Relation Validation via Textual Entailment

Rui Wang[1], Günter Neumann[2]

[1] Saarland University, 66123 Saarbrücken, Germany
rwang@coli.uni-sb.de
[2] LT-Lab, DFKI, Stuhlsatzenhausweg 3, 66123 Saarbrücken, Germany
neumann@dfki.de

Abstract. This paper addresses a subtask of relation extraction, namely *Relation Validation*. Relation validation can be described as follows: given an instance of a relation and a relevant text fragment, the system is asked to decide whether this instance is true or not. Instead of following the common approaches of using statistical or context features directly, we propose a method based on textual entailment (called ***ReVaS***). We set up two different experiments to test our system: one is based on an annotated data set; the other is based on real web data via the integration of ***ReVaS*** with an existing IE system. For the latter case, we examine in detail the two aspects of the validation process, i.e. *directionality* and *strictness*. The results suggest that textual entailment is a feasible way for the relation validation task.

Keywords: Relation Validation, Textual Entailment, Information Extraction

1 Introduction and Relation Work

Information extraction (IE) has been a hot topic for many years both in the area of natural language processing. An important task involved is relation extraction, which automatically identifies instances of certain relations of interest in some document collection, e.g. *work_for*(<person>, <company>, <location>).

Conventional IE systems are usually domain-dependent and adapting the system to a new domain requires a high amount of manual labor, such as specifying and implementing relation-specific extraction patterns or annotating large amounts of training corpora. A new trend in information extraction is trying to collect information directly from the web and "understand" it (Etzioni et al., 2005; Banko et al., 2007). One crucial point for such relation extraction systems is to be able to estimate the quality of the extracted instances. Web documents are relatively noisy compared with corpora constructed for particular usages. Therefore, a careful evaluation (or *validation*) step is needed after the extraction process.

Another effort made by researchers developing unsupervised IE systems, e.g. Shinyama and Sekine (2006), Xu et al. (2007), and Downey et al. (2007). Here, the evaluation of those newly obtained instances with a good confidence score has a great impact on the final results (Agichtein, 2006). This also adds more burdens to, in our context, the validation module.

As far as we know, *Relation Validation* has not been addressed as an independent subtask of relation extraction in the literature, though many researchers have already mentioned the importance of the estimation metrics. The SnowBall system (Agichtein, 2006) has applied an Expectation-Maximization method to estimate tuple and pattern confidence, which might lead to the problem of overly general patterns. The KnowItAll system (Etzioni et al., 2005) has extended PMI (Turney, 2001) and used heuristics like signal to noise ratio to test the plausibility of the candidates. The former is computationally expensive; and the latter shifts the problem onto the statistical distributions, which might not be correct. The REALM system (Downey et al., 2007) has combined HMM-based and n-gram-based language models and ranked candidate extractions by the likelihood that they are correct. This captures the local features quite well, but may lose long distance linguistic dependencies. Consequently, instead of applying methods of analyzing context or statistical features directly as the previous work, we propose a novel strategy to deal with this validation step – via *textual entailment*. On the one hand, it allows more syntactic/semantic variations for instances of certain relations; on the other hand, a domain-independent credibility is provided.

The *Recognizing Textual Entailment* (RTE) task was proposed by Dagan et al. (2006) and refined by Bar-Haim et al. (2006). It is defined as recognizing, given two text fragments, whether the meaning of one text can be inferred (entailed) from the other. The entailment relationship is a directional one from *Text* – **T** to *Hypothesis* – **H**. We have developed our Relation Validation System (***ReVaS***) based on our previous work on RTE (Wang and Neumann, 2007a). Both the main approach involved and the evaluation results have shown a precision-oriented character of our RTE system. Especially for IE relevant data, we have achieved a large improvement on covered cases, compared with baselines and also state-of-the-art systems. This motivates us to apply our RTE system to tasks requiring high precision, e.g. answer validation for question answering (Wang and Neumann, 2007b), and relation validation for information extraction (this paper).

2 The System Description

Fig. 1 shows the architecture of the ***ReVaS*** system integrated with an IE system. ***ReVaS*** consists of a preprocessing module, an RTE core engine (***Tera*** – *Textual Entailment Recognition for Applications*), and a post-processing module. As an add-on component for the original IE system, ***ReVaS*** glues the instances of relations into natural language sentences (i.e. *hypotheses*) using hypothesized patterns, checks the entailment relation between the relevant documents and the hypotheses, and annotates a confidence score to each instance, so as to perform the validation step.

