Sentence Comparison using Robust Minimal Recursion Semantics and an Ontology

Rebecca Dridan[◊] **and Francis Bond**[♠]
[◊] rdrid@csse.unimelb.edu.au [♠] bond@cslab.kecl.ntt.co.jp
[◊] The University of Melbourne
[♠] NTT Communication Science Laboratories,
Nippon Telegraph and Telephone Corporation

Abstract

We design and test a sentence comparison method using the framework of Robust Minimal Recursion Semantics which allows us to utilise the deep parse information produced by Jacy, a Japanese HPSG based parser and the lexical information available in our ontology. Our method was used for both paraphrase detection and also for answer sentence selection for question answering. In both tasks, results showed an improvement over Bag-of-Words, as well as providing extra information useful to the applications.

1 Introduction

Comparison between sentences is required for many NLP applications, including question answering, paraphrasing, text summarization and entailment tasks. In this paper we show an RMRS (Robust Minimal Recursion Semantics, see Section 1.1) comparison algorithm that can be used to compare sentences in any language that has RMRS generating tools available. Lexical resources of any language can be plugged in to give a more accurate and informative comparison.

The simplest and most commonly used methods of judging sentence similarity use word overlap – either looking for matching word sequences, or comparing a Bag-of-Words representation of each sentence. Bag-of-Words discards word order, and any structure designated by such, so that *the cat snored and the dog slept* is equivalent to *the dog snored and the cat slept*. Sequence matching on the other hand requires exact word order matching and hence *the game began quietly* and *the game quietly began* are not considered a match. Neither method allows for synonym matching.

Hirao et al. (2004) showed that they could get a much more robust comparison using dependency information rather than Bag-of-Words, since they could abstract away from word order but still compare the important elements of a sentence. Using deep parsing information, such as dependencies, but also deep lexical resources where available, enables a much more informative and robust comparison, which goes beyond lexical similarity. We use the RMRS framework as our comparison format because it has the descriptive power to encode the full semantics, including argument structure. It also enables easy combination of deep and shallow information and, due to its flat structure, is easy to manage computationally.

1.1 Robust Minimal Recursion Semantics

Robust Minimal Recursion Semantics (RMRS) is a form of flat semantics which is designed to allow deep and shallow processing to use a compatible semantic representation, while being rich enough to support generalized quantifiers (Frank, 2004). The main component of an RMRS representation is a bag of elementary predicates and their arguments.

An elementary predicate always has a unique label, a relation type, a relation name and an ARG0 feature. The example in Figure 1 has a label of *h5* which uniquely identifies this predicate. Relation types can either be REALPRED for a predicate that relates directly to a content word from the input text, or GPRED for grammatical predicates which may not have a direct referent in the text. For examples in this paper, a REALPRED is distinguished by an underscore (_) before the relation name.

The GPRED relation names come from a

Proceedings of the Workshop on Linguistic Distances, pages 35–42,
Sydney, July 2006. ©2006 Association for Computational Linguistics

$$\begin{bmatrix} \texttt{\_unten\_s} & \\ \textsc{lbl} & h5 \\ \textsc{arg0} & e6 \end{bmatrix}$$

Figure 1: Elementary predicate for 運転 *unten* "drive"

closed-set which specify common grammatical relations, but the REALPRED names are formed from the word in the text they relate to and this is one way in which RMRS allows underspecification. A full relation name is of the form `lemma_pos_sense`, where the pos (part of speech) is drawn from a small set of general types including **n**oun, **v**erb and **s**ahen (verbal noun). The sense is a number that identifies the sense of the word within a particular grammar being used. The POS and sense information are only used when available and hence the `_unten_s_1` is more specific but compatible with `_unten_s` or even `_unten`.

The ARG0 feature (*e6* in Figure 1) is the referential index of the predicate. Predicates with the same ARG0 are said to be referentially co-indexed and therefore have the same referent in the text.

