Inference Rules and their Application to Recognizing Textual Entailment

Georgiana Dinu
Saarland University
Campus, D-66123 Saarbrücken
dinu@coli.uni-sb.de

Rui Wang
Saarland University
Campus, D-66123 Saarbrücken
rwang@coli.uni-sb.de

Abstract

In this paper, we explore ways of improving an inference rule collection and its application to the task of recognizing textual entailment. For this purpose, we start with an automatically acquired collection and we propose methods to refine it and obtain more rules using a hand-crafted lexical resource. Following this, we derive a dependency-based structure representation from texts, which aims to provide a proper base for the inference rule application. The evaluation of our approach on the recognizing textual entailment data shows promising results on precision and the error analysis suggests possible improvements.

1 Introduction

Textual inference plays an important role in many natural language processing (NLP) tasks. In recent years, the recognizing textual entailment (RTE) (Dagan et al., 2006) challenge, which focuses on detecting semantic inference, has attracted a lot of attention. Given a text **T** (several sentences) and a hypothesis **H** (one sentence), the goal is to detect if **H** can be inferred from **T**.

Studies such as (Clark et al., 2007) attest that lexical substitution (e.g. synonyms, antonyms) or simple syntactic variation account for the entailment only in a small number of pairs. Thus, one essential issue is to identify more complex expressions which, in appropriate contexts, convey the same (or similar) meaning. However, more generally, we are also interested in pairs of expressions in which only a uni-directional inference relation holds[1].

A typical example is the following RTE pair in which *accelerate to* in **H** is used as an alternative formulation for *reach speed of* in **T**.

T: *The high-speed train, scheduled for a trial run on Tuesday, is able to* ***reach*** *a maximum* ***speed of*** *up to 430 kilometers per hour, or 119 meters per second.*

H: *The train* ***accelerates*** *to 430 kilometers per hour.*

One way to deal with textual inference is through rule representation, for example *X wrote Y* ≈ *X is author of Y*. However, manually building collections of inference rules is time-consuming and it is unlikely that humans can exhaustively enumerate all the rules encoding the knowledge needed in reasoning with natural language. Instead, an alternative is to acquire these rules automatically from large corpora. Given such a rule collection, the next step to focus on is how to successfully use it in NLP applications. This paper tackles both aspects, acquiring inference rules and using them for the task of recognizing textual entailment.

For the first aspect, we extend and refine an existing collection of inference rules acquired based on the *Distributional Hypothesis* (DH). One of the main advantages of using the DH is that the only input needed is a large corpus of (parsed) text[2]. For the extension and refinement, a hand-crafted lexical resource is used for augmenting the original inference rule collection and exclude some of the incorrect rules.

For the second aspect, we focus on applying these rules to the RTE task. In particular, we use a structure representation derived from the dependency parse trees of **T** and **H**, which aims to capture the essential information they convey.

The rest of the paper is organized as follows: Section 2 introduces the inference rule collection

[1] We will use the term inference rule to stand for such concept; the two expressions can be actual paraphrases if the relation is bi-directional

[2] Another line of work on acquiring paraphrases uses comparable corpora, for instance (Barzilay and McKeown, 2001), (Pang et al., 2003)

Proceedings of the 12th Conference of the European Chapter of the ACL, pages 211–219,
Athens, Greece, 30 March – 3 April 2009. ©2009 Association for Computational Linguistics

we use, based on the Discovery of Inference Rules from Text (henceforth DIRT) algorithm and discusses previous work on applying it to the RTE task. Section 3 focuses on the rule collection itself and on the methods in which we use an external lexical resource to extend and refine it. Section 4 discusses the application of the rules for the RTE data, describing the structure representation we use to identify the appropriate context for the rule application. The experimental results will be presented in Section 5, followed by an error analysis and discussions in Section 6. Finally Section 7 will conclude the paper and point out future work directions.

