A Multi-Dimensional Classification Approach towards Recognizing Textual Semantic Relations

Rui Wang and Yi Zhang

Saarland University and DFKI GmbH
Saarbruecken, 66123, Germany
rwang@coli.uni-sb.de, yzhang@coli.uni-sb.de

Abstract. Recognizing textual entailment has been known as a challenging task, with many proposed approaches focusing on solving it independently. From a broader perspective, there are other semantic relations between pairs of texts, e.g., paraphrase, contradiction, overlapping, independence, etc. In this paper, we propose three basic measurements: *relatedness*, *inconsistency*, and *inequality*, to characterize these closely related *Textual Semantic Relations*. We show empirically the effectiveness of these measurements for the recognition tasks (e.g. an improvement of 3.1% of accuracy for entailment recognition) with features extracted from dependency paths of the joint syntactic and semantic graph. With the semantic relation space based on these three dimensions, we show this is a way to achieve a better understanding of general semantic relations between texts.

1 Introduction

Recognizing Textual Entailment (RTE) has been known as a challenging task, with interesting close relations to both natural language understanding (i.e. meaning interpretation) and natural language processing (i.e. applicable to various tasks). The task was defined as to recognize a specific relation (i.e. *entailment*) between two texts, *text* (T) and *hypothesis* (H). While many attempts have been made to solve the problem in a standalone manner, fewer investigated the relation between entailment and other possible semantic relations between pairs of texts.

From this perspective, we could categorize most approaches into two groups. In the first group, either the system deals with different cases of entailment with specialized modules [1, 2], to learn various lexical or inference rules [3, 4] from large scale corpora, or to apply logic inference techniques with manually-crafted rules [5]. In the second group, people work on (seemingly) different tasks, e.g. identifying *contradiction* [6], acquiring paraphrase [7], and finding answers to the questions [8]. They try to connect these tasks with the RTE research and this paper falls into this category as well.

The term *semantic relation* refers to the relations that hold between the meaning of two linguistic units. It is commonly used to describe relations between pairs of words, e.g., synonym, hypernym, etc. However, it has also been used in a wider sense to refer to relations between larger linguistic expressions or texts, such as paraphrasing, textual entailment, etc. [9]. We refer to the latter relations as *Textual Semantic Relations* (TSRs), to differentiate them from the study of lexical semantic relations. At a first glance, such generalization makes the already challenging recognition tasks even more complex. However, if these TSRs are mutually related, the simultaneous prediction will make much sense.

In previous work, [10] have shown that recognizing *relatedness* between two texts can be viewed as an intermediate step for entailment and contradiction recognition. [11] proposed five elementary relations between text pairs, EQUIVALENT, FORWARD (ENTAILMENT), REVERSE (ENTAILMENT), INDEPENDENT, and EXCLUSIVE and represent them in terms of entailment and negation. [12] proposed an annotation scheme for semantic relations between text pairs, including six labels, BACKWARD ENTAILMENT, FORWARD ENTAILMENT, EQUALITY, CONTRADICTION, OVERLAPPING, and INDEPENDENT.

In order to obtain a better characterization of all these TSRs, in this paper, we propose three basic numerical features, *relatedness*, *inconsistency*, and *inequality*. We show empirically these features are effective for the TSR recognition tasks, e.g. an improvement of 3.1% of accuracy on entailment recognition and 2.3% on paraphrase identification (Section 5.2). Although these three values are not entirely orthogonal to each other, we can still build an approximate three-dimensional semantic relation space, and observe distributional difference between various TSRs.

2 Related Work

While textual entailment analysis is now widely spotted in many NLP applications, e.g. question answering [13] and machine translation evaluation [14], the state-of-the-art performance of RTE systems is far from satisfactory. According to the yearly RTE challenges (from RTE-1 in 2005 [15] to 2009 [16]), the average performance of the particpating systems is around 60% on the two-way annotated data (ENTAILMENT vs. NON-ENTAILMENT) and even worse on the three-way annotated data (ENTAILMENT, CONTRADICTION, and UNKNOWN) starting from the RTE-3 pilot task[1]. Nevertheless, successful systems include both machine-learning-based classifier [17] and logic-form-based inferencer [18].

