Dependency-based paraphrasing for recognizing textual entailment

Erwin Marsi, Emiel Krahmer
Communication & Cognition
Tilburg University
The Netherlands
e.c.marsi@uvt.nl
e.j.krahmer@uvt.nl

Wauter Bosma
Human Media Interaction
University of Twente
The Netherlands
w.e.bosma@ewi.utwente.nl

Abstract

This paper addresses syntax-based paraphrasing methods for Recognizing Textual Entailment (RTE). In particular, we describe a dependency-based paraphrasing algorithm, using the DIRT data set, and its application in the context of a straightforward RTE system based on aligning dependency trees. We find a small positive effect of dependency-based paraphrasing on both the RTE3 development and test sets, but the added value of this type of paraphrasing deserves further analysis.

1 Introduction

Coping with paraphrases appears to be an essential subtask in Recognizing Textual Entailment (RTE). Most RTE systems incorporate some form of lexical paraphrasing, usually relying on WordNet to identify synonym, hypernym and hyponym relations among word pairs from text and hypothesis (Bar-Haim et al., 2006, Table 2). Many systems also address paraphrasing above the lexical level. This can take the form of identifying or substituting equivalent multi-word strings, e.g., (Bosma and Callison-Burch, 2006). A drawback of this approach is that it is hard to cope with discontinuous paraphrases containing one or more gaps. Other approaches exploit syntactic knowledge in the form of parse trees. Hand-crafted transformation rules can account for systematic syntactic alternation like active-passive form, e.g., (Marsi et al., 2006). Alternatively, such paraphrase rules may be automatically derived from huge text corpora (Lin and Pantel, 2001). There are at least two key advantages of syntax-based over string-based paraphrasing which are relevant for RTE: (1) it can cope with discontinuous paraphrases; (2) syntactic information such as dominance relations, phrasal syntactic labels and dependency relations, can be used to refine the coarse matching on words only.

Here we investigate paraphrasing on the basis of of syntactic dependency analyses. Our sole resource is the DIRT data set (Lin and Pantel, 2001), an extensive collection of automatically derived paraphrases. These have been used for RTE before (de Salvo Braz et al., 2005; Raina et al., 2005), and similar approaches to paraphrase mining have been applied as well (Nielsen et al., 2006; Hickl et al., 2006). However, in these approaches paraphrasing is always one factor in a complex system, and as a result little is known of the contribution of paraphrasing for the RTE task. In this paper, we focus entirely on dependency-based paraphrasing in order to get a better understanding of its usefulness for RTE. In the next Section, we describe the DIRT data and present an algorithm for dependency-based paraphrasing in order to bring a pair's text closer to its hypothesis. We present statistics on coverage as well as qualitative discussion of the results. Section 3 then describes our RTE system and results with and without dependency-based paraphrasing.

2 Dependency-based paraphrasing

2.1 Preprocessing RTE data

Starting from the text-hypothesis pairs in the RTE XML format, we first preprocess the data. As the text part may consist of more than one sentence, we first perform sentence splitting using Mxterminator (Reynar and Ratnaparkhi, 1997), a maximum

Proceedings of the Workshop on Textual Entailment and Paraphrasing, pages 83–88,

entropy-based end of sentence classifier trained on the Penn Treebank data. Next, each sentence is tokenized and syntactically parsed using the Minipar parser (Lin, 1998). From the parser's tabular output we extract the word forms, lemmas, part-of-speech tags and dependency relations. This information is then stored in an ad-hoc XML format which represents the trees as an hierarchy of node elements in order to facilitate tree matching.

2.2 DIRT data

The DIRT (Discovering Inference Rules from Text) method is based on extending Harris Distributional Hypothesis, which states that words that occurred in the same contexts tend to be similar, to dependency paths in parse trees (Lin and Pantel, 2001). Each dependency path consists of at least three nodes: a root node, and two non-root terminal nodes, which are nouns. The DIRT data set we used consists of over 182k paraphrase clusters derived from 1GB of newspaper text. Each cluster consists of a unique dependency path, which we will call the *paraphrase source*, and a list of equivalent dependency paths, which we will refer to as the *paraphrase translations*, ordered in decreasing value of point-wise mutual information. A small sample in the original format is

```
(N:by:V<buy>V:obj:N (sims
N:to:V<sell>V:obj:N      0.211704
N:subj:V<buy>V:obj:N     0.198728
...
))
```

The first two lines represent the inference rule: *X bought by Y* entails *X sold to Y*.