2 Background

A number of automatically acquired inference rule/paraphrase collections are available, such as (Szpektor et al., 2004), (Sekine, 2005). In our work we use the DIRT collection because it is the largest one available and it has a relatively good accuracy (in the 50% range for top generated paraphrases, (Szpektor et al., 2007)). In this section, we describe the DIRT algorithm for acquiring inference rules. Following that, we will overview the RTE systems which take DIRT as an external knowledge resource.

2.1 Discovery of Inference Rules from Text

The DIRT algorithm has been introduced by (Lin and Pantel, 2001) and it is based on what is called the *Extended Distributional Hypothesis*. The original DH states that *words* occurring in similar contexts have similar meaning, whereas the extended version hypothesizes that *phrases* occurring in similar contexts are similar.

An inference rule in DIRT is a pair of binary relations $\langle\ pattern_1(X,Y),\ pattern_2(X,Y)\ \rangle$ which stand in an inference relation. $pattern_1$ and $pattern_2$ are chains in dependency trees[3] while X and Y are placeholders for nouns at the end of this chain. The two patterns will constitute a candidate paraphrase if the sets of X and Y values exhibit relevant overlap. In the following example, the two patterns are *prevent* and *provide protection against.*

$$\mathbf{X} \xleftarrow{subj} prevent \xrightarrow{obj} \mathbf{Y}$$

$$\mathbf{X} \xleftarrow{subj} provide \xrightarrow{obj} protection \xrightarrow{mod} against \xrightarrow{pcomp} \mathbf{Y}$$

X put emphasis on Y
≈ *X pay attention to Y*
≈ *X attach importance to Y*
≈ *X increase spending on Y*
≈ *X place emphasis on Y*
≈ *Y priority of X*
≈ *X focus on Y*

Table 1: Example of DIRT algorithm output. Most confident paraphrases of *X put emphasis on Y*

Such rules can be informally defined (Szpektor et al., 2007) as directional relations between two text patterns with variables. The left-hand-side pattern is assumed to entail the right-hand-side pattern in certain contexts, under the same variable instantiation. The definition relaxes the intuition of inference, as we only require the entailment to hold in *some* and not *all* contexts, motivated by the fact that such inferences occur often in natural text.

The algorithm does not extract directional inference rules, it can only identify candidate paraphrases; many of the rules are however unidirectional. Besides syntactic rewriting or lexical rules, rules in which the patterns are rather complex phrases are also extracted. Some of the rules encode lexical relations which can also be found in resources such as WordNet while others are lexical-syntactic variations that are unlikely to occur in hand-crafted resources (Lin and Pantel, 2001). Table 1 gives a few examples of rules present in DIRT[4].

Current work on inference rules focuses on making such resources more precise. (Basili et al., 2007) and (Szpektor et al., 2008) propose attaching selectional preferences to inference rules. These are semantic classes which correspond to the anchor values of an inference rule and have the role of making precise the context in which the rule can be applied [5]. This aspect is very important and we plan to address it in our future work. However in this paper we investigate the first and more basic issue: how to successfully use rules in their current form.

[3] obtained with the Minipar parser (Lin, 1998)

[4] For simplification, in the rest of the paper we will omit giving the dependency relations in a pattern.

[5] For example *X won Y* entails *X played Y* only when Y refers to some sort of competition, but not if Y refers to a musical instrument.

2.2 Related Work

Intuitively such inference rules should be effective for recognizing textual entailment. However, only a small number of systems have used DIRT as a resource in the RTE-3 challenge, and the experimental results have not fully shown it has an important contribution.

In (Clark et al., 2007)'s approach, semantic parsing to clause representation is performed and true entailment is decided only if every clause in the semantic representation of **T** semantically matches some clause in **H**. The only variation allowed consists of rewritings derived from WordNet and DIRT. Given the preliminary stage of this system, the overall results show very low improvement over a random classification baseline.

