SINAI at RTE-7: Integrating Personalized Page Rank Vectors into EDITS

Arturo Montejo-Ráez, Fernando Martínez-Santiago
José Manuel Perea-Ortega, L. Alfonso Ureña-López
Department of Computer Science
University of Jaén
23071 Jaén, Spain
{amontejo, dofer, jmperea, laurena}@ujaen.es

Abstract

This document describes the participation of the SINAI Research Group in the 7^{th} challenge on Recognition of Textual Entailment (RTE). Our approach extends the promising results obtained in the last campaing into a well know framework for textual entailment recognition known as EDITS. Although the proposed solution is modest, results encourage us in the use of Personalized Page Rank as a technique for generating weighted vectors of synsets per term in texts and applying distance metrics to compute a value of lexical similarity between terms found in hypotheses and those in candidate texts.

1 Introduction

The SINAI Research group has participated in the 7^{th} RTE challenge, organized as a workshop within the Text Analysis Conference in 2011 (TAC 2011). This document describes the system implemented for resolving the task of recognizing textual entailment. The approach followed consists in the integration of a lexical similarity module into the EDITS framework. Regarding our last participation (Montejo-Ráez et al., 2010), this time we apply PPV (Personalized Page Rank Vectors) calculus into a non supervised solution.

2 RTE-7 challenge

Recognizing textual entailment is a task that has attracted the attention of a large group of researchers in the area of Natural Language Processing during the last years. From 2008, the organizer of a related challenge has been the Information Technology Laboratory, at the National Institute of Standards and Technology, becoming RTE a track at the Text Analysis Conference[1]. This has brought the opportunity to take RTE challenge to more realistic and more application-oriented scenarios.

This year, we only participated in the *Main taks*. Again, ablation tests were required to participants in order to provide analysis of the different modules involved in the systems proposed, so a better understanding of the effect of each component to the final performance of a system can be reached.

3 System architecture

Edit Distance Textual Entailment Suite (EDITS) is a software package aimed at recognizing entailment relations between two portions of text (Kouylekov and Negri, 2010), termed as T and H. The system is based on edit distance algorithms, and computes the T-H distance as the cost of the edit operations (i.e. insertion, deletion and substitution) that are necessary to transform T into H. The way Personalized Page Rank Vectors (PPVs) are integrated into EDITS has been by means of rules. Rules are used to provide the Entailment Engine with knowledge (e.g. lexical, syntactic, semantic) about the probability of entailment or contradiction between elements of T and H (see figure 3. Rules are invoked by cost schemes to influence the cost of substitutions between elements of T and H. Typically, the cost of the substitution between two elements A and B is inversely proportional to the probability that A entails B. EDITS applies the lexical rules without taking into account the context of the word and this is a drawback for our approach because PPVs consider not only the T-H pair of words but the context of both words in order to calculate the probability of entailment. Thus we have modified EDITS to provide the context of the words to the Rule Repository manager.

We have tested several configurations of EDITS varying the distance and similarity algorithms used to compute a distance score between T-H pairs. EDITS provides a set of predefined algorithms, includ-

[1] http://www.nist.gov/tac

ing edit distance algorithms, and similarity algorithms adapted to the EDITS framework.

EDITS provides several distance algorithms implementations. We have found the best results by using Edit Distance Algorithm. This is a token-based version of the Levenshtein distance algorithm, with edit operations defined over sequences of tokes of T and H. The distance measures have been calculated by using word overlap which computes an overall (distance) score as the proportion of common words in T and H. One word from T can substitute more than one of the words in H. The score returned by the algorithm is the sum of the cost of all substitutions divided by the number of words in H.

Similarity algorithms are adapted to the EDITS distance framework by transforming measures of the lexical/semantic similarity between T and H into distance measures. These algorithms are also adapted to use the three edit operations to support overlap calculation, and define term weights. For instance, substitutable terms in T and H can be treated as equal, and non-overlapping terms can be weighted proportionally to their insertion/deletion costs. When tunning EDITS, we found the best results by using Word Overlap and Jaro-Winkler distance:

- *Word Overlap*: computes an overall (distance) score as the proportion of common words in T and H. In the current implementation the algorithm uses the cost scheme to find the less costly substitution of a word from H with a word form T. One word from T can substitute more than one of the words in H. The score returned by the algorithm is the sum of the cost of all substitutions divided by the number of words in H;

- *Jaro-Winkler distance*: a similarity algorithm between strings, adapted to similarity between words. The algorithm uses the cost scheme to define if two words are the same (they have a 0 cost of substitution). The entailment score is obtained by subtracting the obtained Jaro-Winkler metric from 1 (i.e score(A,B)=1-JW(A,B)).