A shallow parse might provide only the features shown in Figure 1, but a deep parse can also give information about other arguments as well as scoping constraints. The features ARG1..ARG4 specify the indices of the semantic arguments of the relevant predicate, similar to PropBank's argument annotation (Kingsbury et al., 2002). While the RMRS specification does not define semantic roles for the ARGn features, in practice ARG1 is generally used for the AGENT and ARG2 for the PATIENT. Features ARG3 and ARG4 have less consistency in their roles.

We will use (1) and (2) as examples of similar sentences. They are definition sentences for one sense of ドライバー *doraiba-* "driver", taken from two different lexicons.

(1)	自動車	を	運転	する	人
	jidōsha	*wo*	*unten*	*suru*	*hito*
	car	ACC	drive	do	person

"a person who drives a car"

(2)	自動車	など	の	運転	者
	jidōsha	*nado*	*no*	*unten*	*sha*
	car	etc.	ADN	drive	-er

"a driver of cars etc."

Examples of deep and shallow RMRS results for (1) are given in Figure 2. Deep results for (2) are given in Figure 3.

2 Algorithm

The matching algorithm is loosely based on RMRS comparison code included in the LKB (Copestake, 2002: ⟨`http://www.delph-in.net/lkb/`⟩), which was used in Ritchie (2004), however that code used no outside lexical resources and we have substantially changed the matching algorithm.

The comparison algorithm is language independent and can be used for any RMRS structures. It first compares all elementary predicates from the RMRSs to construct a list of match records and then examines, and potentially alters, the list of match records according to constraints encoded in the ARGn variables. Using the list of scored matches, the lowest scoring possible match set is found and, after further processing on that set, a similarity score is returned. The threshold for deciding whether a pair of sentences should be considered similar or not can be determined separately for different applications.

2.1 Matching Predicates

The elementary predicates (EPs) of our RMRS structures are divided into two groups - those that have a referent in the text, hereafter known as content EPs, and those that don't. There are three kinds of content EP: REALPREDs, which correspond to content bearing words that the grammar knows; GPREDs with a CARG (Constant ARGument) feature, which are used to represent proper names and numbers; and GPREDs with a predicate name starting with `generic` such as `generic_verb` which are used for unknown words that have only been identified by their part of speech. All other EPs have no referent and are used to provide information about the content EPs or about the structure of the sentence as a whole. These non-content EPs can provide some useful information, but generally only in relation to other content EPs.

Each content EP of the first RMRS is compared to all content EPs in the second RMRS, as shown in Figure 4.

Matches are categorised as EXACT, SYNONYM, HYPERNYM, HYPONYM or NO MATCH and a numerical score is assigned. The nu-