2.1 The RTE Core Engine

The RTE core engine contains a main approach with two backup strategies (Anonymous, 2007a). In brief, the main approach firstly extracts common nouns between **T** and **H**; then it locates them in the dependency parse tree as *Foot Nodes*

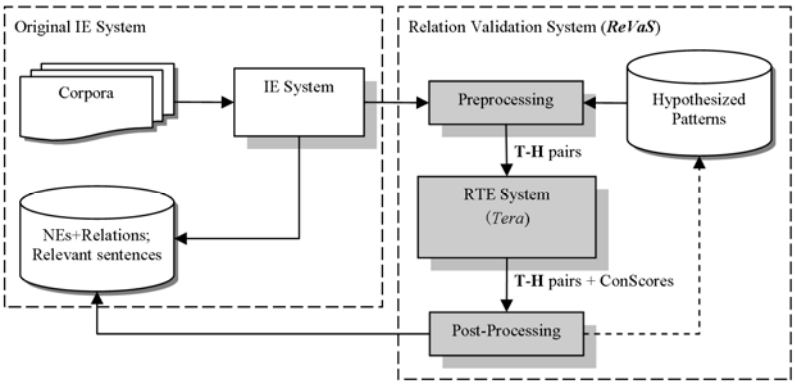

Fig. 1. The architecture of the integration of ***ReVaS*** with an IE system

(FNs). Starting from the FNs, a common parent node, which will be named as *Root Node* (RN), can be found in each tree; Altogether, FNs, the RN, and the dependency paths in-between will form a *Tree Skeleton* (TS) for each tree. On top of this feature space, we can apply subsequence kernels to represent these TSs and perform kernel-based machine learning to predict the final answers discriminatively.

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**.

2.2 The Relation Validation Procedure

Since the input for the RTE system is one or more **T**-**H** pairs, we need to preprocess the output of the IE system. Usually, the output is a list of relations and the corresponding NEs, together with the text from which the relations are extracted. For instance, consider the following text,

"*The union has hired a number of professional consultants in its battle with the company, including* ***Ray Rogers*** *of Corporate Campaign Inc., the* ***New York*** *labor consultant who developed the strategy at* ***Geo. A. Hormel*** *& Co.'s* ***Austin***, ***Minn.***, *meatpacking plant last year. That campaign, which included a strike, faltered when the company hired new workers and the International Meatpacking Union wrested control of the local union from* ***Rogers'*** *supporters.*"

And the target relation type obtained might be birthplace relation, which is between a *Person Name* (PN) and a *Location Name* (LN). Back to the text, several PNs and LNs could be found,

PN: "*Ray Rogers*", "*Rogers*"

LN: "*New York*", "*Austin*", "*Minn.*"

Consequently, the possible NE pairs with birthplace relation are,

<PN, LN>: <"*Ray Rogers*", "*New York*">, <"*Rogers*", "*Austin*">, …

Assume that those instances are extracted from the text by a relation extraction system. Now our task is to check each of them whether the relation holds for those NE pairs.

The adaptation into an RTE problem is straightforward. Using NE pairs with relations, we can construct the following sentences using simple natural language patterns,

"*Ray Rogers* ***is born in*** *New York.*"

"***The birthplace of*** *Rogers* ***is*** *Austin.*"

…

These sentences serve as the **H** in a **T**-**H** pair, and the **T** is naturally the original text. Thus, several **T**-**H** pairs can be constructed accordingly. Afterwards, the RTE system will determine a confidence score to each instance of relations, together with a judgment of validated or rejected under a certain threshold, which can be learned from another corpus or set manually.

The main difference of our RTE-based validation module from other common evaluation metrics is that we can obtain semantic variations via textual entailment. Though the patterns we are using to construct the hypotheses are rather simple, the entailment-based validation process makes it more semantically flexible than the direct feature-based similarity calculation (cf. Wang and Neumann 2007a).

3 The System Evaluation

In order to fully evaluate our ***ReVaS*** system, we have set up two different experiments: one is to test the system independently based on an annotated data set; the other is to integrate ***ReVaS*** into an existing IE system as a validation component and test it on real web data.

