Combining Specialized Entailment Engines

Bernardo Magnini[1] and Elena Cabrio[1] [2]

[1] FBK- Irst, Trento (Italy), [2] University of Trento (Italy)
{magnini, cabrio}@fbk.eu

Abstract

In this paper we propose a general method for the combination of specialized textual entailment engines. Each engine is supposed to address a specific language phenomenon, which is considered relevant for drawing semantic inferences. The model is based on the idea that the distance between the Text and the Hypothesis can be conveniently decomposed into a combination of distances estimated by single and disjoint engines over distinct linguistic phenomena. We provide both the formal definition of the model and preliminary empirical evidences supporting the underlying intuition.

1. Introduction

Textual entailment (TE)[1] (Dagan *et al.* , 2004) comes at various levels of complexity, involving almost all linguistic phenomena of natural languages, including lexical, syntactic and semantic variations. While most of the research so far in TE has proposed omnicomprehensive approaches, we think that crucial progress may derive from a focus on single linguistic phenomena involved in TE and on their combination.

Being able to decompose the complexity of the TE problem, we expect both the possibility of evaluating performances and progress on single aspects of TE, and the possibility to train specialized engines on single linguistic phenomena.

According to this intuition, we attempt to define a general method for specialized entailment engines, each of which able to deal with a certain aspect of language variability. We introduce transformation-based TE engines within a tree edit distance approach, and provide precise definitions for their behaviour. We argue that disjointness is a crucial property of specialized engines and we define it basing on the idea of monothematic datasets. Finally, we discuss two main issues underlying their combination: the order of application and the combination of individual results in order to produce a global result.

2. Distance-Based Textual Entailment

We assume a distance-based framework, where the distance d between a Text T and an Hypothesis H is inversely proportional to the entailment relation in the pair. Although there can be several ways to calculate the distance d, in this paper we assume an edit distance approach (Kouylekov and Magnini, 2005), where d is estimated as the sum of the costs of the edit operations (i.e. insertion, deletion, substitution), which are necessary to transform T into H.

We use tree edit distance over the dependency trees of T and H, and we use an implementation of the (Zhang and

Shasha , 1990) algorithm, which have been shown to be optimal (i.e., the algorithm returns the less costly sequence of edit operations that transform T into H), and whose computational complexity is $O(n^4)$, where n is the sum of the sizes of two trees T and H.

We also assume "three-way" judgements (Giampiccolo *et al.* , 2008), where, given a (T, H) pair, the system can output YES, if there is a degree of entailment between T and H, NO, when some degree of contradiction is found, and UNKNOWN, in case neither entailment or contradiction is detected between T and H.

3. Specialized Entailment Engines

The hypothesis of this paper is that the distance $ED(T, H)$ can be profitably decomposed as the combination of the distances related to the different linguistic phenomena involved in the entailment relation between T and H. In other words, given $ED_i(T, H)$, the edit distance between T and H for a certain linguistic aspect i, we assume that:

$$ED(T, H) = COMB_{i=1}^n[ED_i(T, H)] \qquad (1)$$

where i potentially ranges over all the linguistic phenomena involved in textual entailment.

3.1. General definition

A specialized distance-based entailment engine ED_i for a certain linguistic phenomenon i determines a distance d between T and H, such that:

$$ED_i(T, H) = \begin{cases} 0 & \text{if } i \text{ does not affect T and H} \\ 0 < d \le t_i & \text{if } i \text{ contributes to the} \\ & \text{entailment in T and H} \\ > t_i & \text{if } i \text{ contributes to the} \\ & \text{contradiction in T and H} \end{cases}$$

where t_i is a threshold that separates the entailment and the contradiction due to the phenomenon i.

As an example, let's suppose a specialized entailment engine ED_{a-p} which only detects entailment due to the active-passive alternation between T and H, and suppose the following T-H pairs:

T1   John paints the wall.
$H1$ The wall is white.
$H2$ The wall is painted by John.
$H3$ The wall is painted by Bob.
$H4$ The wall is coloured by John.