Personalized Page Rank vectors used in the lexical similarity module consists in a ranked sequence of WordNet (Fellbaum, 1998) synsets weighted according to a random walk algorithm. We have used the UKB software (Agirre and Soroa, 2009) to generate the PPVs used in our system. Random walk algorithms are inspired originally by the Google PageRank algorithm (Page et al., 1999) and the idea behind its use is to represent each term as a group of semantically close synsets, so the lexical similarity is computed as a distance between these vectors. A similar approach has been used by (Ramage et al., 2009) to compute text semantic similarity in RTE environments, and also as solution for word sense disambiguation (Agirre and Soroa, 2009).

```
<rule entailment="ENTAILMENT">
        <t>acquire</t>
        <h>own</h>
        <probability>0.95</probability>
</rule>
<rule entailment="CONTRADICTION">
        <t>beautiful</t>
        <h>ugly</h>
        <probability>0.88</probability>
</rule>
```

Figure 1: Example of XML Rule Repository in EDITS

Last year, a vector of weighted synset nodes was computed for each sentence found in every text and hypotheses. The cosine distance between these vectors was used as feature in a supervised learning process. This time, the PPVs are computed `per term`, i.e. a vector of weighted synsets is generated for each term through a random walk process over the WordNet graph in its version 3.0. All the terms in the context (the text where the term appears) are used to initialize the graph state. We found that version 3.0 performed better than 1.7 in the previous work (10% better), so experiments run only on 3.0 version of WordNet. Finally, the distance between two terms is computed as the distance between respective PPVs applying the cosine formula and, for the rules passed to EDITS, each pair of terms is associated with 1 minus the cosine of vectors (as EDITS expects a *cost* values instead of distances).

As an example of a PPV, when processed, the text *"Overall, we're still having a hard time with it, mainly because we're not finding it in an early phase."* becomes the vector of weighted synsets: `[02190088-a:0.0016, 12613907-n:0.0004, 01680996-a:0.0002, 00745831-a:0.0002, ...]`

4 Experiments and results

Three runs were submitted: **SINAI1**, **SINAI2** and **SINAI3**.

- SINAI1: EDITS with its Distance Algorithm (based on Levenshtein algorithm at word level) with Word Overlap (as described above).

- SINAI2: Same as above, with using Jaro-

Winkler distance instead of Word Overlap.

- SINAI3: Same as SINAI1, but adding lexical rules for term-to-term similarity based on PPVs.

Therefore, SINAI1 can be seen as the ablation test of SINAI3, as the lexical rule based on PPVs distances is not applied.

	Microaveraged		
Run	**Precision**	**Recall**	**F1**
SINAI1	47.08	8.64	14.60
SINAI2	42.99	3.52	6.50
SINAI3	47.30	8.72	14.72

Table 1: Microaveraged results for Main task

	Macroaveraged		
Run	**Precision**	**Recall**	**F1**
SINAI1	50.15	9.21	15.56
SINAI2	42.95	3.75	6.89
SINAI3	50.60	9.27	15.68

Table 2: Macroaveraged results for Main task

In the RTE-7 Main task, 13 teams submitted a total of 33 runs. Statistics over 33 runs ranked by micro-averaged F1 determine that the highest value was 0.48 and the median was 0,419. Therefore, our best result achieved with the *SINAI3* experiment (15.68) is far from the highest F1 but also from the median. Complete evaluation measures are detailed in Table 1 and Table 2.

The integration of lexical rules contributes to a small increase in precision and recall, but not as relevant as we expected regarding the results obtained with this technique in the RTE-6 campaing. It remains to analyze the very low recall reached with EDITS .

5 Conclusions and further work

The results obtained were discouraging. We expected to obtain better results with a standard configuration of the EDITS framework and that the integration of PPVs into the process performed by the tool would improve significantly our results. According to results obtained, we have to explore a better configuration of EDITS and also the integration of PPVs into other architectures like, for example, Biutee (Stern et al., 2010).

6 Acknowledgements

This work has been partially supported by a grant from the Spanish Government, project TEXT-COOL 2.0 (TIN2009-13391-C04-02), the “Fondo Europeo de Desarrollo Regional” (FEDER), a grant from the Andalusian Government, project GeOasis (P08-TIC-41999) and a grant from Spanish Government through the scientific network *TIMM: Tratamiento de Información Multimodal y Multilingüe* (TIN2009-06135-E).

References

[Agirre and Soroa2009] Eneko Agirre and Aitor Soroa. Personalizing pagerank for word sense disambiguation. In *EACL '09: Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics*, pages 33–41, Morristown, NJ, USA, 2009. Association for Computational Linguistics.

[Fellbaum1998] Christiane Fellbaum, editor. *WordNet: An Electronic Lexical Database*. MIT Press, Cambridge, MA, 1998.

[Kouylekov and Negri2010] M. Kouylekov and M. Negri. An open-source package for recognizing textual entailment. In *The Annual Meeting of the Association for Computational Linguistics ACL 2010*, 2010.

[Montejo-Ráez et al.2010] A. Montejo-Ráez, J.M. Perea-Ortega, F. Martínez-Santiago, and L.A. Ure na López. Sinai at rte-6: Personalized page rank vectors and named entities. In *Proceedings of the Third Text Analysis Conference (TAC 2010)*, Gaithersburg, Maryland, USA, November 2010.

[Page et al.1999] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. The pagerank citation ranking: Bringing order to the web. Technical report, Stanford University, 1999.

[Ramage et al.2009] Daniel Ramage, Anna N. Rafferty, and Christopher D. Manning. Random walks for text semantic similarity. In *TextGraphs-4: Proceedings of the 2009 Workshop on Graph-based Methods for Natural Language Processing*, pages 23–31, Morristown, NJ, USA, 2009. Association for Computational Linguistics.

[Stern et al.2010] Asher Stern, Eyal Shnarch, Amnon Lotan, Shachar Mirkin, Lili Kotlerman, Naomi Zeichner, Jonathan Berant, and Ido Dagan. Rule chaining and approximate match in textual inference. In *Proceedings of the Text Analysis Conference (TAC)*, Gaithersburg, Maryland, 2010.