Hypothesis Transformation and Semantic Variability Rules Used in Recognizing Textual Entailment

Adrian Iftene
„Al. I. Cuza" University, Faculty of Computer Science, Iasi, Romania
adiftene@info.uaic.ro

Alexandra Balahur-Dobrescu
„Al. I. Cuza" University, Faculty of Computer Science, Iasi, Romania
abalahur@info.uaic.ro

Abstract

Based on the core approach of the tree edit distance algorithm, the system central module is designed to target the scope of TE – semantic variability. The main idea is to transform the hypothesis making use of extensive semantic knowledge from sources like DIRT, WordNet, Wikipedia, acronyms database. Additionally, we built a system to acquire the extra background knowledge needed and applied complex grammar rules for rephrasing in English.

1 Introduction

Many NLP applications need to recognize when the meaning of one text can be expressed by, or inferred from, another text. Information Retrieval (IR), Question Answering (QA), Information Extraction (IE), Text Summarization (SUM) are examples of applications that need to assess such a semantic relationship between text segments. Textual Entailment Recognition (RTE) (Dagan et al., 2006) has recently been proposed as an application independent task to capture such inferences.

This year our textual entailment system participated for the first time in the RTE[1] competition. Next chapters present its main parts, the detailed results obtained and some possible future improvements.

2 System description

The process requires an initial pre-processing, followed by the execution of a core module which uses the output of the first phase and obtains in the end the answers for all pairs. Figure 1 shows how the pre-processing is realized with the MINIPAR (Lin, 1998) and LingPipe[2] modules which provide the input for the core module. This one uses four databases: DIRT, Acronyms, Background knowledge and WordNet.

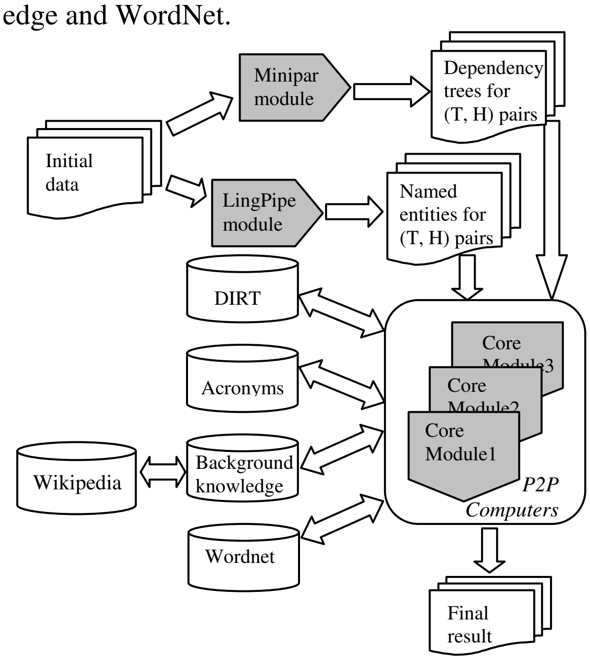

Figure 1: System architecture

The system architecture is based on a peer-to-peer networks design, in which neighboring computers collaborate in order to obtain the global fitness for every text-hypothesis pair. Eventually, based on the computed score, we decide for which pairs we have entailment. This type of architecture was used in order to increase the computation speed.

3 Initial pre-processing

The first step splits the initial file into pairs of files for text and hypothesis. All these files are then sent to the LingPipe module in order to find the Named entities.

[1] http://www.pascal-network.org/Challenges/RTE3/

[2] http://www.alias-i.com/lingpipe/

Proceedings of the Workshop on Textual Entailment and Paraphrasing, pages 125–130,
Prague, June 2007.

In parallel, we transform with MINIPAR both the text and the hypothesis into dependency trees. Figure 2 shows the output associated with the sentence: "*Le Beau Serge was directed by Chabrol.*".

