

# Information Synthesis for Answer Validation

Rui Wang<sup>1</sup> and Günter Neumann<sup>2</sup>

<sup>1</sup> Saarland University  
66123 Saarbrücken, Germany  
rwang@coli.uni-sb.de

<sup>2</sup> LT-Lab, DFKI  
Stuhlsatzenhausweg 3, 66123 Saarbrücken, Germany  
neumann@dfki.de

**Abstract.** This paper proposes an integration of *Recognizing Textual Entailment* (RTE) with other additional information to deal with the *Answer Validation* task. The additional information used in our participation in the *Answer Validation Exercise* (AVE 2008) is from named-entity (NE) recognizer, question analysis component, etc. We have submitted two runs, one run for English and the other for German, achieving f-measures of 0.64 and 0.61 respectively. Compared with our system last year, which purely depends on the output of the RTE system, the extra information does show its effectiveness.

## 1 Introduction and Related Work

Using *Recognizing Textual Entailment* (RTE-1 – [3]; RTE-2 – [1]) to do *Answer Validation* has shown a great success [9]. We also developed our own RTE system and participated in AVE2007 [12]. The RTE system proposed a new sentence representation extracted from the dependency structure, and utilized the Subsequence Kernel method [2] to perform machine learning. We have achieved fairly high results on both the RTE-2 data set [10] and the RTE-3 data set [11], especially on *Information Extraction* (IE) and *Question Answering* (QA) pairs.

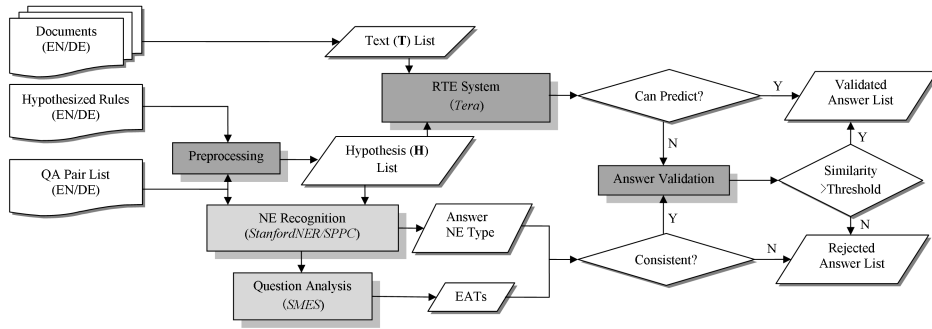
However, on the AVE data sets, we still found much space for the improvement. Therefore, based on the system we developed last year, our motivation this year is to see whether using extra information, e.g. *named-entity* (NE) recognition, question analysis, etc., can make further improvement on the final results.

## 2 The RTE System

The RTE system ([10]; [11]) is developed for the RTE-3 Challenge [5]. The system contains a main approach with two backup strategies. The main approach extracts parts of the dependency structures to form a new representation, named *Tree Skeleton*, as the feature space and then applies *Subsequence Kernels* to represent TSs and perform machine learning. The backup strategies will deal with the **T-H** pairs which cannot be solved by the main approach. One backup strategy is called *Triple Matcher*,

as it calculates the overlapping ratio on top of the dependency structures in a triple representation; the other is simply a *Bag-of-Words* (BoW) method, which calculates the overlapping ratio of words in **T** and **H**.

### 3 The AVE System



**Fig. 1.** Our AVE system uses the RTE system (**Tera** – *Textual Entailment Recognition for Application*) as a core component. The preprocessing module mainly adapts questions, their corresponding answers, and supporting documents into **Text (T)**-**Hypothesis (H)** pairs, assisted by some manually designed patterns. The post-processing module (i.e. the *Answer Validation* in the picture) will validate each answer and select a most proper one based on the output of the RTE system. The new modules added are the *NE Recognition* and *Question Analysis*. Thus, we will have extra information like NEs in the answers, *Expected Answer Types* (EATs).

#### 3.1 Preprocessing and Post-processing

Since the input of the AVE task is a list of questions, their corresponding answers and the documents containing these answers, we need to adapt them into **T-H** pairs for the RTE system. In order to combine the question and the answer into a statement, manually construct some language patterns for the input questions (cf. [12] for more details). The constructed **T-H** pairs can be the input for any generic RTE systems. In practice, after applying our RTE system, if the **T-H** pairs are covered by our main approach, we will directly use the answers; if not, we will use a threshold to decide the answer based on the two similarity scores and together with other information (see the following subsection).

The post-processing is straightforward, the “YES” entailment cases will be validated answers and the “NO” entailment cases will be rejected answers. In addition, the selected answers (i.e. the best answers) will naturally be the pairs covered by our main approach or (if not,) with the highest similarity scores.

### 3.2 Additional Components

The RTE system is used as a core component of the AVE system. Based on the error analysis of last year’s results, this year, we use additional components to filter out noisy candidates. Therefore, two extra components are added to the architecture, the NE recognizer and the question analyzer. For NE recognition, we use StanfordNER [4] for English and SMES [8] for German; and for question analysis, we use the SMES system [8]. The detailed workflow is as follows,

1. Annotate NEs in **H**, store them in an NE list; if the answer is an NE, store the NE type as A’\_Type;
2. Analyze the question and obtain expected answer type, store it as A\_Type;
3. Synthesize all the information, i.e. NE list, A\_Type, A’\_Type, BoW similarity, Triple similarity, etc.