(Bar-Haim et al., 2007) implement a proof system using rules for generic linguistic structures, lexical-based rules, and lexical-syntactic rules (these obtained with a DIRT-like algorithm on the first CD of the Reuters RCV1 corpus). The entailment considers not only the strict notion of proof but also an approximate one. Given premise p and hypothesis h, the lexical-syntactic component marks all lexical noun alignments. For every pair of alignment, the paths between the two nouns are extracted, and the DIRT algorithm is applied to obtain a similarity score. If the score is above a threshold the rule is applied. However these lexical-syntactic rules are only used in about 3% of the attempted proofs and in most cases there is no lexical variation.

(Iftene and Balahur-Dobrescu, 2007) use DIRT in a more relaxed manner. A DIRT rule is employed in the system if at least one of the anchors match in **T** and **H**, i.e. they use them as unary rules. However, the detailed analysis of the system that they provide shows that the DIRT component is the least relevant one (adding 0.4% of precision).

In (Marsi et al., 2007), the focus is on the usefulness of DIRT. In their system a paraphrase substitution step is added on top of a system based on a tree alignment algorithm. The basic paraphrase substitution method follows three steps. Initially, the two patterns of a rule are matched in **T** and **H** (instantiations of the anchors X, Y do not have to match). The text tree is transformed by applying the paraphrase substitution. Following this, the transformed text tree and hypothesis trees are aligned. The coverage (proportion of aligned content words) is computed and if above some threshold, entailment is true. The paraphrase component adds 1.0% to development set results and only 0.5% to test sets, but a more detailed analysis on the results of the interaction with the other system components is not given.

X write Y →X author Y
X, founded in Y →X, opened in Y
X launch Y → X produce Y
X represent Z → X work for Y
death relieved X → X died
X faces menace from Y ↔ X endangered by Y
X, peace agreement for Y *→ X is formulated to end war in Y*

Table 2: Example of inference rules needed in RTE

3 Extending and refining DIRT

Based on observations of using the inference rule collection on the real data, we discover that 1) some of the needed rules still lack even in a very large collection such as DIRT and 2) some systematic errors in the collection can be excluded. On both aspects, we use WordNet as additional lexical resource.

Missing Rules

A closer look into the RTE data reveals that DIRT lacks many of the rules that entailment pairs require.

Table 2 lists a selection of such rules. The first rows contain rules which are structurally very simple. These, however, are missing from DIRT and most of them also from other hand-crafted resources such as WordNet (i.e. there is no short path connecting the two verbs). This is to be expected as they are rules which hold in specific contexts, but difficult to be captured by a sense distinction of the lexical items involved.

The more complex rules are even more difficult to capture with a DIRT-like algorithm. Some of these do not occur frequently enough even in large amounts of text to permit acquiring them via the DH.

Combining WordNet and DIRT

In order to address the issue of missing rules, we investigate the effects of combining DIRT with an exact hand-coded lexical resource in order to create new rules.

For this we extended the DIRT rules by adding

X face threat of Y	≈ *X at risk of Y*
face	≈ *confront, front, look, face up*
threat	≈ *menace, terror, scourge*
risk	≈ *danger, hazard, jeopardy, endangerment, peril*

Table 3: Lexical variations creating new rules based on DIRT rule *X face threat of Y* → *X at risk of Y*

rules in which any of the lexical items involved in the patterns can be replaced by WordNet synonyms. In the example above, we consider the DIRT rule *X face threat of Y* → *X, at risk of Y* (Table 3).

Of course at this moment due to the lack of sense disambiguation, our method introduces lots of rules that are not correct. As one can see, expressions such as *front scourge* do not make any sense, therefore any rules containing this will be incorrect. However some of the new rules created in this example, such as *X face threat of Y* ≈ *X, at danger of Y* are reasonable ones and the rules which are incorrect often contain patterns that are very unlikely to occur in natural text.

The idea behind this is that a combination of various lexical resources is needed in order to cover the vast variety of phrases which humans can judge to be in an inference relation.

The method just described allows us to identify the first four rules listed in Table 2. We also acquire the rule *X face menace of Y* ≈ *X endangered by Y* (via *X face threat of Y* ≈ *X threatened by Y*, $menace \approx threat$, $threaten \approx endanger$).