3.1. The Experiment on Annotated Data

The data set we have used for this experiment is from the *BinRel* corpus (Roth and Yih, 2004), which contains three parsed corpora with NEs and binary relations of NEs listed after each sentence: 1) the *kill* relation corpus; 2) the *birthplace* relation corpus; and 3) the negative corpus (i.e. there are NEs annotated, but no instances of such two kinds of relations).

We have used the original texts as **T**s, and combined NEs using simple patterns of the *kill* relation and the *birthplace* relation into **H**s. In detail, a positive *kill* **T**-**H** pair will be an existing *kill* relation between two NEs, which are both PNs; a negative one will be two PNs with no *kill* relation in-between (similar to Yangarber et al. (2000)). The positive *birthplace* cases are similar to the example mentioned in 2.2, and negative ones contain other relations between the PN and the LN, e.g. *workplace* relation.

In all, 918 *kill* pairs (268 positive cases) and 849 *birthplace* pairs (199 positive cases) have been constructed from the corpus. The evaluation metrics here is the *accuracy*. 10-fold cross validation has been performed and the results are shown in the following table,

Table 1 Results of the Relation Validation Task

Systems	*kill* relation	*birthplace* relation
BoW (Baseline1)	72.0%	75.0%
Triple (Baseline2)	70.3%	76.4%
Main + Backups	84.1%	86.5%

As we described in 2.1, the RTE system consists of a main approach plus two backup strategies. We take the two backup strategies as two baseline systems for comparison.

3.2. The Experiment on Web Data

To further test our ***ReVaS*** system, we have integrated it into an existing unsupervised IE system ***IDEX*** developed in our lab (Eichler et al., 2008). If a topic (in form of keywords) is given to ***IDEX***, it will use it as a query to a search engine on the World Wide Web. The retrieved documents will be analyzed using a dependency parser and an NE recognizer. The relations of NEs are identified via locating NEs in the dependency parse tree and finding the common parent node, which is normally a verb. The extracted instances of relations will be further clustered into different relation groups.

We have collected in all 2674 instances of binary relations from the ***IDEX*** system, including various relation types. The following table gives out some examples,

Table 2 Output examples of the ***IDEX*** system

Relation	NE1	NE2
located	*Berlin*	*Germany*
working	*Tricon*	*Bangkok*
say	*Britons*	*Slovakians*
…	…	…

Being different from the annotated data set, these instances of relations returned by ***IDEX*** are all positive examples for the system. However, even with the clustering, it is not trivial to identify the names of relation types. To make full use of the data, we hypothesize a relation type first and then check each instance whether it is of this relation type. Therefore, instances consistent with this relation type are positive cases (as a gold standard here), and all the other instances are negative ones.

The evaluation metrics we have applied are precision and relative recall (Frické, 1998). The reason for using relative recall instead of normal recall is that we do not know how many instances of one particular relation we can find from the web. Thus, we take one setting of our system as a reference (i.e. its recall is assumed as 1.0) and other settings' recalls will be compared to it. The precision is more interesting to us in this validation task, since it tells us how accurate the system is.

Two aspects we want to analyze based on the experiments, i.e. *directionality* and *strictness*.

A relation is said to be directional if the action of that relation is from one NE to the other, i.e. the NE pair is asymmetric; a relation is non-directional if the pair is symmetric. As we know, some relations containing two NEs with the same type are

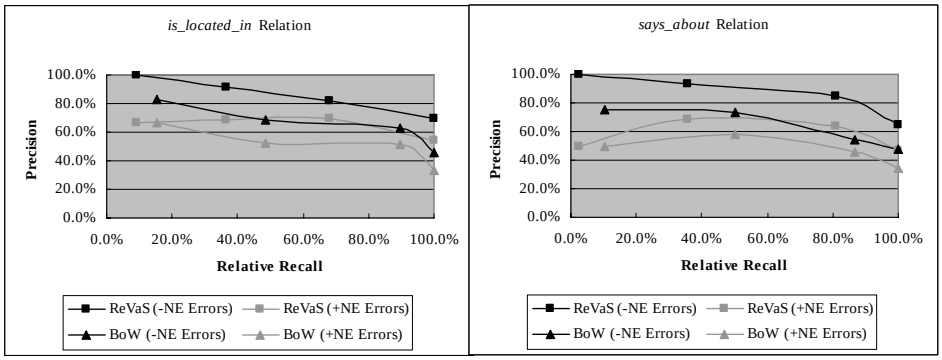

Fig. 2. The results of our system on *is_located_in* relation and *says_about* relation