When ED_{a-p} is applied to the examples, according to our definition, we will obtain the following results:

$ED_{a-p}(T1, H1) = 0$
because there is no active-passive alternation in the pair;

$ED_{a-p}(T1, H2) = 0 < d \leq t_i$
because the application of an active-passive rule allows to preserve the entailment between T1 and H2;

$ED_{a-p}(T1, H3) \geq t_i$
because, although an active-passive alternation is present in the pair, the corresponding rule can not be applied, this way contributing to the contradiction in the pair.

More generally, we distinguish three cases in the behaviour of a specialized entailment engine ED_i:

the neutral case,   when the linguistic phenomenon i does not occur in a certain pair. We say that the TE engine ED_i is "neutral" with respect to i, when it can not produce any evidence either for the entailment or the contradiction between T and H; the distance d is conventionally set to 0, according to the intuition that no knowledge of i (i.e., no effort) is applied. This case corresponds to the *Unknown* situation of the RTE evaluation, although, in the context of specialized engines, it has to be interpreted more precisely, as the absence of a certain linguistic phenomenon;

the positive case,   when the phenomenon i occurs and contributes to establish an entailment relation between T and H. The distance d is higher than 0, because we penalize the application of some specific knowledge about i, and below a threshold t_i, which separates the negative cases. We consider *equality*, i.e. when T and H are made of the same sequence of tokens, as a special case of the positive situation. In this case we set the distance to a constant ε, kept close to 0, according to the intuition that the entailment relation holds, and that a minimal effort is required to detect equality;

the negative case,   when the phenomenon i occurs and contributes to establish a contradiction relation between T and H. Negative cases may correspond to two situations: (i) explicit knowledge about contradiction (e.g. antonyms, negation) or (ii) a mismatch situation, where it is not possible to apply an entailment rule, and as a consequence, a certain degree of contradiction emerges from the T-H pair (see the T1-H3 pair on active-passive alternation). When ED_i is negative, the distance d is higher than the threshold t_i, meaning a high cost of transforming T into H due to phenomenon i.

In addition to the distance d, a specialized engine returns the set of transformations between T and H allowed by the application of specific knowledge of phenomenon i.

More precisely, an engine ED_i returns the transformations $TR_i = \{T_1 \rightarrow H_1', T_1 \rightarrow H_n'\}$ which allow to transform T into H' by virtue of phenomenon i. For instance, the application of ED_{a-p} to the example T1-H4 will produce H'="The wall is painted by John", because of the application of the active-passive transformation to T1.

3.2.   Monothematic datasets and disjoint engines

We say that a certain TE engine ED_i is *neutral* (notated with the symbol $\sim$) with respect to a certain RTE dataset $[T, H]$, if the sum of the distances for each T-H pair of the dataset is 0, meaning that the engine ED_i is neutral for all the pairs in $[T, H]$ (2).

$ED_i \sim$ on $[T, H]$
IF $\sum_{j=1}^{n} ED_i(T, H)_j = 0$ with $(T, H)_j \in [T, H]$ (2)

Conversely, we say that an engine ED_i is *non-neutral* (notated with the symbol $\nsim$) for a certain dataset $[T, H]$ when, for each pair in $[T, H]$, the application of ED_i produces a distance different by 0 (3).

$ED_i \nsim$ on $[T, H]$
IF $\forall$(T,H)$_j$ in $[T, H]$ $ED_i(T, H)_j \neq 0$ (3)

In addition to define specialized TE engines, it is useful to use specialized Text-Hypothesis datasets for a certain linguistic phenomenon i (notated as $[T, H]_i$), made of *monothematic* Text-Hypothesis pairs which, according to human judgements can be resolved by means of a single linguistic phenomenon i. Here "resolved" means that a human can identify one of the three cases mentioned in the previous Section, i.e. neutrality, positivity, negativity. As an example, the pair T1-H2 can be resolved (i.e. judged as positive) by means of a syntactic transformation based on active-passive alternation in English: there is no other knowledge involved in the entailment, and we say that this is a monothematic pair with respect to the active-passive phenomenon. On the contrary, pair T1-H4, in order to be resolved, requires that both an active-passive alternation and a lexical similarity rule (between "paint" and "colour") are applied, and for this reason this is not a monothematic pair.