Our extension is application-oriented therefore it is not intended to be evaluated as an independent rule collection, but in an application scenario such as RTE (Section 6).

In our experiments we also made a step towards removing the most systematic errors present in DIRT. DH algorithms have the main disadvantage that not only phrases with the same meaning are extracted but also phrases with opposite meaning.

In order to overcome this problem and since such errors are relatively easy to detect, we applied a filter to the DIRT rules. This eliminates inference rules which contain WordNet antonyms. For such a rule to be eliminated the two patterns have to be identical (with respect to edge labels and content words) except from the antonymous words; an example of a rule eliminated this way is *X **have** confidence in Y* ≈ *X **lack** confidence in Y*.

As pointed out by (Szpektor et al., 2007) a thorough evaluation of a rule collection is not a trivial task; however due to our methodology we can assume that the percentage of rules eliminated this way that are indeed contradictions gets close to 100%.

4 Applying DIRT on RTE

In this section we point out two issues that are encountered when applying inference rules for textual entailment. The first issue is concerned with correctly identifying the pairs in which the knowledge encoded in these rules is needed. Following this, another non-trivial task is to determine the way this knowledge interacts with the rest of information conveyed in an entailment pair. In order to further investigate these issues, we apply the rule collection on a dependency-based representation of text and hypothesis, namely Tree Skeleton.

4.1 Observations

A straightforward experiment can reveal the number of pairs in the RTE data which contain rules present in DIRT. For all the experiments in this paper, we use the DIRT collection provided by (Lin and Pantel, 2001), derived from the DIRT algorithm applied on 1GB of news text. The results we report here use only the most confident rules amounting to more than 4 million rules (top 40 following (Lin and Pantel, 2001)).[6]

Following the definition of an entailment rule, we identify RTE pairs in which $pattern_1(w1, w2)$ and $pattern_2(w1, w2)$ are matched one in **T** and the other one in **H** and $\langle pattern_1(X, Y), pattern2(X, Y)\rangle$ is an inference rule. The pair bellow is an example of this.

T: *The sale was made to pay Yukos US$ 27.5 billion tax bill, Yuganskneftegaz was originally sold for US$ 9.4 billion to a little known company* ***Baikalfinansgroup*** *which was later* ***bought by*** *the Russian state-owned oil company* ***Rosneft.***

H: ***Baikalfinansgroup*** *was* ***sold to Rosneft.***

[6] Another set of experiments showed that for this particular task, using the entire collection instead of a subset gave similar results.

On average, only 2% of the pairs in the RTE data is subject to the application of such inference rules. Out of these, approximately 50% are lexical rules (one verb entailing the other). Out of these lexical rules, around 50% are present in WordNet in a synonym, hypernym or sister relation. At a manual analysis, close to 80% of these are correct rules; this is higher than the estimated accuracy of DIRT, probably due to the bias of the data which consists of pairs which are entailment candidates.

However, given the small number of inference rules identified this way, we performed another analysis. This aims at determining an upper bound of the number of pairs featuring entailment phrases present in a collection. Given DIRT and the RTE data, we compute in how many pairs the two patterns of a paraphrase can be matched irrespective of their anchor values. An example is the following pair,

T: *Libya's case against Britain and the US* ***concerns*** *the dispute over their demand for extradition of Libyans charged with blowing up a Pan Am jet over Lockerbie in 1988.*

H: *One case* ***involved*** *the extradition of Libyan suspects in the Pan Am Lockerbie bombing.*

This is a case in which the rule is correct and the entailment is positive. In order to determine this, a system will have to know that *Libya's case against Britain and the US* in **T** entails *one case* in **H**. Similarly, in this context, *the dispute over their demand for extradition of Libyans charged with blowing up a Pan Am jet over Lockerbie in 1988* in **T** can be replaced with *the extradition of Libyan suspects in the Pan Am Lockerbie bombing* preserving the meaning.

Altogether in around 20% of the pairs, patterns of a rule can be found this way, many times with more than one rule found in a pair. However, in many of these pairs, finding the patterns of an inference rule does not imply that the rule is truly present in that pair.