directional[1], e.g. *kill*(PN1, PN2), *is_located_in*(LN1, LN2); while some are not, e.g. *friend_of*(PN1, PN2). Therefore, in practice, once we obtain the two NEs and relation in-between, we have to check both directions, i.e. relation(NE1, NE2) and relation(NE2, NE1). If the hypothesized relation is directional, only one of them passes the check; if it is a non-directional one, both of them pass; and all the other cases are negative instances.

The other aspect is strictness. The ***ReVaS*** system could be set up with different thresholds for the confidence scores from the RTE system, which leads to different effects of validation. Generally speaking, the stricter the system is, the fewer results will be validated, but the higher accuracy it will have. This strictness will reflect on the relation validation task as the tolerance of semantic variation among all the instances.

For the RTE system, we have combined the main approach with two backup strategies (the same ones as before in 3.1) by taking average of them. The main approach will contribute 1.0 – *positive*, 0.0 – *negative*, or 0.5 – *not covered*. The baseline system here is the Bag-of-Words system. Figure 2 above shows the system performance with hypothesized relation types *is_located_in* and *say_about*.

For each relation, we have tested the system with four different thresholds (i.e. strictness) for the confidence score, i.e. 0.9, 0.8, 0.7, and 0.6. We have taken the threshold 0.6 as a reference, namely its recall is set to be 100.0%. Then other recall scores are the percentage of the number those settings correctly validate divided by the number the reference setting correctly validates. Two lines respectively represent the precisions with NE errors and without. We will present a detailed error analysis and discussion in the following section.

3.3 Discussions

After taking a close look at the results, our system can successfully capture some linguistic variations as we expected. For example, the following example which can be correctly validated by our system indicates the implicit *is_located_in* relation

[1] Those relations with different NE types are naturally directional.

between the two NEs, "…***The City Partner Hotel am Gendarmenmarkt*** *offers our guests a personal home in the heart of* ***Berlin***." Using parsing instead of extracting statistical features also helps us to jump over the apposition to identify the *say_about* relation, "***Randall Lewis****, a spokesman for the Squamish First Nation, said* ***CN*** *...*"

As shown in the two graphs above, errors concerning wrong NEs have occupied a large portion of all the errors. For instance, "***CCNB*** *office and core facility The CCNB Core Facility will be centrally located in a designated building on the Campus of the Charité in Berlin* ***Mitte***." The partial recognition of the NE "*Berlin Mitte*" makes the validated relation trivial, though correct. Another interesting example is "*She received her* ***PhD*** *from the* ***University of Wisconsin-Madison*** *in 1997*." Although "*PhD*" is not an NE, the *is_located_in* relation still holds.

Errors concerning relations mainly fall into the following two categories: 1) similar relations, e.g. between birthplace relation and workplace relation, "…***David W. Roubik****, a staff scientist with the Smithsonian Tropical Research Institute in* ***Balboa****, Panama.*" and 2) the depth of analyzing modifiers, e.g. "*Geography Setting* ***Berlin*** *is located in eastern* ***Germany****, about 110 kilometers (65 miles) west of the border with* ***Poland***."

The complexity of real web data also impairs the performance. For instance, the following paragraph is extracted from a blog,

"*But the end of* ***Zoo Station*** *is the end of yet another era in* ***Berlin****, the '60s through the '80s, and one can only wonder where the junkies in west Berlin will congregate after it's gone. posted by* ***Ed Ward*** *@ 1:22 AM 2 comments 2 Comments: At 3:08 PM,* ***Daniel Rubin*** *said... First time I saw the* ***Hamburg Bahnhof*** *it was like a scene from a horror movie - - all these grizzled creatures staggering around as the loudspeakers blasted* ***Mozart****…*"

In the RTE system, we have a method to deal with cross-sentence relations, by adjoining tree skeletons of different sentences. However, this makes the situation worse, when we want to figure out who ("*Ed Ward*", "*Daniel Rubin*", or even "*Mozart*") says about what ("*Zoo Station*", "*Berlin*", "*Hamburg Bahnhof*", or "*Mozart*"). Here, the structure tags of the web document might help to separate the paragraphs, but it needs further investigations.