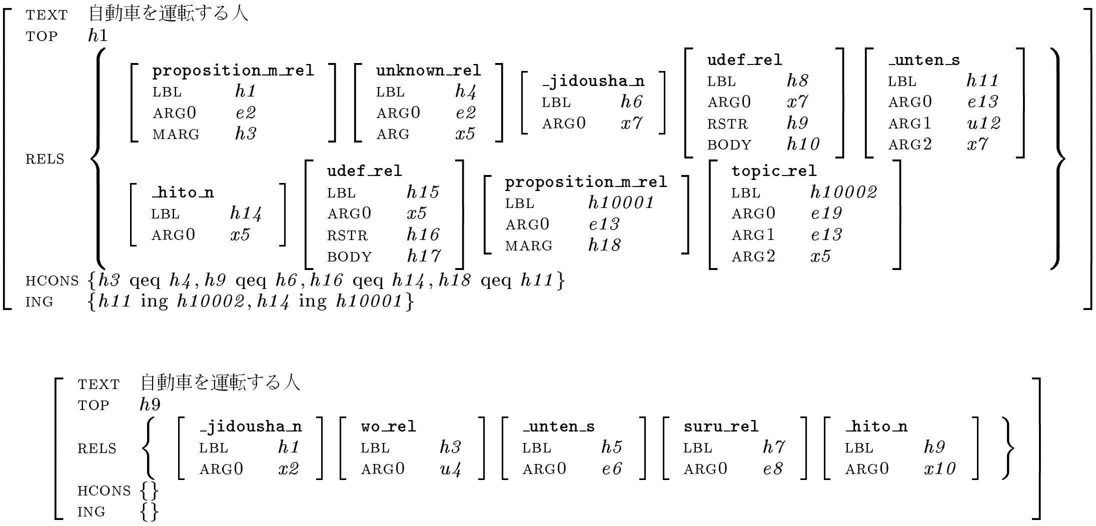
TEXT 自動車を運転する人
TOP h1
RELS
proposition_m_rel LBL h1 ARG0 e2 MARG h3
unknown_rel LBL h4 ARG0 e2 ARG x5
_jidousha_n LBL h6 ARG0 x7
udef_rel LBL h8 ARG0 x7 RSTR h9 BODY h10
_unten_s LBL h11 ARG0 e13 ARG1 u12 ARG2 x7
_hito_n LBL h14 ARG0 x5
udef_rel LBL h15 ARG0 x5 RSTR h16 BODY h17
proposition_m_rel LBL h10001 ARG0 e13 MARG h18
topic_rel LBL h10002 ARG0 e19 ARG1 e13 ARG2 x5
HCONS {h3 qeq h4, h9 qeq h6, h16 qeq h14, h18 qeq h11}
ING {h11 ing h10002, h14 ing h10001}
TEXT 自動車を運転する人
TOP h9
RELS
_jidousha_n LBL h1 ARG0 x2
wo_rel LBL h3 ARG0 u4
_unten_s LBL h5 ARG0 e6
suru_rel LBL h7 ARG0 e8
_hito_n LBL h9 ARG0 x10
HCONS {}
ING {}

Figure 2: Deep (top) and shallow (bottom) RMRS results for 自動車 を 運転 する 人

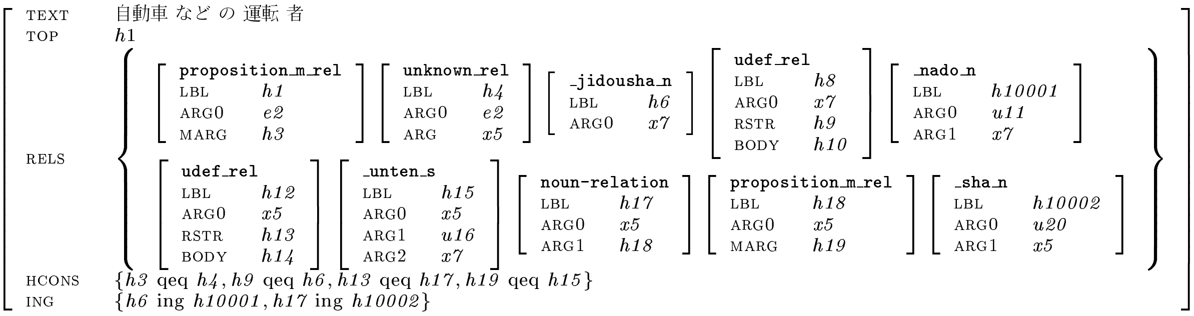
TEXT 自動車 など の 運転 者
TOP h1
RELS
proposition_m_rel LBL h1 ARG0 e2 MARG h3
unknown_rel LBL h4 ARG0 e2 ARG x5
_jidousha_n LBL h6 ARG0 x7
udef_rel LBL h8 ARG0 x7 RSTR h9 BODY h10
_nado_n LBL h10001 ARG0 u11 ARG1 x7
udef_rel LBL h12 ARG0 x5 RSTR h13 BODY h14
_unten_s LBL h15 ARG0 x5 ARG1 u16 ARG2 x7
noun-relation LBL h17 ARG0 x5 ARG1 h18
proposition_m_rel LBL h18 ARG0 x5 MARG h19
_sha_n LBL h10002 ARG0 u20 ARG1 x5
HCONS {h3 qeq h4, h9 qeq h6, h13 qeq h17, h19 qeq h15}
ING {h6 ing h10001, h17 ing h10002}