We preprocess the DIRT data by restoring prepositions, which were originally folded into a dependency relation, to individual nodes, as this eases alignment with the parsed RTE data. For the same reason, paths are converted to the same ad-hoc XML format as the parsed RTE data.

2.3 Paraphrase substitution

Conceptually, our paraphrase substitution algorithm takes a straightforward approach. For the purpose of explanation only, Figure 1 presents pseudo-code for a naive implementation. The main function takes two arguments (cf. line 1). The first is a preprocessed RTE data set in which all sentences from text and hypothesis are dependency parsed. The second is a collection of DIRT paraphrases, each one mapping a source path to one or more translation paths. For each text/hypothesis pair (cf. line 2), we look at all the subtrees of the text parses (cf. line 3-4) and attempt to find a suitable paraphrase of this subtree (cf. line 5). We search the DIRT paraphrases (cf. line 8) for a source path that matches the text subtree at hand (cf. line 9). If found, we check if any of the corresponding paraphrase translation paths (cf. line 10) matches a subtree of the hypothesis parse (cf. line 11-12). If so, we modify the text tree by substituting this translation path (cf. line 13). The intuition behind this is that we only accept paraphrases that bring the text closer to the hypothesis. The DIRT paraphrases are ordered in decreasing likelihood, so after a successful paraphrase substitution, we discard the remaining possibilities and continue with the next text subtree (cf. line 14).

The Match function, which is used for matching the source path to a text subtree and the translation path to an hypothesis subtree, requires the path to occur in the subtree. That is, all lemmas, part-of-speech tags and dependency relations from the path must have identical counterparts in the subtree; skipping nodes is not allowed. As the path's terminals specify no lemma, the only requirement is that their counterparts are nouns.

The Substitute function replaces the matched path in the text tree by the paraphrase's translation path. Intuitively, the path "overlays" a part of the subtree, changing lemmas and dependency relations, but leaving most of the daughter nodes unaffected. Note that the new path may be longer or shorter than the original one, thus introducing or removing nodes from the text tree.

As an example, we will trace our algorithm as applied to the first pair of the RTE3 dev set (id=1).

Text: *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.*
Hypothesis: *Baikalfinansgroup was sold to Rosneft.*
Entailment: *Yes*

While traversing the parse tree of the text, our algorithm encounters a node with POS tag V and lemma *buy*. The relevant part of the parse tree is shown at the right top of Figure 2. The logical arguments inferred by Minipar are shown between curly

```
def Paraphrase(parsed-rte-data, dirt-paraphrases):
    for pair in parsed-rte-data:
        for text-tree in pair.text-parses:
            for text-subtree in text-tree:
                 Paraphrase-subtree(text-subtree, dirt-paraphrases, pair.hyp-parse)

def Paraphrase-subtree(text-subtree, dirt-paraphrases, hyp-tree):
    for (source-path, translations) in dirt-paraphrases:
        if Match(source-path, text-subtree):
            for trans-path in translations:
                for hyp-subtree in hyp-tree:
                    if Match(trans-path, hyp-subtree):
                        text-subtree = Substitute(trans-path, text-subtree)
                        return
```

Figure 1: *Pseudo-code for a naive implementation of the dependency-based paraphrase substitution algorithm*

brackets, e.g., *US$ 9.4 billion*. For this combination of verb and lemma, the DIRT data contains 340 paraphrase sets, with a total of 26950 paraphrases. The algorithm starts searching for a paraphrase source which matches the text. It finds the path shown at the left top of Figure 2: *buy* with a PP modifier headed by preposition *by*, and a nominal object. This paraphrase source has 108 alternative translations. It searches for paraphrase translations which match the hypothesis. The first, and therefore most likely (probability is 0.22) path it finds is rooted in *sell*, with a PP-modifier headed by *to* and a nominal object. This translation path, as well as its alignment to the hypothesis parse tree, is shown in the middle part of Figure 2. Finally, the source path in the text tree is substituted by the translation path. The bottom part of Figure 2 shows the updated text tree as well as its improved alignment to the hypothesis tree. The paraphrasing procedure can in effect be viewed as making the inference that Baikalfinansgroup was bought by Rosneft, therefore Baikalfinansgroup was sold to Rosneft.

The naive implementation of the algorithm is of course not very efficient. Our actual implementation uses a number of shortcuts to reduce processing time. For instance, the DIRT paraphrases are indexed on the lemma of their root in order to speed up retrieval. As another example, text nodes with less than two child nodes (i.e. terminal and unary-branching nodes) are immediately skipped, as they will never match a paraphrase path.