A variant of the logic inference rule is the textual inference rule or other (syntactic or semantic) representations closer to the surface text than the logic form. The DIRT rule collection [19] has been applied to the RTE task, although the improvement is limited [20]. [4] acquired unary rules instead of the binary DIRT-style rules and showed improvement on the accuracy, although it is still far from satisfactory. Both the logic-rule-based and textual-rule-based systems suffer from either a laborious and fragile system with hand-crafted rules (i.e. being lack of recall) or a large collection of “noisy” rules (i.e. being lack of precision). In order to avoid these disadvantages, we will treat RTE as a classification task and apply feature-based machine learning techniques to achieve robustness.

As for the feature space of the machine learning approaches, tree and graph structures are widely considered. For instance, [21] and their following work used tree editing distance algorithms; and [22] chose a graph matching method. An alternative to the feature engineering attempts, support vector machines (SVMs) with different kernels

[1] http://nlp.stanford.edu/RTE3-pilot/

are also popular in this classification task. Both the (constituent) tree kernel [23, 24] and the subsequence kernel based on syntactic dependency paths [25] were quite successful. Therefore, in our work, we will also use an SVM-based classifier. Instead of using the tree kernels, we extract features based on both syntactic and semantic dependency paths (or triples) as an approximation of the meaning, which greatly reduce the number of dimensions of the feature vectors and achieve better efficiency.

As we mentioned in the introduction, besides RTE, the main goal of this paper is to build a general framework for recognizing different TSRs. Previous work on this aspect includes [11]'s proposal of five elementary relations between texts and our own inventory of six semantic relations [12]. [11] tested their natural logic system on the FraCaS dataset [26], which is manually constructed and focuses more on the different linguistic (semantic) phenomena. While the system achieved quite good results on this "text-book" style dataset, the evaluation on the real world texts (e.g. the RTE datasets) did not show much advantage of their approach.

Apart from the entailment recognition, [6] attempted to discover contradiction, although it was then proved to be an even harder problem. There is also rich literature on paraphrase (which can be viewed as a bi-directional entailment relation) acquisition and application [27, etc.]. [9] mainly focused on EQUIVALENCE and CONTRADICTION recognition in terms of subjective texts, i.e. opinions. The recent work by [8] proposed a generic system based on a tree editing model to recognize textual entailment, paraphrase, and answers to questions. We follow this line of research and draw a more general picture of all these semantic relations.

3 Textual Semantic Relations

We firstly introduce the TSRs we consider in this paper, and then the three features we use to characterize the different relations.

In previous study [12], we proposed six relations, BACKWARD ENTAILMENT, FORWARD ENTAILMENT, EQUALITY, CONTRADICTION, OVERLAPPING, and INDEPENDENT. If we consider the unidirectional relations between an ordered pair of texts (i.e. from the first one (**T**) to the second one (**H**)), the first two relations can be collapsed into one. We use the name ENTAILMENT, but we mean a strict directional relation, i.e. **T** entails **H**, but **H** does not entail **T**. The original goal of having both OVERLAPPING and INDEPENDENT is to capture the spectrum of relatedness. However, in practice, even the human annotators found it difficult to agree on many cases. Therefore, we also collapse the last two relations into one, UNKNOWN, following the RTE label convention. After changing EQUALITY into PARAPHRASE, the TSRs we mention in the rest of the paper would be, CONTRADICTION (C), ENTAILMENT (E), PARAPHRASE (P), and UNKNOWN (U).

Although semantic relations are supposed to be situation-independent (i.e. consistently true or false in every possible world), in practice, every text pair is always in a certain context. Our goal here is to differentiate these four TSRs using some latent features shared by them, instead of verifying them in all possible worlds. We assume, there exists a simplified low-dimension semantic relation space. While the identification of effective dimensions is a complex question (see Section 5.3 for more detailed discus-

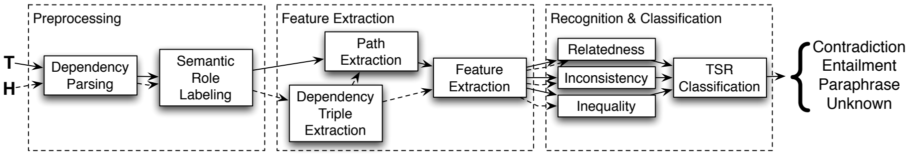

Fig. 1. Workflow of the System

sion), we start only with three dimensions: *Relatedness (Rel)*, *Inconsistency (Inc)*, and *Inequality (Ine)*, and assume that the different TSRs would be scattered on this space with different distributions.