Considering a system is capable of correctly identifying the cases in which an inference rule is needed, subsequent issues arise from the way these fragments of text interact with the surrounding context. Assuming we have a correct rule present in an entailment pair, the cases in which the pair is still not a positive case of entailment can be summarized as follows:

- The entailment rule is present in parts of the text which are not relevant to the entailment value of the pair.
- The rule is relevant, however the sentences in which the patterns are embedded block the entailment (e.g. through negative markers, modifiers, embedding verbs not preserving entailment)[7]
- The rule is correct in a limited number of contexts, but the current context is not the correct one.

To sum up, making use of the knowledge encoded with such rules is not a trivial task. If rules are used strictly in concordance with their definition, their utility is limited to a very small number of entailment pairs. For this reason, 1) instead of forcing the anchor values to be identical as most previous work, we allow more flexible rule matching (similar to (Marsi et al., 2007)) and 2) furthermore, we control the rule application process using a text representation based on dependency structure.

4.2 Tree Skeleton

The Tree Skeleton (TS) structure was proposed by (Wang and Neumann, 2007), and can be viewed as an extended version of the predicate-argument structure. Since it contains not only the predicate and its arguments, but also the dependency paths in-between, it captures the essential part of the sentence.

Following their algorithm, we first preprocess the data using a dependency parser[8] and then select overlapping topic words (i.e. nouns) in **T** and **H**. By doing so, we use fuzzy match at the substring level instead of full match. Starting with these nouns, we traverse the dependency tree to identify the lowest common ancestor node (named as *root node*). This sub-tree without the inner yield is defined as a Tree Skeleton. Figure 1 shows the TS of **T** of the following positive example,

T *For their discovery of ulcer-causing bacteria, Australian doctors Robin Warren and Barry Marshall have received the 2005 Nobel Prize in Physiology or Medicine.*

H *Robin Warren was awarded a Nobel Prize.*

Notice that, in order to match the inference rules with two anchors, the number of the dependency

[7] See (Nairn et al., 2006) for a detailed analysis of these aspects.

[8] Here we also use Minipar for the reason of consistence

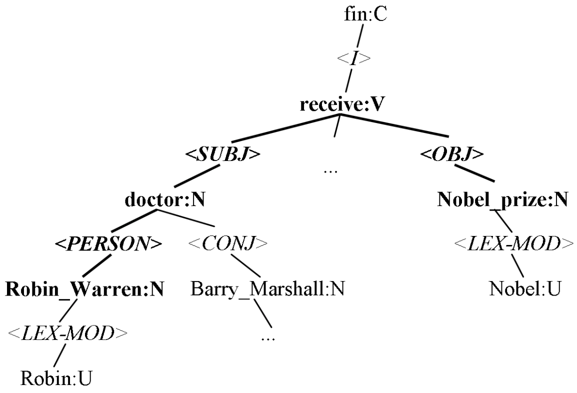

Figure 1: Dependency structure of text. Tree skeleton in bold

paths contained in a TS should also be two. In practice, among all the 800 T-H pairs of the RTE-2 test set, we successfully extracted tree skeletons in 296 text pairs, i.e., 37% of the test data is covered by this step and results on other data sets are similar.

Applying DIRT on a TS

Dependency representations like the tree skeleton have been explored by many researchers, e.g. (Zanzotto and Moschitti, 2006) have utilized a tree kernel method to calculate the similarity between T and H, and (Wang and Neumann, 2007) chose subsequence kernel to reduce the computational complexity. However, the focus of this paper is to evaluate the application of inference rules on RTE, instead of exploring methods of tackling the task itself. Therefore, we performed a straightforward matching algorithm to apply the inference rules on top of the tree skeleton structure. Given tree skeletons of **T** and **H**, we check if the two left dependency paths, the two right ones or the two root nodes contain the patterns of a rule.