The use of monothematic datasets allows to define disjoint TE engines. Intuitively, we can say that two specialized engines ED_i and ED_j are disjoint one from the other (notated with the symbol $\varnothing$) on a monothematic dataset $[T, H]_k$, if both of them do not cover the linguistic phenomenon k of the dataset. In our terminology, this happens when the two engines are not non-neutral with respect to the dataset. According to this intuition, we can formulate the following *disjointness condition* (4):

$ED_i \varnothing ED_j$ on $[T, H]_k$

IF $\nexists \, ED_i \approx$ on $[T, H]_k$ AND $ED_j \approx$ on $[T, H]_k$ (4)

We notice that a special case of disjointness is when both the TE engines are neutral with respect to the dataset. The disjointness condition (4) can be easily extended to a set of n specialized engines, just checking that the condition holds for all the engines of the set. Moreover, once the disjointness has been tested on a certain monothematic dataset, in principle, it can be extended to an arbitrary number of monothematic datasets, covering most of the linguistic aspects involved in textual entailment. Finally, assuming that the disjointness among two entailment engines is tested on a large variety of monothematic datasets, we can reasonably induce that such disjointness is maintained over a non monothematic dataset (5):

$$ED_i \varnothing ED_j \text{ on } [T, H]_{a,b,...z} \cong ED_i \varnothing ED_j \text{ on } [T, H] \quad (5)$$

which opens the possibility to use specialized engines over general datasets, like those used in the RTE evaluations.

4. Combining disjoint engines

Given a set of disjoint TE engines, they can run separately on a generic dataset $[T, H]$. There are two issues that need to be addressed: (i) defining the order in which the engines are applied; (ii) defining how to combine their individual results in order to produce a global result.

4.1. Order of application

In general, the fact that two engines are disjoint does not guarantee that they are independent, which means that the order of their application does affect the final result. For instance, considering pair T1-H1, it seems difficult to apply the active-passive transformation before the lexical transformation between "paint" and "colour" has been applied. We assume a *cascade* of disjoint entailment engines, where each engine takes as input the output of the previous engine, defined as the set of edit transformations from T to H' related to phenomenon i (see Section 3.1.). As a first approximation, we first run the engines whose transformations apply to smaller portions of text. For instance, ED_{lex} comes first with respect to ED_{a-p}, because lexical rules apply to single tokens, while active-passive alternation applies to groups of tokens.

4.2. Combination of individual results

Since a single engine can output three results (entailment, notated with +, contradiction, notated with −, and neutrality, notated with =) for a set of n engines we have 3^n different combinations. The following examples show the nine combinations among two linguistic phenomena (L1 for active-passive alternation, L2 for lexical similarity), as well as their respective behaviors (judged by humans). For instance, pair T1-H1 is positive for both the engines (i.e.

both active-passive and lexical similarity contribute to the entailment relation), while pair T1-H4 is positive regarding lexical similarity, because "paint" and "colour" in this context are synonyms, and it is negative regarding active-passive, because the subject of T1 (i.e. John) is not the object of H4 (Bob).

$T1$	John paints the wall.			
$H1$	The wall is coloured by John.	$L1+$	$L2+$	Y
$H2$	The wall is jumped by John.	$L1+$	$L2=$	U
$H3$	The wall is cleaned by John.	$L1+$	$L2-$	C
$H4$	The wall is coloured by Bob.	$L1-$	$L2+$	C
$H5$	The wall is jumped by Bob.	$L1-$	$L2=$	U
$H6$	The wall is cleaned by Bob.	$L1-$	$L2-$	C
$H7$	John colours the wall.	$L1=$	$L2+$	Y
$H8$	John jumps the wall.	$L1=$	$L2=$	U
$H9$	John cleans the wall.	$L1=$	$L2-$	C

While there can be several ways to combine the distances estimated by specialized engines, in this Section we discuss two options: the sum of the distances, and a simple voting mechanism. In both cases we assume a set $E = \{ED_a, ED_b, ...ED_z\}$ of specialized and disjoint entailment engines over a generic dataset $[T, H]$. On each pair in $[T, H]$ each specialized engine ED_i contributes with 0 when the engine in neutral on the pair, with an integer between 0 and the threshold t_i when the engine is positive on the pair, and with an integer greater that t_i when the engine is negative on the pair.