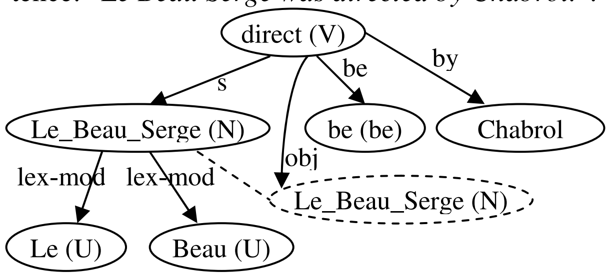

Figure 2: MINIPAR output – dependency tree

For every node from the MINIPAR output, we consider a stamp called **entity** with three main features: the node lemma, the father lemma and the edge label (which represents the relation between words) (like in Figure 3).

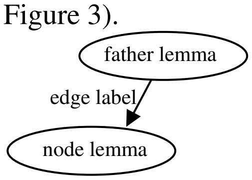

Figure 3: Entity components

Using this stamp, we can easily distinguish between nodes of the trees, even if these have the same lemma and the same father. In the example from Figure 1, for the "son" nodes we have two entities (*Le_Beau_Serge, direct, s*) and (*Le_Beau_Serge, direct, obj*).

4 The hypothesis tree transformation

Presently, the core of our approach is based on a tree edit distance algorithm applied on the dependency trees of both the text and the hypothesis (Kouylekov, Magnini 2005). If the distance (i.e. the cost of the editing operations) among the two trees is below a certain threshold, empirically estimated on the training data, then we assign an entailment relation between the two texts.

The main goal is to map every entity in the dependency tree associated with the hypothesis (called from now on *hypothesis tree*) to an entity in the dependency tree associated with the text (called from now on *text tree*).

For every mapping we calculate a **local fitness** value which indicates the appropriateness between entities. Subsequently, the global fitness is calculated from these partial values.

For every node (refers to the word contained in the node) which can be mapped directly to a node from the text tree, we consider the local fitness value to be 1. When we cannot map one word of the hypothesis to one node from the text, we have the following possibilities:

- If the word is a verb in the hypothesis tree, we use the DIRT resource (Lin and Pantel, 2001) in order to transform the hypothesis tree into an equivalent one, with the same nodes except the verb. Our aim in performing this transformation is to find a new value for the verb which can be better mapped in the text tree.

- If the word is marked as named entity by LingPipe, we try to use an acronyms' database[3] or if the word is a number we try to obtain information related to it from the background knowledge. In the event that even after these operations we cannot map the word from the hypothesis tree to one node from the text tree, no fitness values are computed for this case and we decide the final result: No entailment.

- Else, we use WordNet (Fellbaum, 1998) to look up synonyms for this word and try to map them to nodes from the text tree.

Following this procedure, for every transformation with DIRT or WordNet, we consider for local fitness the similarity value indicated by these resources. If after all checks, one node from the hypothesis tree cannot be mapped, some penalty is inserted in the value of the node local fitness.

4.1 The DIRT resource

For the verbs in the MINIPAR output, we extract templates with DIRT- like format. For the sample output in Figure 2, where we have a single verb "direct", we obtain the following list of "full" templates:N:s:V<direct>V:by:N and N:obj:V<direct> V:by:N. To this list we add a list of "partial" templates: N:s:V<direct>V:, :V<direct>V:by:N, :V<direct>V:by:N, and N:obj:V<direct>V:.

In the same way, we build a list with templates for the verbs in the text tree. With these two lists we perform a search in the DIRT database and extract the "best" trimming, considering the template type (full or partial) and the DIRT score.

According to the search results, we have the following situations:

[3] http://www.acronym-guide.com

a) *left – left* relations similarity

This case is described by the following two templates for the hypothesis and the text:

relation1 HypothesisVerb relation2

relation1 TextVerb relation3

This is the most frequent case, in which a verb is replaced by one of its synonyms or equivalent expressions

The transformation of the hypothesis tree is done in two steps:

1. Replace the relation2 with relation3,

2. Replace the verb from the hypothesis with the corresponding verb from the text. (see Figure 4).

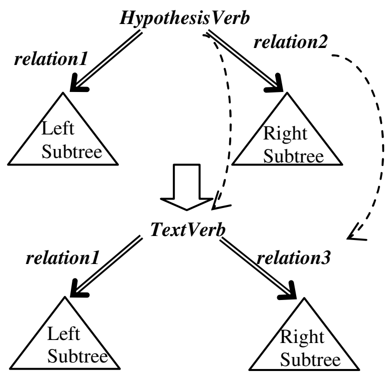

Figure 4: Left-left relation similarity

b) *right – right* relations similarity: the same idea from the previous case.

c) *left – right* relations similarity

This case can be described by the following two templates for the hypothesis and the text:

relation1 HypothesisVerb relation2

relation3 TextVerb relation1

The transformation of the hypothesis tree is:

1. Replace the relation2 with relation3,

2. Replace the verb from the hypothesis with the corresponding verb from the text.

3. Rotate the subtrees accordingly: left subtree will be right subtree and vice-versa right subtree will become left-subtree (as it can be observed in Figure 5).