4 Conclusion and Future Work

We have presented our work on a subtask of information extraction, i.e. relation validation. It is rarely addressed as a separate task as far as we know. The novelty of our approach is to apply textual entailment techniques to deal with the validation task. Due to the precision-oriented method of our RTE system, experiments on both annotated data and web data with an integration of an existing IE system have shown the advantages of our approach. The results suggest textual entailment as a feasible way for validation tasks, which requires a high confidence.

In principle, our approach can be applied for validating more complex relations than binary ones. Either decomposing the complex relations into several binary ones or extending our tree skeleton structure is a possible way. Furthermore, the entailment-based confidence score can be directly used as a criterion for relation

extraction. The method is exactly the same: to make a hypothesized relation and then extract "validated" instances from the texts. Apart from these, our method might also be an interesting way to automatically evaluate the outputs of different information extraction systems.

Acknowledgements

The work presented here was partially supported by a research grant from BMBF to the DFKI project HyLaP (FKZ: 01 IW F02) and the EC-funded project QALL-ME.

References

1. R. Wang and G. Neumann. 2007a. Recognizing Textual Entailment Using a Subsequence Kernel Method. In Proceedings of AAAI-2007, Vancoucer.
2. R. Wang and G. Neumann. 2007b. DFKI–LT at AVE 2007: Using Recognizing Textual Entailment for Answer Validation. In online proceedings of CLEF 2007 Working Notes, Budapest, September, 2007, ISBN: 2-912335-31-0.
3. E. Agichtein. 2006. Confidence Estimation Methods for Partially Supervised Relation Extraction. In SDM 2006.
4. M. Banko, M. Cafarella, and O. Etzioni. 2007. Open Information Extraction from the Web. In Proceedings of IJCAI 2007. Hyderabad, India.
5. R. Bar-Haim, I. Dagan, B. Dolan, L. Ferro, D. Giampiccolo, B. Magnini, and I. Szpektor. 2006. The Second PASCAL Recognising Textual Entailment Challenge. In Proceedings of the PASCAL RTE-2 Workshop, Venice, Italy.
6. I. Dagan, O. Glickman, and B. Magnini. 2006. The PASCAL Recognising Textual Entailment Challenge. In Quiñonero-Candela et al., editors, MLCW 2005, LNAI Volume 3944, pages 177-190. Springer-Verlag.
7. K. Eichler, H. Hemsen and G. Neumann. 2008. Unsupervised Relation Extraction from Web Documents. In Proceedings of LREC 2008, Marrakesh, Morroco.
8. O. Etzioni, M. Cafarella, D. Downey, A. Popescu, T. Shaked, S. Soderland, D. S. Weld, and A. Yates. 2005. Unsupervised Named-Entity Extraction from the Web: An Experimental Study. Artificial Intelligence 165(1):91-134.
9. D. Downey, S. Schoenmackers, and O. Etzioni. 2007. Sparse Information Extraction: Unsupervised Language Models to the Rescue. In Proceedings of ACL 2007, pages 696–703, Prague, Czech Republic.
10. M. Frické. 1998. Measuring recall. Journal of Information Science, Vol. 24, No. 6, 409-417.
11. D. Roth and W. Yih. 2004. A linear programming formulation for global inference in natural language tasks. In Proceedings of CoNLL 2004, pp1-8.
12. Y. Shinyama and S. Sekine. 2006. Preemptive Information Extraction using Unristricted Relation Discovery. In Proceedings of HLT-NAACL06.
13. P. D. Turney. 2001. Mining the Web for Synonyms: PMI-IR versus LSA on TOEFL. In Proceedings of ECML 2001, pages 491-502, Freiburg, Germany.
14. F. Xu, H. Uszkoreit, and H. Li. 2007. A Seed-driven Bottom-up Machine Learning Framework for Extracting Relations of Various Complexity. In Proceedings of ACL 2007.
15. R. Yangarber, R. Grishman, P. Tapanainen, and S. Huttunen. 2000. Automatic Acquisition of Domain Knowledge for Information Extraction. In Proceedings of COLING 2000, Saarbrücken, Germany.