In the example above, the rule $X \xleftarrow{obj} receive \xrightarrow{subj} Y \approx X \xleftarrow{obj2} award \xrightarrow{obj1} Y$ satisfies this criterion, as it is matched at the root nodes. Notice that the rule is correct only in restricted contexts, in which the object of *receive* is something which is conferred on the basis of merit. However in this pair, the context is indeed the correct one.

5 Experiments

Our experiments consist in predicting positive entailment in a very straightforward rule-based manner (Table 4 summarizes the results using three different rule collections). For each collection we select the RTE pairs in which we find a tree skeleton and match an inference rule. The first number in our table entries represents how many of such pairs we have identified, out the 1600 of development and test pairs. For these pairs we simply predict positive entailment and the second entry represents what percentage of these pairs are indeed positive entailment. Our work does not focus on building a complete RTE system; however, we also combine our method with a bag of words baseline to see the effects on the whole data set.

5.1 Results on a subset of the data

In the first two columns (Dirt$_{TS}$ and Dirt+WN$_{TS}$) we consider DIRT in its original state and DIRT with rules generated with WordNet as described in Section 3; all precisions are higher than 67%[9]. After adding WordNet, approximately in twice as many pairs, tree skeletons and rules are matched, while the precision is not harmed. This may indicate that our method of adding rules does not decrease precision of an RTE system.

In the third column we report the results of using a set of rules containing only the trivial identity ones (Id$_{TS}$). For our current system, this can be seen as a $precision$ upper bound for all the other collections, in concordance with the fact that identical rules are nothing but inference rules of highest possible confidence. The fourth column (Dirt+Id+WN$_{TS}$) contains what can be considered our best setting. In this setting considerably more pairs are covered using a collection containing DIRT and identity rules with WordNet extension.

Although the precision results with this setting are encouraging (65% for RTE2 data and 72% for RTE3 data), the coverage is still low, 8% for RTE2 and 6% for RTE3. This aspect together with an error analysis we performed are the focus of Section 7.

The last column (Dirt+Id+WN) gives the precision we obtain if we simply decide a pair is true entailment if we have an inference rule matched in it (irrespective of the values of the anchors or of the existence of tree skeletons). As expected, only identifying the patterns of a rule in a pair irrespective of tree skeletons does not give any indication of the entailment value of the pair.

[9] The RTE task is considered to be difficult. The average accuracy of the systems in the RTE-3 challenge is around 61% (Giampiccolo et al., 2007)

RTE Set	Dirt$_{TS}$	Dirt + WN$_{TS}$	Id$_{TS}$	Dirt + Id + WN$_{TS}$	Dirt + Id + WN
RTE2	49/69.38	94/67.02	45/66.66	130/65.38	673/50.07
RTE3	42/69.04	70/70.00	29/79.31	93/72.05	661/55.06

Table 4: Coverage/precision with various rule collections

RTE Set	BoW	Main
RTE2 (85 pairs)	51.76%	60.00%
RTE3 (64 pairs)	54.68%	62.50%

Table 5: Precision on the covered RTE data

RTE Set (800 pairs)	BoW	Main & BoW
RTE2	56.87%	57.75%
RTE3	61.12%	61.75%

Table 6: Precision on full RTE data

5.2 Results on the entire data

At last, we also integrate our method with a bag of words baseline, which calculates the ratio of overlapping words in **T** and **H**. For the pairs that our method covers, we overrule the baseline's decision. The results are shown in Table 6 (Main stands for the Dirt + Id + WN$_{TS}$ configuration). On the full data set, the improvement is still small due to the low coverage of our method, however on the pairs that are covered by our method (Table 5), there is a significant improvement over the overlap baseline.

6 Discussion

In this section we take a closer look at the data in order to better understand how does our method of combining tree skeletons and inference rules work. We will first perform error analysis on what we have considered our best setting so far. Following this, we analyze data to identify the main reasons which cause the low coverage.

For error analysis we consider the pairs incorrectly classified in the RTE3 test data set, consisting of a total of 25 pairs. We classify the errors into three main categories: rule application errors, inference rule errors, and other errors (Table 7).