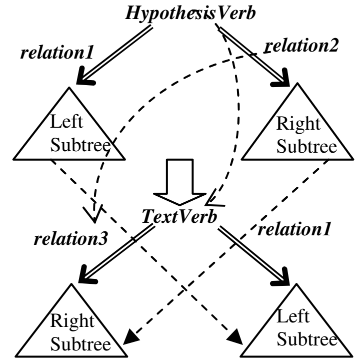

Figure 5: Left-right relation similarity

This case appears for pair 161 with the verb "attack":

T: "The demonstrators, convoked by the solidarity with Latin America committee, verbally attacked Salvadoran President Alfredo Cristiani."

H: "President Alfredo Cristiani was attacked by demonstrators."

In this case, for the text we have the template N:subj:V<attack>V:obj:N, and for the hypothesis the template N:obj:V<attack>V:by:N. Using DIRT, hypothesis H is transformed into:

H': Demonstrators attacked President Alfredo Cristiani.

Under this new form, H is easier comparable to T.

d) *right – left* relations similarity: the same idea from the previous case

For every node transformed with DIRT, we consider its local fitness as being the similarity value indicated by DIRT.

4.2 Extended WordNet

For non-verbs nodes from the hypothesis tree, if in the text tree we do not have nodes with the same lemma, we search for their synonyms in the extended WordNet[4]. For every synonym, we check to see if it appears in the text tree, and select the mapping with the best value according to the values from Extended WordNet. Subsequently, we change the word from the hypothesis tree with the word from WordNet and also its fitness with its indicated similarity value. For example, the relation between "relative" and "niece" is accomplished with a score of 0.078652.

[4] http://xwn.hlt.utdallas.edu/downloads.html

4.3 Acronyms

The acronyms' database helps our program find relations between the acronym and its meaning: "*US - United States*", and "*EU - European Union*". We change the word with the corresponding expression from this database. Since the meaning is the same, the local fitness is considered maximum, i.e. 1.

4.4 Background Knowledge

Some information cannot be deduced from the already used databases and thus we require additional means of gathering extra information of the form:

Argentine [is] Argentina
Netherlands [is] Holland
2 [is] two
Los Angeles [in] California
Chinese [in] China

Table 1: Background knowledge

Background knowledge was built semi-automatically, for the named entities (NEs) and for numbers from the hypothesis without correspondence in the text. For these NEs, we used a module to extract from Wikipedia[5] snippets with information related to them. Subsequently, we use this file with snippets and some previously set patterns of relations between NEs, with the goal to identify a known relation between the NE for which we have a problem and another NE.

If such a relation is found, we save it to an output file. Usually, not all relations are correct, but those that are will help us at the next run.

Our patterns identify two kinds of relations between words:

- "is", when the module extracts information of the form: *'Argentine Republic' (Spanish: 'Republica Argentina', IPA)'* or when explanations about the word are given in brackets, or when the extracted information contains one verb used to define something, like "is", "define", "represent": *'2' ('two') is a number*.

- "in" when information is of the form: *'Chinese' refers to anything pertaining to China* or in the form *Los Angeles County, California*, etc.

[5] http://en.wikipedia.org/wiki/Main_Page

In this case, the local fitness for the node is set to the maximum value for the [is]-type relations, and it receives some penalties for the [in]-type relation.

5 Determination of entailment

After transforming the hypothesis tree, we calculate a global fitness score using the ***extended local fitness*** value for every node from the hypothesis - which is calculated as sum of the following values:

1. local fitness obtained after the tree transformation and node mapping,

2. parent fitness after parent mapping,

3. mapping of the node edge label from the hypothesis tree onto the text tree,

4. node position (left, right) towards its father in the hypothesis and position of the mapping nodes from the text.