Relatedness captures how relevant the two texts are. PARAPHRASE would be one extreme (fully related), and UNKNOWN would the other extreme. *Inconsistency* measures whether or how contradictory the two texts are. CONTRADICTION has the highest inconsistency, and the others do not have. *Inequality* mainly differentiates the asymmetric ENTAILMENT from the symmetric PARAPHRASE. Although the other two relations are symmetric as well, we assume unequal information is contained in **T** and **H**. All three features will be numerical.

There are two approximations here: i) the number of dimensions in the real semantic relation space is much higher; ii) these three dimensions we pick are not really orthogonal to each other (as shown in the experiments). Nevertheless, we hope to benefit from the generality of these measures in the TSR recognition task and will show the empirical results in Section 5.

4 General Framework

The workflow of the system is shown in Figure 1 and the details of the important components will be elaborated on in the following sections.

4.1 Preprocessing

In this paper, we generally refer to all the linguistic analyses on the texts as *preprocessing*. The output of this procedure is a unified graph representation, which approximates the meaning of the input text. In particular, after tokenization and POS tagging, we did dependency parsing and semantic role labeling.

Tokenization and POS Tagging We use the Penn Treebank style tokenization throughout the various processing stages. **TnT**, an HMM-based POS tagger trained with Wall Street Journal sections of the PTB, was used to automatically predict the part-of-speech of each token in the texts and hypotheses.

Dependency Parsing For obtaining the syntactic dependencies, we use two dependency parsers, MSTParser [28] and MaltParser [29]. MSTParser is a graph-based dependency parser where the best parse tree is acquired by searching for a spanning tree which maximize the score on an either partially or fully connected dependency graph. MaltParser is a transition-based incremental dependency parser, which is language-independent and data-driven. It contains a deterministic algorithm, which can be viewed as a variant of the basic shift-reduce algorithm. The combination of two parsers achieves state-of-the-art performance.

Semantic Role Labeling The statistical dependency parsers provide shallow syntactic analyses of the entailment pairs through the limited vocabulary of the dependency relations. In our case, the CoNLL shared task dataset from 2008 were used to train the statistical dependency parsing models. While such dependencies capture interesting syntactic relations, when compared to the parsing systems with deeper representations, the contained information is not as detailed. To compensate for this, we used a shallow semantic parser to predict the semantic role relations in the **T** and **H** of entailment pairs. The shallow semantic parser was also trained with CoNLL 2008 shared task dataset, with semantic roles extracted from the Propbank and Nombank annotations [30].

4.2 Feature Extraction

We firstly extract all the dependency triples from **H**, like <word, dependency relation, word>, excluding those having stop words. Then, we use the word pairs contained in the extracted dependency triples as anchors to find the corresponding *dependency paths* in **T**. For the following three representations, we apply slightly different algorithms to find the dependency path between two words,

Syntactic Dependency Tree We traverse the tree and find the corresponding dependency path connecting the two words;

Semantic Dependency Graph We use Dijkstra's algorithm to find the shortest path between the two words;

Joint Dependency Graph We assign different weights to syntactic and semantic dependencies and apply Dijkstra's algorithm to find the shortest path (with the lowest cost)[2].

For the features, we firstly check whether there are dependency triples extracted from **H** as well as whether the same words can be found in **T**. Only if the corresponding dependency paths are successfully located in **T**, we could extract the following features. The direction of each dependency relation or path could be interesting. We use a boolean value to represent whether **T**-path contains dependency relations with different directions of the **H**-path.

Notice that all the dependency paths from **H** have length 1[3]. If the length of the **T**-path is also 1, we can directly compare the two dependency relations; otherwise, we compare each of the dependency relation contained the **T**-path with **H**-path one by one[4].