2.4 Paraphrasing results

We applied our paraphrasing algorithm to the RTE3 development set. Table 1 gives an impression of how many paraphrases were substituted. The first row lists the total number of nodes in the dependency trees of the text parts. The second row shows that for roughly 15% of these nodes, the DIRT data contains a paraphrase with the same lemma. The next two rows show in how many cases the source path matches the text and the translation path matches the hypothesis (i.e. giving rise to a paraphrase substitution). Clearly, the number of actual paraphrase substitutions is relatively small: on average about 0.5% of all text subtrees are subject to paraphrasing. Still, about one in six sentences is subject to paraphrasing, and close to half of all pairs is paraphrased at least once. Sentences triggering more than one paraphrase do occur. Also note that paraphrasing occurs more frequently in true entailment pairs than in false entailment pairs. This is to be expected, given that text and hypothesis are more similar when an entailment relation holds.

2.5 Discussion on paraphrasing

Type of paraphrases A substantial number of the paraphrases applied are single word synonyms or verb plus particle combinations which might as well be obtained from string-based substitution on the basis of a lexical resource like WordNet. Some randomly chosen examples include *X announces Y* entails *X supports Y*, *X makes Y* entails *X sells Y*, and *locates X at Y, discovers X at Y*. Nevertheless, more interesting paraphrases do occur. In the pair below (id=452), we find the paraphrase *X wins Y* entails *X*

Table 1: *Frequency of (partial) paraphrase matches on the RTE3 dev set*

	IE:	IR:	QA:	SUM:	Total:
Text nodes:	8899	10610	10502	8196	38207
Matching paraphrase lemma:	1439	1724	1581	1429	6173
Matching paraphrase source:	566	584	543	518	2211
Matching paraphrase translation:	71	55	23	79	228
Text sentences:	272	350	306	229	1157
Paraphrased text sentences:	63	51	20	66	200
Paraphrased true-entailment pairs:	32	25	12	39	108
Paraphrased false-entailment pairs:	26	21	5	23	75

(is) Y champion.

Text: *Boris Becker is a true legend in the sport of tennis. Aged just seventeen, he won Wimbledon for the first time and went on to become the most prolific tennis player.*
Hypothesis: *Boris Becker is a Wimbledon champion.*
Entailment: *True*

Another intriguing paraphrase, which appears to be false on first sight, is *X flies from Y* entails *X makes (a) flight to Y*. However, in the context of the next pair (id=777), it turns out to be correct.

Text: *The Hercules transporter plane which flew straight here from the first round of the trip in Pakistan, touched down and it was just a brisk 100m stroll to the handshakes.*
Hypothesis: *The Hercules transporter plane made a flight to Pakistan.*
Entailment: *True*

Coverage Although the DIRT data constitutes a relatively large collection of paraphrases, it is clear that many paraphrases required for the RTE3 data are missing. We tried to improve coverage to some extent by relaxing the Match function: instead of an exact match, we allowed for small mismatches in POS tag and dependency relation, reversing the order of a path's left and right side, and even for skipping nodes. However, subjective evaluation suggested that the results deteriorated. Alternatively, the coverage might be increased by deducing paraphrases on the fly using the web as a corpus, e.g., (Hickl et al., 2006).

Somewhat surprisingly, the vast majority of paraphrases concerns verbs. Even though the DIRT data contains paraphrases for nouns, adjectives and complementizers, the coverage of these word classes is apparently not nearly as extensive as that of verbs.

Another observation is that fewer paraphrases occur in pairs from the QA task. We have no explanation for this.

False paraphrases Since the DIRT data was automatically derived and was not manually checked, it contains noise in the form of questionable or even false paraphrases. While some of these surface in paraphrased RTE3 data (e.g. *X leaves for Y* entails *X departs Y*, and *X feeds Y* entails *Y feeds X*), their number appears to be limited. We conjecture this is because of the double constraint that a paraphrase must match both text and hypothesis.

Relevance Not all paraphrase substitutions are relevant for the purpose of recognizing textual entailment. Evidently, paraphrases in false entailment pairs are counterproductive. However, even in true entailment pairs paraphrases might occur in parts of the text that are irrelevant to the task at hand. Consider the following pair from the RTE3 dev set (id=417).