In the first category, the tree skeleton fails to match the corresponding anchors of the inference rules. For instance, if someone founded *the Institute of Mathematics (Instituto di Matematica) at the University of Milan*, it does not follow that they founded *The University of Milan*. The *Institute of Mathematics* should be aligned with the *University of Milan*, which should avoid applying the inference rule for this pair.

A rather small portion of the errors (16%) are caused by incorrect inference rules. Out of these, two are correct in some contexts but not in the entailment pairs in which they are found. For example, the following rule *X generate Y* $\approx$ *X earn Y* is used incorrectly, however in the restricted context of $money$ or $income$, the two verbs have similar meaning. An example of an incorrect rule is *X issue Y* $\approx$ *X hit Y* since it is difficult to find a context in which this holds.

The last category contains all the other errors. In all these cases, the additional information conveyed by the text or the hypothesis which cannot be captured by our current approach, affects the entailment. For example *an imitation diamond* is not a $diamond$, and *more than 1,000 members of the Russian and foreign media* does not entail *more than 1,000 members from Russia*; these are not trivial, since lexical semantics and fine-grained analysis of the restrictors are needed.

For the second part of our analysis we discuss the coverage issue, based on an analysis of uncovered pairs. A main factor in failing to detect pairs in which entailment rules should be applied is the fact that the tree skeleton does not find the corresponding lexical items of two rule patterns.

Issues will occur even if the tree skeleton structure is modified to align all the corresponding fragments together. Consider cases such as *threaten to boycott* and *boycott* or similar constructions with other embedding verbs such as *manage*, *forget*, *attempt*. Our method can detect if the two embedded verbs convey a similar meaning, however not how the embedding verbs affect the implication.

Independent of the shortcomings of our tree skeleton structure, a second factor in failing to detect true entailment still lies in lack of rules. For instance, the last two examples in Table 2 are entailment pair fragments which can be formulated as inference rules, but it is not straightforward to acquire them via the DH.

Source of error	% pairs
Incorrect rule application	32%
Incorrect inference rules	16%
Other errors	52%

Table 7: Error analysis

7 Conclusion

Throughout the paper we have identified important issues encountered in using inference rules for textual entailment and proposed methods to solve them. We explored the possibility of combining a collection obtained in a statistical, unsupervised manner, DIRT, with a hand-crafted lexical resource in order to make inference rules have a larger contribution to applications. We also investigated ways of effectively applying these rules. The experiment results show that although coverage is still not satisfying, the precision is promising. Therefore our method has the potential to be successfully integrated in a larger entailment detection framework.

The error analysis points out several possible future directions. The tree skeleton representation we used needs to be enhanced in order to capture more accurately the relevant fragments of the text. A different issue remains the fact that a lot of rules we could use for textual entailment detection are still lacking. A proper study of the limitations of the DH as well as a classification of the knowledge we want to encode as inference rules would be a step forward towards solving this problem.

Furthermore, although all the inference rules we used aim at recognizing positive entailment cases, it is natural to use them for detecting negative cases of entailment as well. In general, we can identify pairs in which the patterns of an inference rule are present but the anchors are mismatched, or they are not the correct hypernym/hyponym relation. This can be the base of a principled method for detecting structural contradictions (de Marneffe et al., 2008).

8 Acknowledgments

We thank Dekang Lin and Patrick Pantel for providing the DIRT collection and to Grzegorz Chrupała, Alexander Koller, Manfred Pinkal and Stefan Thater for very useful discussions. Georgiana Dinu and Rui Wang are funded by the IRTG and PIRE PhD scholarship programs.

References

Roy Bar-Haim, Ido Dagan, Iddo Greental, Idan Szpektor, and Moshe Friedman. 2007. Semantic inference at the lexical-syntactic level for textual entailment recognition. In *Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*, pages 131–136, Prague, June. Association for Computational Linguistics.