Figure 3: RMRS representation for 自動車 など の 運転 者

```
foreach ep1 in contentEPs1
 foreach ep2 in contentEPs2
  (score, match) = match_EPs(ep1, ep2)
  if match != NO_MATCH
    add_to_matches(ep1, ep2, score, match)
  endif
 done
done
```

Figure 4: Predicate match pseudo-code

merical score represents the distance between the two EPs, and hence an EXACT match is assigned a score of zero.

The level of matching possible depends on the lexical resources available. With no extra resources, or only a dictionary to pick up orthographic variants, the only match types possible are EXACT and NO MATCH. By adding a thesaurus, an ontology or a gazetteer, it is then possible to return SYNONYM, HYPERNYM and HYPONYM match relations. In our experiments we used the ontology described in Section 3.2.2, which provides all three extra match types. Adding a thesaurus only would enable SYNONYM matching, while a gazetteer could be added to give, for example, `Tokyo` is a HYPONYM of `city`.

```
Matches:
   hito_n     - sha_n     : HYPERNYM (2)
   jidosha_n - jidosha_n: EXACT (0)
   unten_s_2 - unten_s_2: EXACT (0)
```

Figure 5: First pass match list for (1) and (2)

At the end of the first pass, a list of match records shows all EP matches with their match type and score. Each EP can have multiple possible matches. The output of comparing (1) and (2), with the RMRSes in Figures 2 and 3, is shown in Figure 5. This shows `hito_n` (人 *hito* "person") tagged as a HYPERNYM of

```
foreach match in matches
 gpreds1 = get_gpreds_arg0(ep1{arg0})
 gpreds2 = get_gpreds_arg0(ep2{arg0})
 totalgpreds = len gpreds1 + len gpreds2
 foreach ep1 in gpreds1
  foreach ep2 in gpreds2
   if(match_gram_eps(ep1, ep2)
    remove(ep1, gpreds1)
    remove(ep2, gpreds2)
   endif
  done
 done
 gpreds_left = len gpreds1 + len gpreds2
 left = gpreds_left/totalgpreds
 match{score}+= left*gpredWeight
done
```

Figure 6: Matching ARG0s

sha_n (者 *sha* "-er" is a suffix indicating a person, normally the agent of a verb: it is more restrictive than English *-er*, in that it only refers to people).

2.2 Constraints Pass

For each possible match, all the non-content EPs that have the same ARG0 value as the content EPs in the match are examined, since these have the same referent. If each non-content EP related to the content EP on one side of the match can be matched to the non-content EPs related to the other content EP, no change is made. If not, however, a penalty is added to the match score, as shown in Figure 6. In our example, unten_s_2 from the first sentence has a proposition_m_rel referentially co-indexed, while the second unten_s_2 has a proposition_m_rel, a noun-relation and a udef_rel, and so a small penalty is added as shown in Figure 7.

The second check in the constraint match pass examines the arguments (ARG1, ARG2, ARG3, ARG4) of each of the matches. It looks for possible matches found between the EPs listed as ARGn for each match. This check can result in three separate results: both EPs have an ARGn but there is no potential match found between the respective ARGn EPs, a potential match has been found between the ARGn EPs, or only one of the EPs in the match has an ARGn feature.