Sum of Distances. For each engine ED_i we sum the distance on $[T, H]$ (6) and the final result is compared against a threshold t_{all} estimated over the training data of $[T, H]$ so that the sum of all thresholds of the specialized engines does not exceed the global threshold t_{all} (7).

$$ED[T, H] = \sum_{i=a}^{z} ED_i[T, H] \quad (6)$$

$$\sum_{i=a}^{z} t_i \leq t_{all} \quad (7)$$

The intuition behind this model is that each specialized engine contributes with a piece of the distance between T and H and that, because of the disjoint assumption, just adding the distances will provide a good estimation of the overall distance. We notice that, since the neutral judgements are 0, they do not affect in any way the sum. For instance, for the pair T1-H1, supposing a correct behaviour, engine L1 will output a value in between 0 and $t_{active-passive}$, while L2 will output a value in between 0 and t_{lex}. Then, we sum up $t_{active-passive}$ and t_{lex} and we expect a value in between 0 and t_{all}, which means entailment.

Voting. This combination mechanism selects the most voted result among those produced by the engine of the set E. A voting strategy which seems to fit well with entailment judgements is to set up some forms of weighting in order to resolve cases of parity of votes. We have hypothesized the following two precedence orders:

$$neutral < positive < negative$$
$$positive < neutral < negative$$

the latter meaning that, in case of parity of votes, negatives win over neutrals and positives, and neutrals win over positives. For instance, in case of pair T1-H9, where we have one neutral and one negative judgement, the final result would be contradiction.

5. Experiments and Results

In this Section we report on preliminary experiments showing that designing specialized and disjoint entailment engines is a profitable research direction.

5.1. Building monothematic datasets

In order to build a monothematic dataset we set up the following procedure. Given a $[T, H]$ pair and a number of linguistic phenomena $L = \{L_a, L_b, ...L_z\}$ we manually build a monothematic pair $[T, H1]_i$ for a linguistic aspect i such that T is exactly the same, and $H1$ is obtained applying to T the minimal textual transformations allowed by rules of phenomenon i in order to mach the portion of H interested by i. The intuition is that, for a system in order to resolve $[T, H]$ it would be necessary, among the other, the ability to resolve $[T, H1]_i$.

As an example, starting from the pair T1-H1, and focusing on active-passive alternation and lexical similarity, we can derive the following monothematic pairs.

$T1$	John paints the wall.			
$H1$	The wall is painted by John.	$L1+$	$L2 =$	Y

$T1$	John paints the wall.			
$H7$	John colours the wall.	$L1 =$	$L2+$	Y

5.2. Implementing specialized engines with the EDITS system

For our experiments we have used EDITS[2] (Negri et al., 2009), an entailment system based on the estimation of the edit distance between T and H. For each specialized engine ED_i we have defined a separate set of rules R_i covering the main aspects of phenomenon i. Rules may express either entailment (notated with r_i^+), or contradiction (notated with r_i^-). We have adopted the following general cost schemas for substitution, deletion and insertion:

$$SUBS_i(x,y) = \begin{cases} (1+\varepsilon) \text{ - } P(r_i^+) & \text{if } r_i^+ \text{ applies} \\ P(r_i^-) + 1 + 2\varepsilon & \text{if } r_i^- \text{ applies} \\ 0 & \text{if } r_i \text{ does not apply} \end{cases}$$

$$DEL_i(x,-) = \begin{cases} P(r_i) + 2\varepsilon & \text{if } r_i \text{ applies} \\ 0 & \text{if } r_i \text{ does not apply} \end{cases}$$

$$INS_i(-,y) = \begin{cases} P(r_i) + 2\varepsilon & \text{if } r_i \text{ applies} \\ 0 & \text{if } r_i \text{ does not apply} \end{cases}$$

where $P(r_i)$ is the probability that rule r_i preserves either the entailment or the contradiction relation, and ε is a constant defined close to 0 in order to avoid the positive cases to be equal to 0 (i.e. the neutral case).