[2] In practice, we simply set semantic dependency costs at 0.5 and syntactic dependency costs at 1.0, to show the preferences on the former when both exist.

[3] The length of one dependency path is defined as the number of dependency relations contained in the path.

[4] Enlightened by [25], we exclude some dependency relations like "CONJ", "COORD", "APPO", etc., heuristically, since usually they will not change the relationship between the two words at both ends of the path.

By comparing the **T**-path with **H**-path, we mainly focus on two values, the category of the dependency relation (e.g. syntactic dependency vs. semantic dependency) and the content of the dependency relation (e.g. A1 vs. AM-LOC). We also incorporate the string value of the dependency relation pair and make it boolean depending on whether it occurs or not.

Table 1 shows the feature types we extract from each **T**-**H** pair. H_NULL? means whether **H** has dependencies; T_NULL? means whether **T** has the corresponding paths (using the same word pairs found in **H**); DIR is whether the direction of the path **T** the same as **H**; MULTI? adds a prefix, M_, to the REL_PAIR features, if the **T**-path is longer than one dependency relation; DEP_SAME? checks whether the two dependency types are the same, i.e. syntactic and semantic dependencies; REL_SIM? only occurs when two semantic dependencies are compared, meaning whether they have the same prefixes, e.g. C-, AM-, etc.; REL_SAME? checks whether the two dependency relations are the same; and REL_PAIR simple concatenates the two relation labels together.

Table 1. Feature types of different settings of the system.

	H_NULL?	T_NULL?	DIR	MULTI?	DEP_SAME?	REL_SIM?	REL_SAME?	REL_PAIR
Syn Dep		+	+	+			+	+
Sem Dep	+	+	+	+		+	+	+
Joint	+	+	+	+	+	+	+	+

4.3 TSR Recognition

After obtaining all the features for text pairs with different TSRs, we train three classifiers for the three measurements, *relatedness*, *inconsistency*, and *inequality*, and test on the whole dataset to obtain the numerical values. The training data are labeled according the scheme shown in Table 2. The later recognition of the TSRs are based on these three measurements.

Table 2. Training data of the three classifiers

	relatedness	*inconsistency*	*inequality*
PARAPHRASE	+	−	−
ENTAILMENT	+	−	+
CONTRADICTION	+	+	+
UNKNOWN	−	−	+

5 Experiments

5.1 Datasets

Table 3 gives an overview of all the datasets we use in our experiments and we briefly describe them in the following.

Table 3. Collection of heterogenous datasets with different annotation schemes.

Corpora	Paraphrase(P)	Entailment(E)	Contradiction(C)	Unknown(U)
AMT (584)		Facts (406)	Counter-Facts (178)	
MSR (5841)	Paraphrase (3940)	Non-Paraphrase (1901)		
PETE (367)		YES (194)	NO (173)	
RTE (2200)	ENTAILMENT (1100)		CONTRADICTION (330)	UNKNOWN (770)
TSR (260)	Equality (3)	Forward/Backward Entailment (10/27)	Contradiction (17)	Overlapping&Independent (203)
Total (9252)	3943	637	525	973

AMT is a dataset we constructed using the crowd-sourcing technique [31]. We used Amazon's Mechanical Turk[5], online non-expert annotators [32] to perform the task. Basically, we show the Turkers a paragraph of text with one highlighted named-entity and ask them to write some facts or counter-facts about it. There are three blank lines given for the annotators to fill in. For each task, we show five texts, and for each text, we ask three Turkers to do it. In all, we collected 406 valid facts and 178 counter-facts, which will be viewed as E and C respectively.

MSR is a paraphrase corpus provided by Microsoft Research [33]. It is a collection of manually annotated sentential paraphrases. This dataset consists of 5841 pairs of sentences which have been extracted from news sources on the web, along with human annotations indicating whether each pair captures a paraphrase/semantic equivalence relationship.

PETE is taken from the SemEval-2010 Task #12, Parser Evaluation using Textual Entailment[6] [34]. The dataset contains 367 pairs of texts in all and has a focus on entailments involving mainly the syntactic information. The annotation is two-way, YES would be converted into ENTAILMENT and NO could be either CONTRADICTION or UNKNOWN. Since each text pair only concerns about one syntactic phenomenon, the entailment relation is directional, excluding the paraphrases.