Text: *When comparing Michele Granger and Brian Goodell, Brian has to be the clear winner. In 1976, while still a student at Mission Viejo High, Brian won two Olympic gold medals at Montreal, breaking his own world records in both the 400 - and 1,500 - meter freestyle events. He went on to win three gold medals in he 1979 Pan American Games.*
Hypothesis: *Brian Goodell won three gold medals in the 1979 Pan American Games.*
Entailment: *True*

The second text sentence and hypothesis match the paraphrases: (1) *X medal at Y* entails *X medal in Y*, and (2) *X record in Y* entails *X medal in Y*. Even so, virtually all of the important information is in the third text sentence.

3 Results on RTE3 data

Since our contribution focuses on syntactic paraphrasing, our RTE3 system is a simplified version

Table 2: *Percent accuracy on RTE3 set without paraphrasing (−) and with paraphrasing (+)*

Task	Dev−	Dev+	Test−	Test+
IE	59.5	61.0	53.0	53.5
IR	67.0	68.0	58.5	61.5
QA	76.0	76.5	69.0	68.0
SUM	66.0	67.5	53.0	53.5
Overall	66.9	68.2	58.6	59.1

of our RTE2 system as described in **(ref supressed for blind reviewing)** The core of the system is still the tree alignment algorithm from (Meyers et al., 1996), but without normalization of node weights and applied to Minipar instead of Maltparser output. To keep things simple, we do not apply syntactic normalization, nor do we use WordNet or other resources to improve node matching. Instead, we simply align each text tree to the corresponding hypothesis tree and calculate the *coverage*, which is defined as the proportion of aligned content words in the hypothesis. If the coverage is above a task-specific threshold, we say entailment is true, otherwise it is false.

The results are summarized in Table 2. Overall results on the test set are considerably worse than on the development set, which is most likely due to overfitting task-specific parameters for node matching and coverage. Our main interest is to what extent dependency-based paraphrasing improves our baseline prediction. The improvement on the development set is more than 1%. This is reduced to 0.5% in the case of the test set.

Our preliminary results indicate a small positive effect of dependency-based paraphrasing on the results of our RTE system. Unlike most earlier work, we did not add resources other than Minipar dependency trees and DIRT paraphrase trees, in order to isolate the contribution of syntactic paraphrases to RTE. Nevertheless, our RTE3 system may be improved by using WordNet or other lexical resources to improve node matching, both in the paraphrasing step and in the tree-alignment step. In future work, we hope to improve both the paraphrasing method (along the lines discussed in Section 2.5) and the RTE system itself.

Acknowledgments We would like to thank Dekang Lin and Patrick Pantel for allowing us to use the DIRT data. This work was jointly conducted within the DAESO project funded by the Stevin program (De Nederlandse Taalunie) and the IMOGEN project funded by the Netherlands Organization for Scientific Research (NWO).

References

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 Second PASCAL Challenges Workshop on Recognising Textual Entailment*, pages 1–9, Venice, Italy.

W. Bosma and C. Callison-Burch. 2006. Paraphrase substitution for recognizing textual entailment. In *Proceedings of CLEF*.

R. de Salvo Braz, R. Girju, V. Punyakanok, D. Roth, and M. Sammons. 2005. An inference model for semantic entailemnt in natural language. In *Proceedings of the First Pascal Challenge Workshop on Recognizing Textual Entailment*, pages 29–32.

A. Hickl, J. Williams, J. Bensley, K. Roberts, B. Rink, and Y. Shi. 2006. Recognizing textual entailment with lccs groundhog system. In *Proceedings of the Second PASCAL Challenges Workshop on Recognising Textual Entailment*, pages 80–85, Venice, Italy.

Dekang Lin and Patrick Pantel. 2001. Discovery of inference rules for question answering. *Natural Language Engineering*, 7(4):343–360.

Dekang Lin. 1998. Dependency-based evaluation of minipar. In *Proceedings of the Workshop on Evaluation of Parsing Systems at LREC 1998*, pages 317–330, Granada, Spain.

E. Marsi, E. Krahmer, W. Bosma, and M. Theune. 2006. Normalized alignment of dependency trees for detecting textual entailment. In *Proceedings of the Second PASCAL Challenges Workshop on Recognising Textual Entailment*, pages 56–61, venice, Italy.

Adam Meyers, Roman Yangarber, and Ralph Grisham. 1996. Alignment of shared forests for bilingual corpora. In *Proceedings of 16th International Conference on Computational Linguistics (COLING-96)*, pages 460–465, Copenhagen, Denmark.

R. Nielsen, W. Ward, and J.H. Martin. 2006. Toward dependency path based entailment. In *Proceedings of the Second PASCAL Challenges Workshop on Recognising Textual Entailment*, pages 44–49, Venice, Italy.