Where both EPs have an ARGn feature, the score (distance) of the match is decreased or increased depending on whether a match between the ARGn variables was found. Given that the RMRS definition does not specify a 'meaning' for the ARGn variables, comparing, for example, ARG1 variables from two different predicates may not necessarily be comparing the same semantic roles. However, because of the consistency found in ARG1 and ARG2 meaning this is still a useful check. Of course, if we are comparing the same relation, the ARGs will all have the same meaning. The comparison method allows for different penalties for each of ARG1 to ARG4, and also includes a scaling factor so that mismatches in ARGs when comparing EXACT EP matches will have more effect on the score than in non EXACT matches. If one EP does not have the ARGn feature, no change is made to the score. This allows for the use of underspecified RMRSs, in the case where the parse fails.

```
Matches:
   hito_n    - sha_n    : HYPERNYM (2.1)
   jidosha_n - jidosha_n: EXACT (0)
   unten_s_2 - unten_s_2: EXACT (0.05)
```

Figure 7: Match list
Slight penalty added to unten_s_2 and hito_n for non-matching non-content EPs

At the end of this pass, the scores of the matches in the match list may have changed but the number of matches is still the same.

2.3 Constructing the Sets

Match sets are constructed by using a branch-and-bound decision tree. Each match is considered in order, and the tree is branched if the next match is possible, given the proceeding decisions. Any branch which is more than two decisions away from the best score so far is pruned. At the end of this stage, the lowest scoring match set is returned and then this is further processed.

If no matches were found, processing stops and a sentinel value is returned. Otherwise, the non matching predicates are grouped together by their ARG0 value. Scoping constraints are checked and if any non matching predicate outscopes a content predicate it is added to that grouping. Hence if it outscopes a matching EP it becomes part of the match, otherwise it becomes part of a non-matching EP group.

Any group of grammatical EPs that shares an ARG0 but does not contain a content predicate is matched against any similar groupings

```
Best score is 0.799 for the match set:
MATCHES:
 hito_n-sha_n: HYPERNYM:2.1
 jidousha_n-jidousha_n:EXACT:0
 unten_s_2-unten_s_2:EXACT:0.05
 proposition_m_rel-proposition_m_rel:EXACT:0
UNMATCHED1:
UNMATCHED2:
 u11: h10001:nado_n
```

Figure 8: Verbose comparison output

in the other RMRS. This type of match can only be EXACT or NO MATCH and will make only a small difference in the final score.

Content predicates that have not been matched by this stage are not processed any further, although this is an area for further investigation. Potentially negation and other modifiers could be processed at this point.

2.4 Output

The output of the comparison algorithm is a numeric score and also a representation of the final best match found.

The numerical score, using the default scoring parameters, ranges between 0 (perfect match) and 3. As well as the no match score (-5), sentinel values are used to indicate missing input data so it is possible to fall back to a shallow parse if the deep parse failed.

Details of the match set are also returned for further processing or examination if the application requires. This shows which predicates were deemed to match, and with what score, and also shows the unmatched predicates. Figure 8 shows the output of our example comparison.

3 Resources

While the comparison method is language independent, the resources required are language specific. The resources fall in to two different categories: parsing and morphological analysis tools that produce the RMRSs, and lexical resources such as ontologies, dictionaries and gazetteers for evaluating matches.

3.1 Parsing

Japanese language processing tools are freely available. We used the Japanese grammar Jacy (Siegel and Bender, 2002), a deep parsing HPSG grammar that produces RMRSs for our primary input source.

When parsing with Jacy failed, comparisons could still be made with RMRS produced from shallow tools such as ChaSen (Matsumoto et al., 2000), a morphological analyser or CaboCha (Kudo and Matsumoto, 2002), a Japanese dependency parser. Tools have been built to produced RMRS from the standard output of both those tools.

The CaboCha output supplies similar dependency information to that of the Basic Elements (BE) tool used by Hovy et al. (2005b) for multi-document summarization. Even this intermediate level of parsing gives better comparisons than either word or sequence overlap, since it is easier to compare meaningful elements (Hovy et al., 2005a).

3.2 Lexical Resources

Whilst deep lexical resources are not available for every language, where they are available, they should be used to make comparisons more informative. The comparison framework allows for different lexical resources to be added to a pipeline. The pipeline starts with a simple relation name match, but this could be followed by a dictionary to extract orthographic variants and then by ontologies such as WordNet (Fellbaum, 1998) or GoiTaikei (Ikehara et al., 1997), gazetteers or named entity recognisers to recognise names of people and places. The sections below detail the lexical resources we used within our experiments.