RTE is a mixture of RTE-4 (1000) and RTE-5 (1200) datasets. Both are annotated in three-way, but the ENTAILMENT cases actually include PARAPHRASE as well. We did not include the unofficial three-way annotation of the RTE-3 pilot task.

TSR is the dataset we annotated under the annotation scheme mentioned in Section 3. The sentence pairs were extracted from the the RST Discourse Treebank (RST-DT)[7]. The annotation was done by two annotators in two rounds. The inter-annotator agreement is 91.2% and the kappa score is 0.775. We take all the valid and agreed sentence pairs (260) as the TSR dataset here.

We randomly sample 250 **T**-**H** pairs from each dataset as the test sets (1000 pairs in all). The rest of the data are then randomly selected to create a balance training set with equal number of instance pairs from each class.

[5] `https://www.mturk.com/mturk/`

[6] `http://pete.yuret.com/guide`

[7] Available from the LDC: `http://www.ldc.upenn.edu/Catalog/CatalogEntry.jsp?catalogId=LDC2002T07`

5.2 Setup & Results

First of all, we take the PETE dataset to do binary classification (ENTAILMENT vs. NON-ENTAILMENT) on a small scale to confirm that both syntactic and semantic dependency structures are useful. The features extracted from the joint dependency graph improve the model of features purely from the syntactic dependency tree by as much as 10% of accuracy. Therefore, in the rest of the experiments, we will take the joint dependency graph as the default structure to extract features.

For comparison, we configure our system in the following two ways to compose different baseline systems: 1) from the classification strategy perspective, the direct four-class classification would be the baseline (*Direct Joint* in Table 4), compared with the main system with a two-stage classification (*3-D Model*); and 2) from the feature set point of view, we take the bag-of-words similarity as the baseline[8] (*Direct BoW*), compared with the main system using both syntactic and semantic dependency structures (i.e. the *3-D Model*). Table 4 shows the results.

Table 4. Results of the system with different configurations and different evaluation metrics.

Systems	4-Way	3-Way	2-Way	
	(C, E, P, U)	(C, E&P, U)	(E&P, Others)	(P, Others)
Direct BoW	39.3%	54.5%	63.2%	62.1%
Direct Joint	42.3%	50.9%	66.8%	77.3%
3-D Model	**45.9%**	**58.2%**	**69.9%**	**79.6%**

Notice that E here indicates the strict directional entailment excluding the bidirectional ones (i.e. P), which makes the task much harder (as we will see it more in Section 5.3). Nevertheless, the main approach, 3-D Model, improves the system performance greatly in all aspects, compared with the baselines. Apart from the self-evaluation, we also compare our approach with others' systems. Due to the difference in datasets, the numbers are only indicative.

Table 5. System comparison under the RTE annotation schemes ($*$ indicates different datasets).

RTE	3-Way	2-Way		
	(C, E&P, U)	Acc.	Prec.	Rec.
3-D Model	58.2%	**69.9%**	**75.9%**	53.4%
*M&M, 2007(NL)	-	59.4%	70.1%	36.1%
*H&S, 2010	-	62.8%	61.9%	**71.2%**
Our Prev.	**59.1%**	69.2%	-	-
*RTE-4 Median	50.7%	61.6%	-	-
*RTE-5 Avg.	52.0%	61.2%	-	-

For the RTE comparison (Table 5), the datasets are partially different due to the mixture of datasets. For reference, we re-run our previous system on the new dataset (indicated as *Our Prev.*, which was one of the top system in the previous RTE challenges).

[8] The bag-of-words similarity has shown to be a strong baseline in the previous RTE challenges.

The results show that our new approach (*3-D Model*) catches the previous system on the three-way RTE and outperforms it on the two-way task. And both systems achieves much better results than the average. [11]'s system based on natural logic (*M&M, 2007*) is precision-oriented while [8]'s (*H&S, 2010*) is recall-oriented. Our system achieves the highest precision among them.

Table 6. System comparison under the paraphrase identification task ($*$ indicates the test sets).