R. Raina, A. Haghighi, C. Cox, J. Finkel, J. Michels, K. Toutanova, B. MacCartney, M.C. de Marneffe, C.D. Manning, and A.Y. Ng. 2005. Robust textual inference using diverse knowledge sources. In *Proceedings of PASCAL Recognising Textual Entailment Workshop*.

J. C. Reynar and A. Ratnaparkhi. 1997. A maximum entropy approach to identifying sentence boundaries. In *Proceedings of the Fifth Conference on Applied Natural Language Processing*, Washington, D.C.

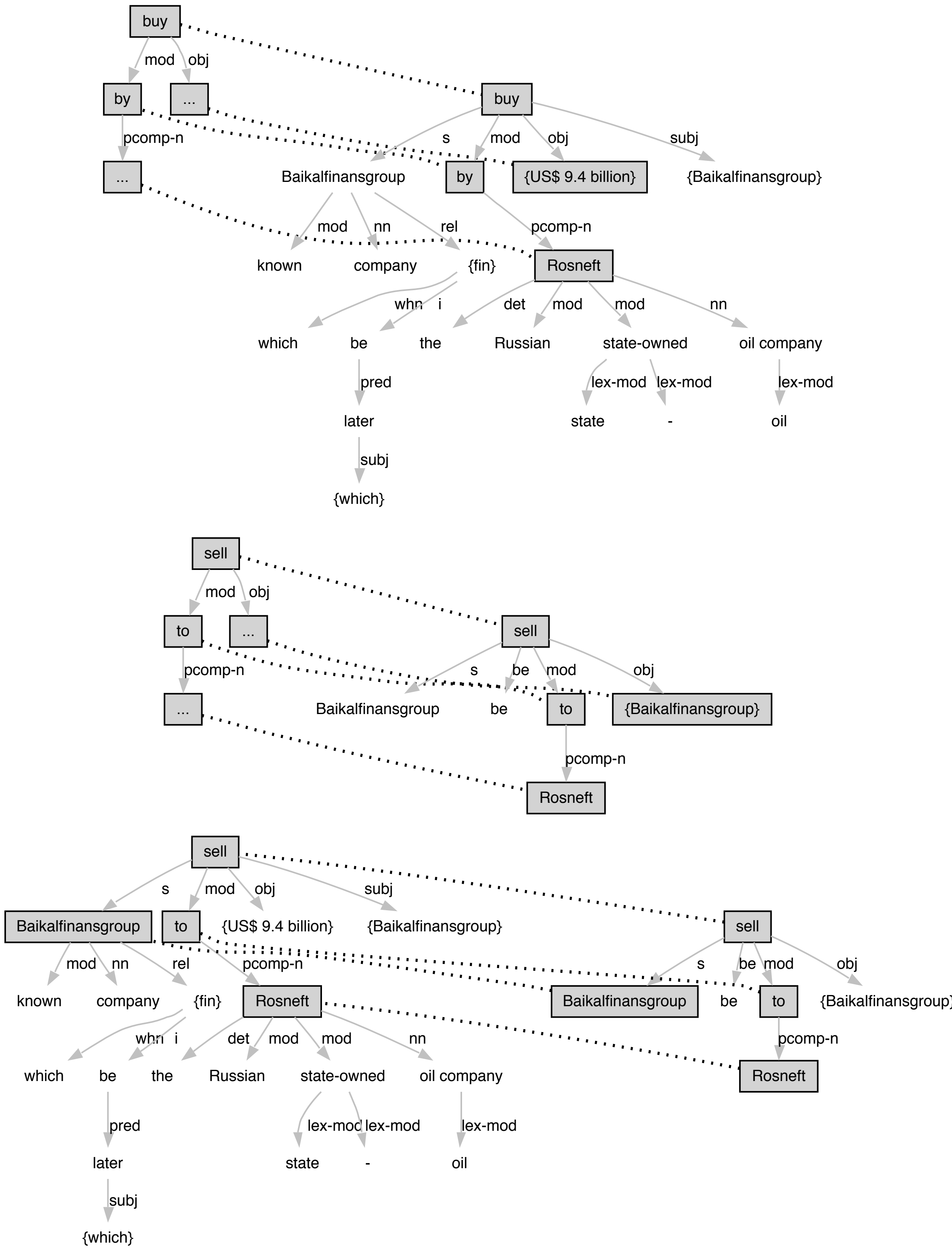

Figure 2: *Alignment of paraphrase source to text (top), alignment of paraphrase translation to hypothesis (mid), and alignment of hypothesis to paraphrased text (bottom) for pair 1 from RTE3 dev set*