After calculating this extended local fitness score, the system computes a total fitness for all the nodes in the hypothesis tree and a negation value associated to the hypothesis tree. Tests have shown that out of these parameters, some are more important (the parameter at 1.) and some less (the parameter at 3.). Below you can observe an example of how the calculations for 3 and 4 are performed and what the negation rules are.

5.1 Edge label mapping

After the process of mapping between nodes, we check how edge labels from the hypothesis tree are mapped onto the text tree. Thus, having two adjacent nodes in the hypothesis, which are linked by an edge with a certain label, we search on the path between the nodes' mappings in the text tree this label. (see Figure 6)

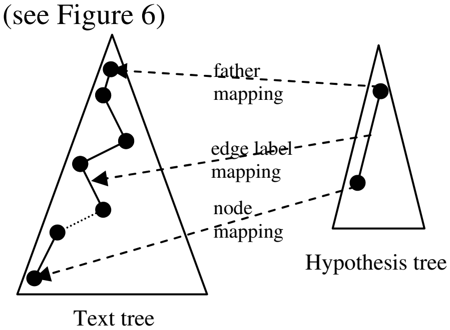

Figure 6: Entity mapping

It is possible that more nodes until the label of the edge linking the nodes in the hypothesis exist, or it is possible that this label is not even found on this path. According to the distance or to the case in which the label is missing, we insert some penalties in the extended local fitness.

5.2 Node position

After mapping the nodes, one of the two following possible situations may be encountered:

- The position of the node towards its father and the position of the mapping node towards its father's mapping are the same (left-left or right-right). In this case, the extended local fitness is incremented.

- The positions are different (left-right or right-left) and in this case a penalty is applied accordingly.

5.3 Negation rules

For every verb from the hypothesis we consider a Boolean value which indicates whether the verb has a negation or not, or, equivalently, if it is related to a verb or adverb "diminishing" its sense or not. Consequently, we check in its tree on its descending branches to see whether one or more of the following words are to be found (pure form of negation or modal verb in indicative or conditional form): "not, may, might, cannot, should, could, etc.". For each of these words we successively negate the initial truth value of the verb, which by default is "false". The final value depends on the number of such words.

Since the mapping is done for all verbs in the text and hypothesis, regardless of their original form in the snippet, we also focused on studying the impact of the original form of the verb on its overall meaning within the text. Infinitives can be identified when preceded by the particle "to". Observing this behavior, one complex rule for negation was built for the particle "to" when it precedes a verb. In this case, the sense of the infinitive is strongly influenced by the active verb, adverb or noun before the particle "to", as follows: if it is being preceded by a verb like "allow, impose, galvanize" or their synonyms, or adjective like "necessary, compulsory, free" or their synonyms or noun like "attempt", "trial" and their synonyms, the meaning of the verb in infinitive form is stressed upon and becomes "certain". For all other cases, the particle "to" diminish the certainty of the action expressed in the infinitive-form verb. Based on the synonyms database with the English thesaurus[6], we built two separate lists – one of "certainty stressing (preserving)" – "positive" and one of "certainty diminishing" – "negative" words. Some examples of these words are "probably", "likely" – from the list of "negative" words and "certainly", "absolutely" – from the list of "positive" words.

5.4 Global fitness calculation

We calculate for every node from the hypothesis tree the value of the extended local fitness, and afterwards consider the normalized value relative to the number of nodes from the hypothesis tree. We denote this result by *TF* (*total fitness*):

$$TF = \frac{\sum_{node \in H} ExtendedLocalFitness_{node}}{HypothesisNodesNumber}$$

After calculating this value, we compute a value *NV* (the *negation value*) indicating the number of verbs with the same value of negation, using the following formula:

$$NV = \frac{Positive\_VerbsNumber}{TotalNumberOfVerbs}$$

where the *Positive_VerbsNumber* is the number of non-negated verbs from the hypothesis using the negation rules, and *TotalNumberOfVerbs* is the total number of verbs from the hypothesis.