Then, heuristic rules are straightforward to be applied, e.g. checking the consistence between A\_Type and A’\_Type, checking whether all (or how many of) the NEs also appear in the documents.

## 4 Results

We have submitted two runs, one for English and one for German.

**Table 1.** Results of our submissions compared with last year’s

Submission Runs	Recall	Precision	F-measure	Estimated QA Performance	QA Accuracy
100% VALIDATED (EN)	1	0.08	0.14	N/A	N/A
50% VALIDATED (EN)	0.5	0.08	0.13	N/A	N/A
Perfect Selection (EN)	N/A	N/A	N/A	0.56	<b>0.34</b>
Best QA System (EN)	N/A	N/A	N/A	0.21	<b>0.21</b>
dfki07-run1 (EN)	0.62	0.37	<b>0.46</b>	N/A	0.16
dfki07-run2 (EN)	0.71	0.44	<b>0.55</b>	N/A	0.21
dfki08run1 (EN)	0.78	0.54	<b>0.64</b>	0.34	<b>0.24</b>
100% VALIDATED(DE)	1	0.12	0.21	N/A	N/A
50% VALIDATED (DE)	0.5	0.12	0.19	N/A	N/A
Perfect Selection (DE)	N/A	N/A	N/A	0.77	<b>0.52</b>
Best QA System (DE)	N/A	N/A	N/A	0.38	<b>0.38</b>
dfki08run1 (DE)	0.71	0.54	0.61	0.52	<b>0.43</b>

In the table, we notice that both for English and German, our validation system outperforms the best QA systems, which suggests the necessity of the validation step. Although there is a gap between the system performance and the perfect selection, the results are quite satisfactory. If we compare this year’s results with last year’s, the additional information does improve the results significantly. Comparing the recall and precision, for both languages, the latter is worse. After an error analysis (cf. the Working Notes), we find that to further synthesize the information we have, i.e. NE annotation and dependency parsing, might be more beneficial.

## 5 Conclusion and Future Work

To sum up, based on the experience of last year's participation, apart from the RTE core system, we add two extra components, NE recognizer and question analyzer, to further improve the results. The strategy is quite successful according to the comparison of system performances. However, the problem has not been fully solved. Filtering some documents in the preprocessing step could be even more effective than working on the post-processing phase; another direction considered by us is to take a closer look at the different performances between different languages.

## References

1. Bar-Haim, R., Dagan, I., Dolan, B., Ferro, L., Giampiccolo, D., Magnini, B. and Szpektor, I. 2006. The Second PASCAL Recognising Textual Entailment Challenge. In Proceedings of the Second PASCAL Challenges Workshop on Recognising Textual Entailment.
2. Bunescu, R. and Mooney, R. 2006. Subsequence Kernels for Relation Extraction. In Advances in Neural Information Processing Systems 18. MIT Press.
3. Dagan, I., Glickman, O., and Magnini, B. 2006. The PASCAL Recognising Textual Entailment Challenge. In Quiñero-Candela et al., editors, MLCW 2005.
4. Finkel, J.R., Grenager, T., and Manning, C. 2005. Incorporating Non-local Information into Information Extraction Systems by Gibbs Sampling. Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL 2005).
5. Giampiccolo, D., Magnini, B., Dagan, I., and Dolan, B. 2007. The Third PASCAL Recognizing Textual Entailment Challenge. In Proceedings of the Workshop on Textual Entailment and Paraphrasing, pages 1–9, Prague, June 2007.
6. Gildea, D. and Palmer, M. 2002. The Necessity of Parsing for Predicate Argument Recognition. In Proceedings of the 40th Meeting of the Association for Computational Linguistics (ACL 2002):239-246, Philadelphia, PA.
7. Lin, D. 1998. Dependency-based Evaluation of MINIPAR. In Workshop on the Evaluation of Parsing Systems.
8. Neumann, G. and Piskorski, J. 2002. A Shallow Text Processing Core Engine. Journal of Computational Intelligence, Volume 18, Number 3, 2002, pages 451-476.
9. Anselmo Peñas, Álvaro Rodrigo, Felisa Verdejo. 2007. Overview of the Answer Validation Exercise 2007. In the CLEF 2007 Working Notes.
10. Wang, R. and Neumann, G. 2007a. Recognizing Textual Entailment Using a Subsequence Kernel Method. In Proc. of AAAI 2007.
11. Wang, R. and Neumann, G. 2007b. Recognizing Textual Entailment Using Sentence Similarity based on Dependency Tree Skeletons. In Proceedings of the Workshop on Textual Entailment and Paraphrasing, pages 36–41, Prague, June 2007.
12. Wang, R. and Neumann, G. 2007c. DFKI-LT at AVE 2007: Using Recognizing Textual Entailment for Answer Validation. In online proceedings of CLEF 2007 Working Notes, ISBN: 2-912335-31-0, September 2007, Budapest, Hungary.