P vs. Non-P	Acc.	Prec	Rec.
3-D Model	**79.6%**	57.2%	72.8%
*D&S, 2009 (QG)	73.9%	74.9%	**91.3%**
*D&S, 2009 (PoE)	76.1%	**79.6%**	86%
*H&S, 2010	73.2%	75.7%	87.8%

Besides the RTE task, we also compare our approach with other paraphrase identification systems (Table 6). [35] proposed two systems, one with high-recall (*D&S, 2009 (QG)*, using a quasi-synchronous grammar) and the other with high-precision (*D&S, 2009 (PoE)*, using a product of experts to combine the QG model with lexical overlap features). *H&S, 2010* is the same system in Table 5. Although our system has lower precision and recall, our accuracy ranks the top, which indicates that our approach is better at non-paraphrase recognition.

Notice that, our system is not fine-tuned to any specific recognition task. Instead, we build a general framework for recognizing all the four TSRs. We also include heterogenous datasets collected by various methods in order to achieve the robustness of the system. On the contrary, if one is interested in recognizing one specific relation, a closer look at the data distribution would help with the feature selection.

5.3 Discussions

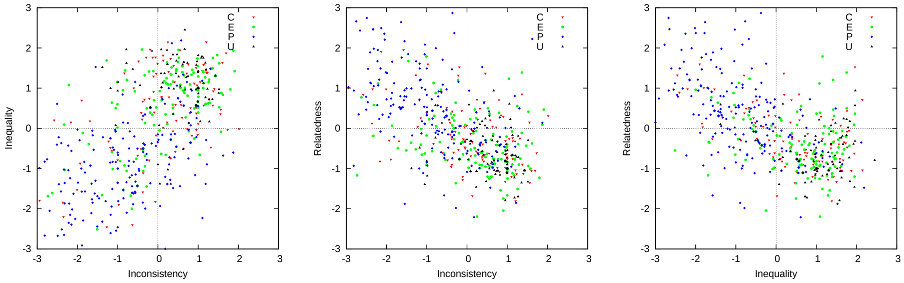

Fig. 2. Test data in the three-dimensional semantic relation space projected onto the three planes.

While the empirical results show a practical advantage of applying the three-dimensional space model in the TSR recognition task, in this subsection, we investigate whether this simplified semantic relation space with the chosen axises is a good approximation for these TSRs. We plot all the test data into this space and Figure 2 shows three different projections onto each two-dimensional plane.

Although the improvement on recognition accuracy is encouraging, these three measurements cannot fully separate different TSRs in this space. P is clearly differentiated from the others and most of the data points stay in the region of low inconsistency (i.e. consistent), low inequality (i.e. equal), and high relatedness. However, the other three TSRs behave rather similarly to each other in terms of the regions.

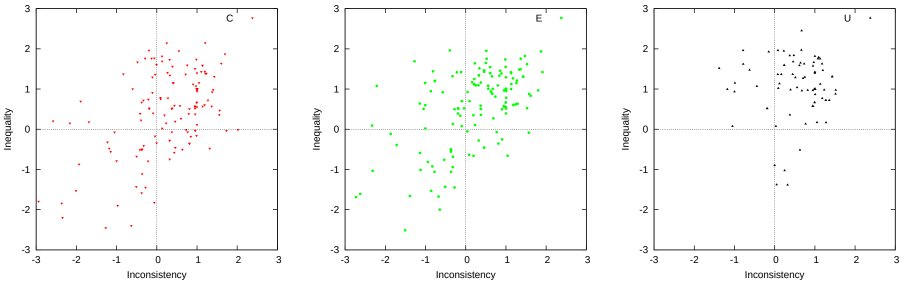

Fig. 3. C, E, and U test data projected onto the inconsistency-inequality plane.

Figure 3 shows the other three TSRs on the same plane, *inconsistency-inequality*. Although the general trend of these three groups of data points is similar, slight differences do exist. U is rather restricted in the region of high inconsistency and high inequality; while the other two spread a bit over the whole plane. We have expected the contrary behavior of C and E in terms of inconsistency, but it seems that our inconsistency measuring module is not as solid as the relatedness measure. This is in accordance with the fact that for the original three-way RTE task C is also the most difficult category to be recognized.