3.2.1 The Lexeed Semantic Database

The Lexeed Semantic Database of Japanese is a machine readable dictionary that covers the most familiar words in Japanese, based on a series of psycholinguistic tests (Kasahara et al., 2004). Lexeed has 28,000 words divided into 46,000 senses and defined with 75,000 definition sentences. Each entry includes a list of orthographic variants, and the pronunciation, in addition to the definitions.

3.2.2 Ontology

The lexicon has been sense-tagged and parsed to give an ontology linking senses with various relations, principally hypernym and synonym (Nichols et al., 2005). For example, ⟨HYPERNYM, ドライバー *doraibā* "driver", クラブ *kurabu* "club"⟩. The ontology entries for nouns have been hand checked and corrected, including adding hypernyms for words where the genus term in the definition was very general, e.g “a word used to refer insultingly to men” where *man* is a more useful hypernym than *word* for the defined term *yarou*.

4 Evaluation

We evaluated the performance of the RMRS comparison method in two tasks. First it was used to indicate whether two sentences were possible paraphrases. In the second task, we used the comparison scores to select the most likely sentence to contain the answer to a question.

4.1 Paraphrasing

In this task we compared definitions sentences for the same head word from two different Japanese dictionaries - the Lexeed dictionary (§3.2.1) and the Iwanami Kokugo Jiten (Iwanami: Nishio et al., 1994), the Japanese dictionary used in the SENSEVAL-2 Japanese lexical task (Shirai, 2002).

There are 60,321 headwords and 85,870 word senses in Iwanami. Each sense in the dictionary consists of a sense ID and morphological information (word segmentation, POS tag, base form and reading, all manually post-edited).

The definitions in Lexeed and Iwanami were linked by headword and three Japanese native speakers assessed each potential pair of sense definitions for the same head word to judge which definitions were describing the same sense. This annotation not only described which sense from each dictionary matched, but also whether the definitions were equal, equivalent, or subsuming.

The examples (1) and (2) are the definitions of sense 2 of ドライバー *doraibā* “driver” from Lexeed and Iwanami respectively. They were judged to be equivalent definitions by all three annotators.

4.1.1 Method

Test sets were built consisting of the Lexeed and Iwanami definition pairs that had been annotated in the gold standard to be either non-matching, equal or equivalent. Leaving out those pairs annotated as having a subsumption relation made it a clearer task judging between paraphrase or not, rather than examining partial meaning overlap. Ten sets of 5,845 definition pairs were created, with each set being equally split between matching and non-matching pairs. This gives data that is to some extent semantically equivalent (the same word sense is being defined), but with no guarantee of syntactic equivalence.

Comparisons were made between the first sentence of each definition with both a Bag-of-Words comparison method and our RMRS based method. If RMRS output was not available from Jacy (due to a failed parse), RMRS from CaboCha was used as a fall back shallow parse result.

Scores were output and then the best threshold score for each method was calculated on one of the 10 sets. Using the calculated threshold score, pairs were classified as either matching or non-matching. Pairs classified as matching were evaluated as correct if the gold standard annotation was either equal or equivalent.

4.1.2 Results

The Bag-of-Words comparison got an average accuracy over all sets of 73.9% with 100% coverage. A break down of the results shows that this method was more accurate (78%) in correctly classifying non-matches than matches (70%). This is to be expected since it won’t pick up equivalences where a word has been changed for its synonym.

The RMRS comparison had an accuracy was 78.4% with almost 100% coverage, an improvement over the Bag-of-Words. The RMRS based method was also more accurate over non matches (79.9%) than matches (76.6%), although the difference is not as large. Considering only those sentences with a parse from JACY gave an accuracy of 81.1% but with a coverage of only 46.1%. This shows that deep parsing improves precision, but must be used in conjunction with a shallower fallback.