Because the maximum value for the extended fitness is 4, the complementary value of the *TF* is *4-TF* and the formula for the **global fitness** used is:

$$GlobalFitness = NV * TF + (1 - NV) * (4 - TF)$$

For pair 518 we have the following:

Initial entity	Node Fitness	Extended local fitness
(the, company, det)	1	3.125
(French, company, nn)	1	3.125
(railway, company, nn)	1	3.125
(company, call, s)	1	2.5
(be, call, be)	1	4
(call, -, -)	0.096	3.048
(company, call, obj)	1	1.125
(SNCF, call, desc)	1	2.625

Table 2: Entities extended fitness

[6] http://thesaurus.reference.com/

TF = (3.125 + 3.125 + 3.125 + 2.5 + 4 + 3.048 + 1.125 + 2.625)/8 = 22.673/8 = 2.834
NV = 1/1 = 1
GlobalFitness = 1*2.834+(1–1)*(4-2.834) = 2.834

Using the development data, we establish a threshold value of 2.06. Thus, pair 518 will have the answer "yes".

6 Results

Our system has a different behavior on different existing tasks, with higher results on Question Answering (0.87) and lower results on Information Extraction (0.57). We submitted two runs for our system, with different parameters used in calculating the extended local fitness. However, the results are almost the same (see Table 3).

	IE	**IR**	**QA**	**SUM**	**Global**
Run01	0.57	0.69	0.87	0.635	**0.6913**
Run02	0.57	0.685	0.865	0.645	**0.6913**

Table 3: Test results

To be able to see each component's relevance, the system was run in turn with each component removed. The results in the table below show that the system part verifying the NEs is the most important.

System Description	**Precision**	**Relevance**
Without DIRT	0.6876	**0.54 %**
Without WordNet	0.6800	**1.63 %**
Without Acronyms	0.6838	**1.08 %**
Without BK	0.6775	**2.00 %**
Without Negations	0.6763	**2.17 %**
Without NEs	0.5758	**16.71 %**

Table 4: Components relevance

7 Conclusions

The system's core algorithm is based on the tree edit distance approach, however, focused on transforming the hypothesis. It presently uses widespread syntactic analysis tools like Minipar, lexical resources like WordNet and LingPipe for Named Entities recognition and semantic resources like DIRT. The system's originality resides firstly in creating a part-of and equivalence ontology using an extraction module for Wikipedia data on NEs (the background knowledge), secondly in using a distinct database of acronyms from different domains, thirdly acquiring a set of important context influencing terms and creating a semantic equivalence set of rules based on English rephrasing concepts and last, but not least, on the technical side, using a distributed architecture for time performance enhancement.

The approach unveiled some issues related to the dependency to parsing tools, for example separating the verb and the preposition in the case of phrasal verbs, resulting in the change of meaning.

Another issue was identifying expressions that change context nuances, which we denoted by "positive" or "negative" words. Although we applied rules for them, we still require analysis to determine their accurate quantification.

For the future, our first concern is to search for a method to establish more precise values for penalties, in order to obtain lower values for pairs with No entailment. Furthermore, we will develop a new method to determine the multiplication coefficients for the parameters in the extended local fitness and the global threshold.

8 Acknowledgements

The authors thank the members of the NLP group in Iasi for their help and support at different stages of the system development. Special thanks go to Daniel Matei which was responsible for preparing all the input data.

The work on this project is partially financed by Siemens VDO Iaşi and by the CEEX Rotel project number 29.

References

Dagan, I., Glickman, O., and Magnini, B. 2006. *The PASCAL Recognising Textual Entailment Challenge*. In Quiñonero-Candela et al., editors, *MLCW 2005, LNAI Volume 3944*, pages 177-190. Springer-Verlag.

Fellbaum, C. 1998. *WordNet: An Electronic Lexical Database*. MIT Press, Cambridge, Mass.

Kouylekov, M. and Magnini, B. 2005. *Recognizing Textual Entailment with Tree Edit Distance Algorithms*. In Proceedings of the First Challenge Workshop Recognising Textual Entailment, Pages 17-20, 25–28 April, 2005, Southampton, U.K.

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

Lin, D., and Pantel, P. 2001. *DIRT - Discovery of Inference Rules from Text*. In Proceedings of ACM Conference on Knowledge Discovery and Data Mining (KDD-01). pp. 323-328. San Francisco, CA.