A even more difficult measurement is the inequality. Among all the four TSRs, the worst result is on E, which roots from the suboptimal inequality recognition. In retrospect, the matching methods applied to the **T**-**H** pair cannot capture the directionality or the semantic implication, but rather obtain a symmetric measurement, and thus it explains the success of paraphrase recognition. Additionally, this might also suggest that, in the traditional RTE task, the high performance might attribute to the P “section” of the entailment, while the real directional E is still very difficult to capture.

6 Conclusion and Future Work

In this paper, we present our approach of recognizing different textual semantic relations based on a three-dimensional model. *Relatedness*, *inconsistency*, and *inequality* are considered as the basic measurements for the recognition task as well as the dimensions of the semantic relation space. We show empirically the effectiveness of this approach with a feature model based on dependency paths of the joint syntactic and semantic graph. We also interpret the results and the remaining difficulties visually.

There are several issues on the list: 1) Inequality seems to be difficult to define and measure, which suggests to consider other possible dimensions; 2) we are looking for a systematic way to tune the general system for specific TSR recognition tasks; and 3) we have not incorporated lexical resources (e.g. WordNet) into our system yet, for a proper way of integration is still up for future research.

Acknowledgment

The first author is funded by the PIRE PhD scholarship program and the EuroMatrixPlus project (IST-231720) which is funded by the European Commission under the Seventh Framework Programme. The second author thanks the German Excellence Cluster of Multimodal Computing and Interaction for the support of the work. Many thanks to Hans Uszkoreit for the useful discussion and Sebastian Riedel for the tool to visualize dependency trees/graphs.

References

1. Wang, R., Neumann, G.: An accuracy-oriented divide-and-conquer strategy for recognizing textual entailment. In: Proceedings of TAC Workshop. (2009)
2. Cabrio, E.: Specialized entailment engines: Approaching linguistic aspects of textual entailment. In: Natural Language Processing and Information Systems. Springer (2010) 305–308
3. Szpektor, I., Shnarch, E., Dagan, I.: Instance-based evaluation of entailment rule acquisition. In: Proceedings of ACL. (2007) 456–463
4. Szpektor, I., Dagan, I.: Learning entailment rules for unary templates. In: Proceedings of COLING 2008, Manchester, UK (2008) 849–856
5. Bos, J., Markert, K.: Recognising textual entailment with logical inference. In: Proceedings of HLT-EMNLP 2005. (2005) 628–635
6. de Marneffe, M.C., Rafferty, A.N., Manning, C.D.: Finding contradictions in text. In: Proceedings of ACL-08: HLT. (2008)
7. Bar-Haim, R., Berant, J., Dagan, I.: A compact forest for scalable inference over entailment and paraphrase rules. In: Proceedings of EMNLP. (2009)
8. Heilman, M., Smith, N.A.: Tree edit models for recognizing textual entailments, paraphrases, and answers to questions. In: Proceedings of NAACL-HLT. (2010) 1011–1019
9. Murakami, K., Masuda, S., Matsuyoshi, S., Nichols, E., Inui, K., Matsumoto, Y.: Annotating semantic relations combining facts and opinions. In: ACL-IJCNLP '09: Proceedings of the Third Linguistic Annotation Workshop. (2009)
10. Wang, R., Zhang, Y.: Recognizing textual relatedness with predicate-argument structures. In: Proceedings of Conference on Empirical Methods in Natural Language Processing (EMNLP 2009), Singapore, Singapore, Association for Computational Linguistics (2009)
11. MacCartney, B., Manning, C.D.: Natural logic for textual inference. In: Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing. (2007) 193–200
12. Wang, R., Sporleder, C.: Constructing a textual semantic relation corpus using a discourse treebank. In: Proceedings of the seventh international conference on Language Resources and Evaluation (LREC), Valletta, Malta (2010)
13. Harabagiu, S., Hickl, A., Lacatusu, F.: Negation, contrast and contradiction in text processing. In: Proceedings of AAAI. (2006)