To explore what effect the ontology was having on the results, another evaluation was performed without the ontology matching. This had an accuracy of 77.3% (78.1% using Jacy, 46.1% coverage). This shows that the information available in the ontology definitely improves scores, but that even without that sort of deep lexical resource, the RMRS matching can still improve on Bag-of-Words using just surface form abstraction and argument matching.

4.2 Answer Sentence Selection

To emulate a part of the question answering pipeline, we used a freely available set of 2000 Japanese questions, annotated with, among other things, answer and answer document ID (Sekine et al., 2002). The document IDs for the answer containing documents refer to the Mainichi Newspaper 1995 corpus which has been used as part of the document collection for NTCIR's Question Answering Challenges. The documents range in length from 2 to 83 sentences.

4.2.1 Method

For every question, we compared it to each sentence in the answer document. The sentence that has the best similarity to the question is returned as the most likely to contain the answer. For this sort of comparison, an *entails* option was added that changes the similarity scoring method slightly so that only non-matches in the first sentence increase the score. The rationale being that in Question Answering (and also in entailment), everything present in the question (or hypothesis) should be matched by something in the answer, but having extra, unmatched information in the answer should not be penalised.

The task is evaluated by checking if the answer does exist in the sentence selected. This means that more than one sentence can be the correct answer for any question (if the answer is mentioned multiple times in the article).

4.2.2 Results

The Bag-of-Words comparison correctly found a sentence containing the answer for 62.5% of the 2000 questions. The RMRS comparison method gave a small improvement, with a result of 64.3%. Examining the data showed this to be much harder than the paraphrase task because of the language level involved. In the paraphrasing task, the sentences averaged around 10 predicates each, while the questions and sentences in this task averaged over 3 times longer, with about 34 predicates. The words used were also less likely to be in the lexical resources both because more formal, less familiar words were used, and also because of the preponderance of named entities. Adding name lists of people, places and organisations would greatly improve the matching in this instance.

5 Future Directions

5.1 Applications

Since the comparison method was written to be language independent, the next stage of evaluation would be to use it in a non-Japanese task. The PASCAL Recognising Textual Entailment (RTE) Challenge (Dagan et al., 2005) is one recent English task where participants used sentence comparison extensively. While the task appears to call for inference and reasoning, the top 5 participating groups used statistical methods and word overlap only. Vanderwende et al. (2005) did a manual evaluation of the test data and found that 37% could be decided on syntactic information alone, while adding a thesaurus could increase that coverage to 49%. This means that RMRS comparison has the potential to perform well. Not only does it improve on basic word overlap, but it allows for easy addition of a thesaurus or dictionary. Further, because of the detailed match output available, the method could be extended in post processing to encompass some basic inference methods.

Aside from comparing sentences, the RMRS comparison can be used to compare the RMRS output of different tools for the same sentence so that the compatibility of the outputs can be evaluated and improved.

5.2 Extensions

One immediate future improvement planned is to add named entity lists to the lexical resources so that names of people and places could be looked up. This would allow partial matches between, e.g., `Clinton` is a HYPONYM of `person`, which would be particularly useful for Question Answering.

Another idea is to add a bilingual dictionary and try cross-lingual comparisons. As the RMRS abstracts away much of the surface specific details, this might be useful for sentence alignment.

To go beyond sentence by sentence comparison, we have plans to implement a method for multi-sentence comparisons by either combining the RMRS structures before comparison, or post-processing the sentence comparison outputs. This could be particularly interesting for text summarization.

6 Conclusions

Deep parsing information is useful for comparing sentences and RMRS gives us a useful framework for utilising this information when it is available. Our RMRS comparison was more accurate then basic word overlap similarity measurement particularly in the paraphrase task where synonyms were often used. Even when the ontology was not used, abstracting away from surface form, and matching arguments did give an improvement. Falling back to shallow parse methods increases the robustness which is often an issue for tools that use deep processing, while still allowing the use of the most accurate information available.