According to the above schema, the cost γ for substitution ranges from ε to $(1+\varepsilon)$ for the positive case, from $(1+2\varepsilon)$ to $(2+2\varepsilon)$ for the negative case, and it is 0 for the neutral case, which fits with the general definition provided in Section 3.1. In addition, the three schemas satisfy the *triangle inequality* condition that guarantees optimality in edit distance algorithms (Zhang and Shasha , 1990), so that, given $P(r_i)$ constant for the three operations, we always have that:

$$DEL(x,-) + INS(-,y) \geq SUBS(x,y)$$

As an example of how a cost schema works, suppose that the following lexical rules are defined:

$r1_{lex}^+$: paint $\Rightarrow$ color (.8)
$r2_{lex}^-$: paint $\nRightarrow$ clean (.7)

The cost of substituting "paints" with "color" in T1-H7 would be:
$(1+\varepsilon)$ - $P(r_i) = (1+\varepsilon)$ - $0.8 = 0.15 + \varepsilon$
while the cost of substituting "paints" with "color" in T1-H9 would be:
$P(r_i) + 1 + 2\varepsilon = 0.7 + 1 + 2\varepsilon = 1.7 + 2\varepsilon$
A similar inference holds for active-passive rule:

$r1_{ap}^+$: $x \leftarrow$ paint $\rightarrow y \Rightarrow y \leftarrow$ is painted by $\rightarrow x$ (1)
$r2_{ap}^-$: $x \leftarrow$ paint $\rightarrow y \nRightarrow y \leftarrow$ is painted by $\rightarrow z$ (.6)

which allows to capture the degree of contradiction in pair T1-H4.

We have implemented two specialized engines, which have been trained on their corresponding monothematic datasets, resulting in the two thresholds t_{a-p} and t_{lex}. As for the order of application, first we run t_{lex} and then t_{a-p} was run on the output of the first.

5.3. Experiment 1

In this experiment we have built two small (i.e. 50 pairs each) monothematic datasets[3], one for negation ($[T, H]_{neg}$) and one for lexical similarity ($[T, H]_{lex}$), and we have tested both the neutrality and the disjointness conditions using two corresponding specialized engines, ED_{neg} and ED_{lex}. The two datasets, partially derived from T-H pairs included in the RTE data, are balanced

between positive and negative entailment and are aligned with respect to their Ts, while the Hs are built removing all the linguistic phenomena but the one under consideration, following the procedure described in Section 5.1. ED_{neg} was built with a number of manually created rules for negation, while ED_{lex} takes advantage of lexical rules automatically derived from WordNet.

Neutrality has been tested running ED_{neg} on $[T, H]_{lex}$ and running ED_{lex} on $[T, H]_{neg}$. In both cases we obtained a sum of the distances equal to 0, which verifies our definition (2). Then we verified the non-neutrality running ED_{neg} on $[T, H]_{neg}$ and running ED_{lex} on $[T, H]_{lex}$ and we obtained that no pair returned a 0 distance, according to definition (3). Although optimizing performance is not the main purpose of the experiment, ED_{neg} and ED_{lex} obtained an accuracy of 0.9 and 0.5 on their respective datasets.

5.4. Experiment 2

In this experiment we have built a non-monothematic dateset $[T, H]_{neg+lex}$ merging $[T, H]_{neg}$ and $[T, H]_{lex}$ used in Experiment 1. Then we have tested the two combination schemas, i.e. sum of distances and voting, proposed in Section 4.2. The merging has been obtained through the alignment of the Ts in $[T, H]_{neg}$ and $[T, H]_{lex}$, with the Hs containing both the neg and the lex phenomena. We run first ED_{lex} on $[T, H]_{neg+lex}$, because lexical phenomena involve single tokens (see Section 4.1), and then we run ED_{neg} on the same dataset.

