of English Words and Phrases...* TY Crowell Company.

Deniz Yuret, Aydin Han, and Zehra Turgut. 2010. Semeval-2010 task 12: Parser evaluation using textual entailments. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, pages 51–56. Association for Computational Linguistics.

Fabio Zanzotto, Marco Pennacchiotti, and Alessandro Moschitti. 2009. A machine learning approach to textual entailment recognition. *Natural Language Engineering*, 15(04):551–582.