14. Padó, S., Cer, D., Galley, M., Jurafsky, D., Manning, C.D.: Measuring machine translation quality as semantic equivalence: A metric based on entailment features. Machine Translation **23**(2–3) (2009) 181–193
15. Dagan, I., Glickman, O., Magnini, B.: The pascal recognising textual entailment challenge. In: Machine Learning Challenges. Lecture Notes in Computer Science. Springer (2006)
16. Bentivogli, L., Magnini, B., Dagan, I., Dang, H., Giampiccolo, D.: The fifth pascal recognizing textual entailment challenge. In: Proceedings of the Text Analysis Conference (TAC 2009) Workshop. (2009)
17. Hickl, A., Williams, J., Bensley, J., Roberts, K., Rink, B., Shi, Y.: Recognizing textual entailment with LCC's groundhog system. In: Proceedings of the RTE-2 Workshop. (2006)
18. Tatu, M., Moldovan, D.: A logic-based semantic approach to recognizing textual entailment. In: Proceedings of the COLING/ACL on Main conference poster sessions. (2006) 819–826
19. Lin, D., Pantel, P.: Dirt - discovery of inference rules from text. In: In Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining. (2001) 323–328
20. Dinu, G., Wang, R.: Inference rules and their application to recognizing textual entailment. In: Proceedings of EACL. (2009) 211–219
21. Kouylekov, M., Magnini, B.: Recognizing textual entailment with tree edit distance algorithms. In: Proceedings of the PASCAL Challenges on RTE. (2005) 17–20
22. Haghighi, A., Ng, A., Manning, C.: Robust textual inference via graph matching. In: Proceedings of HLT-EMNLP 2005. (2005) 387–394
23. Zanzotto, F.M., Moschitti, A.: Automatic learning of textual entailments with cross-pair similarities. In: Proceedings of the COLING/ACL, Sydney, Australia (2006) 401–408
24. Mehdad, Y., Moschitti, A., Zanzotto, F.M.: Syntactic/semantic structures for textual entailment recognition. In: Proceedings of NAACL-HLT. (2010) 1020–1028
25. Wang, R., Neumann, G.: Recognizing textual entailment using a subsequence kernel method. In: Proceedings of AAAI. (2007) 937–942
26. Cooper, R., Crouch, D., Eijck, J.V., Fox, C., Genabith, J.V., Jaspars, J., Kamp, H., Milward, D., Pinkal, M., Poesio, M., Pulman, S.: A framework for computational semantics (FraCaS). Technical report, The FraCaS Consortium (1996)
27. Barzilay, R., Lee, L.: Learning to paraphrase: an unsupervised approach using multiple-sequence alignment. In: Proceedings of NAACL-HLT. (2003)
28. McDonald, R., Pereira, F., Ribarov, K., Hajic, J.: Non-Projective Dependency Parsing using Spanning Tree Algorithms. In: Proceedings of HLT-EMNLP 2005. (2005) 523–530
29. Nivre, J., Nilsson, J., Hall, J., Chanev, A., Eryigit, G., Kübler, S., Marinov, S., Marsi, E.: Maltparser: A language-independent system for data-driven dependency parsing. Natural Language Engineering **13**(1) (2007) 1–41
30. Surdeanu, M., Johansson, R., Meyers, A., Màrquez, L., Nivre, J.: The CoNLL-2008 shared task on joint parsing of syntactic and semantic dependencies. In: Proceedings of CoNLL-2008. (2008)
31. Wang, R., Callison-Burch, C.: Cheap facts and counter-facts. In: Proceedings of NAACL-HLT 2010 Workshop on Amazon Mechanical Turk, Los Angeles, California (2010)
32. Snow, R., O'Connor, B., Jurafsky, D., Ng, A.Y.: Cheap and fast - but is it good? Evaluating non-expert annotations for natural language tasks. In: Proceedings of EMNLP. (2008)
33. Dolan, W.B., Brockett, C.: Automatically constructing a corpus of sentential paraphrases. In: Proceedings of the IWP2005. (2005)
34. Yuret, D., Han, A., Turgut, Z.: Semeval-2010 task 12: Parser evaluation using textual entailments. In: Proceedings of the SemEval-2010 Evaluation Exercises on Semantic Evaluation. (2010)
35. Das, D., Smith, N.A.: Paraphrase identification as probabilistic quasi-synchronous recognition. In: Proceedings of ACL-IJCNLP 2009. (2009)