The combination based on voting (Section 4.2) gave an overall accuracy of 0.66 over the 50 pairs. The voting strategy that produced the best results in case of parity of votes is the sequence $neutral < positive < negative$. The combination based on the sum of distances gave an overall accuracy of 0.64 over the same dataset.

6. Related Work

The intuition that specialized TE engines can be profitably devised has been explored in previous works on TE. For instance, negation is investigated in (Cabrio *et al.*, 2008), while temporal expressions and Named Entities in (Wang and Neumann, 2008). While such studies provide useful empirical evidences of the benefit of the idea, in this paper we attempt to provide a formal framework for the definition and the combinations of specialized engines.

Our work is also related and takes advantage of several studies which analyze the linguistic aspects involved in TE. In (Garoufi, 2007), a scheme for manual annotation of TE datasets (ARTE) is proposed, with the aim of highlighting a wide variety of entailment phenomena in the data and their distribution. An attempt to isolate the set of T-H pairs whose categorization can be accurately predicted based solely on syntactic cues has been carried out by (Vanderwende and Dolan, 2006). Finally, (Clark *et al.*, 2007) highlight that the majority of entailment cases rely on significant amount of the so called "common human understanding" of lexical and world knowledge.

7. Conclusions

In this paper we have presented a general framework for the combination of specialized and disjoint entailment engines, each addressing a specific phenomenon relevant for entailment judgements. Decomposing the entailment problem, in our opinion, may bring several advantages, including the possibility to train and evaluate specialized engines over monothematic dataset, allowing a much more modular development of complex systems.

This work can be considered as a proof of concept of the framework we have proposed. However, much work is still necessary in order to provide enough empirical evidences both in terms of the number of linguistic phenomena covered by the entailment engines and in terms of the complexity and naturalness of the dataset used in the experiments.

8. References

Cabrio, E., Kouylekov M., and Magnini, B.: Combining Specialized Entailment Engines for RTE-4. (2008). In *Proc. of the TAC 2008 Workshop on Textual Entailment*. Gaithersburg, Maryland, USA, 17 November.

Clark, P., Harrison, P., Thompson, J., Murray, W., Hobbs, J., and Fellbaum, C. (2007). On the Role of Lexical and World Knowledge in RTE3. In *Proc. of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*. Prague, Czech Republic, 28-29 June.

Dagan, I., Glickman, O. (2004). Probabilistic Textual Entailment: Generic Applied Modeling of Language Variability. In *Proc. of the PASCAL Workshop on Learning Methods for Text Understanding and Mining*. Grenoble, France, 26-29 January.

Garoufi, K. (2007) Towards a Better Understanding of Applied Textual Entailment. *Master Thesis*. Saarland University, Saarbrücken, Germany.

Giampiccolo, D., Trang Dang, H., Magnini, B., Dagan, I., and Cabrio, E. (2008) The Fourth PASCAL Recognising Textual Entailment Challenge. In *Proc. of the TAC 2008 Workshop on Textual Entailment*. Gaithersburg, Maryland, USA, 17 November.

Kouylekov, M. and Magnini, B. (2005). Tree Edit Distance for Textual Entailment, In *Proc. of RALNP-2005*.

Negri M., Kouylekov M., Magnini B., Mehdad Y. and Cabrio E. (2009) Towards Extensible Textual Entailment Engines: the EDITS Package, in *Proc. of the XI Conference of the AI*IA*, to appear.

Vanderwende, L. and Dolan, B. (2006). What Syntax can Contribute in the Entailment Task. In Quinonero-Candela, J., Dagan, I., Magnini, B., d'Alch-Buc, F. (Eds.), *Evaluating Predictive Uncertainty, Visual Object Classification, and Recognizing Textual Entailment, MLCW 2005*, LNCS Volume 3944, Springer-Verlag.

Wang, R., and Neumann, G. (2008). An Accuracy-Oriented Divide-and-Conquer Strategy. *Proc. of the TAC 2008 Workshop on Textual Entailment*. Gaithersburg, Maryland, 17 November.

Zhang, K., and Shasha D. (1990). Fast Algorithm for the Unit Cost Editing Distance Between Trees. In *Journal of Algorithms*. vol.11, December.