Regina Barzilay and Kathleen R. McKeown. 2001. Extracting paraphrases from a parallel corpus. In *Proceedings of 39th Annual Meeting of the Association for Computational Linguistics*, pages 50–57, Toulouse, France, July. Association for Computational Linguistics.

Roberto Basili, Diego De Cao, Paolo Marocco, and Marco Pennacchiotti. 2007. Learning selectional preferences for entailment or paraphrasing rules. In *In Proceedings of RANLP*, Borovets, Bulgaria.

Peter Clark, Phil Harrison, John Thompson, William Murray, Jerry Hobbs, and Christiane Fellbaum. 2007. On the role of lexical and world knowledge in rte3. In *Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*, pages 54–59, Prague, June. Association for Computational Linguistics.

Ido Dagan, Oren Glickman, and Bernardo Magnini. 2006. The pascal recognising textual entailment challenge. In *Lecture Notes in Computer Science, Vol. 3944, Springer*, pages 177–190. Quionero-Candela, J.; Dagan, I.; Magnini, B.; d'Alch-Buc, F. Machine Learning Challenges.

Marie-Catherine de Marneffe, Anna N. Rafferty, and Christopher D. Manning. 2008. Finding contradictions in text. In *Proceedings of ACL-08: HLT*, pages 1039–1047, Columbus, Ohio, June. Association for Computational Linguistics.

Danilo Giampiccolo, Bernardo Magnini, Ido Dagan, and Bill Dolan. 2007. The third pascal recognizing textual entailment challenge. In *Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*, pages 1–9, Prague, June. Association for Computational Linguistics.

Adrian Iftene and Alexandra Balahur-Dobrescu. 2007. Hypothesis transformation and semantic variability rules used in recognizing textual entailment. In *Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*, pages 125–130, Prague, June. Association for Computational Linguistics.

Dekang Lin and Patrick Pantel. 2001. Dirt. discovery of inference rules from text. In *KDD '01: Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 323–328, New York, NY, USA. ACM.

Dekang Lin. 1998. Dependency-based evaluation of minipar. In *Proc. Workshop on the Evaluation of Parsing Systems*, Granada.

Erwin Marsi, Emiel Krahmer, and Wauter Bosma. 2007. Dependency-based paraphrasing for recognizing textual entailment. In *Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*, pages 83–88, Prague, June. Association for Computational Linguistics.

Rowan Nairn, Cleo Condoravdi, and Lauri Karttunen. 2006. Computing relative polarity for textual inference. In *Proceedings of ICoS-5 (Inference in Computational Semantics*, Buxton, UK.

Bo Pang, Kevin Knight, and Daniel Marcu. 2003. Syntax-based alignment of multiple translations: Extracting paraphrases and generating new sentences. In *HLT-NAACL*, pages 102–109.

Satoshi Sekine. 2005. Automatic paraphrase discovery based on context and keywords between NE pairs. In *Proceedings of International Workshop on Paraphrase*, pages 80–87, Jeju Island, Korea.

Idan Szpektor, Hristo Tanev, Ido Dagan, and Bonaventura Coppola. 2004. Scaling web-based acquisition of entailment relations. In *In Proceedings of EMNLP*, pages 41–48.

Idan Szpektor, Eyal Shnarch, and Ido Dagan. 2007. Instance-based evaluation of entailment rule acquisition. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 456–463, Prague, Czech Republic, June. Association for Computational Linguistics.

Idan Szpektor, Ido Dagan, Roy Bar-Haim, and Jacob Goldberger. 2008. Contextual preferences. In *Proceedings of ACL-08: HLT*, pages 683–691, Columbus, Ohio, June. Association for Computational Linguistics.

Rui Wang and Günter Neumann. 2007. Recognizing textual entailment using sentence similarity based on dependency tree skeletons. In *Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*, pages 36–41, Prague, June. Association for Computational Linguistics.

Fabio Massimo Zanzotto and Alessandro Moschitti. 2006. Automatic learning of textual entailments with cross-pair similarities. In *ACL-44: Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, pages 401–408, Morristown, NJ, USA. Association for Computational Linguistics.