The comparison method is language agnostic and can be used for any language that has RMRS generating tools. The output is much more informative than Bag-of-Words, making it useful in many applications that need to know exactly how a sentence matched or aligned.

Acknowledgements

This work was started when the first author was a visitor at the NTT Communication Science Laboratories, Nippon Telegraph and Telephone Corporation. The first author was also supported by the Pam Todd scholarship from St Hilda's College. We would like to thank the NTT Natural Language Research Group and two anonymous reviewers for their valuable input.

References

Ann Copestake. 2002. *Implementing Typed Feature Structure Grammars.* CSLI Publications.

Ido Dagan, Oren Glickman, and Bernado Magnini. 2005. The PASCAL recognising textual entailment challenge. In *Proceedings of the PASCAL Challenges Workshop on Recognising Textual Entailment.*

Christine Fellbaum. 1998. A semantic network of English verbs. In Christine Fellbaum, editor, *WordNet: An Electronic Lexical Database*, chapter 3, pages 70–104. MIT Press.

Anette Frank. 2004. Constraint-based RMRS construction from shallow grammars. In *20th International Conference on Computational Linguistics: COLING-2004*, pages 1269–1272. Geneva.

Tsutomu Hirao, Jun Suzuki, Hideki Isozaki, and Eisaku Maeda. 2004. Dependency-based sentence alignment for multiple document summarization. In *Proceedings of the COLING.*

Eduard Hovy, Junichi Fukumoto, Chin-Yew Lin, and Liang Zhao. 2005a. Basic elements. (`http://www.isi.edu/~cyl/BE`).

Eduard Hovy, Chin-Yew Lin, and Liang Zhao. 2005b. A BE-based multi-document summarizer with sentence compression. In *Proceedings of Multilingual Summarization Evaluation.*

Satoru Ikehara, Masahiro Miyazaki, Satoshi Shirai, Akio Yokoo, Hiromi Nakaiwa, Kentaro Ogura, Yoshifumi Ooyama, and Yoshihiko Hayashi. 1997. *Goi-Taikei — A Japanese Lexicon.* Iwanami Shoten, Tokyo. 5 volumes/CDROM.

Kaname Kasahara, Hiroshi Sato, Francis Bond, Takaaki Tanaka, Sanae Fujita, Tomoko Kanasugi, and Shigeaki Amano. 2004. Construction of a Japanese semantic lexicon: Lexeed. SIG NLC-159, IPSJ, Tokyo. (in Japanese).

Paul Kingsbury, Martha Palmer, and Mitch Marcus. 2002. Adding semantic annotation to the penn treebank. In *Proceedings of the Human Language Technology 2002 Conference.*

Taku Kudo and Yuji Matsumoto. 2002. Japanese dependency analysis using cascaded chunking. In *CoNLL 2002: Proceedings of the 6th Conference on Natural Language Learning 2002 (COLING 2002 Post-Conference Workshops)*, pages 63–69. Taipei.

Yuji Matsumoto, Kitauchi, Yamashita, Hirano, Matsuda, and Asahara. 2000. *Nihongo Keitaiso Kaiseki System: Chasen*, version 2.2.1 manual edition. `http://chasen.aist-nara.ac.jp`.

Eric Nichols, Francis Bond, and Daniel Flickinger. 2005. Robust ontology acquisition from machine-readable dictionaries. In *Proceedings of the International Joint Conference on Artificial Intelligence IJCAI-2005*, pages 1111–1116. Edinburgh.

Minoru Nishio, Etsutaro Iwabuchi, and Shizuo Mizutani. 1994. *Iwanami Kokugo Jiten Dai Go Han [Iwanami Japanese Dictionary Edition 5].* Iwanami Shoten, Tokyo. (in Japanese).

Anna Ritchie. 2004. Compatible RMRS representations from RASP and the ERG. Technical Report UCAM-CL-TR-661.

Satoshi Sekine, Kiyoshi Sudo, Yusuke Shinyama, Chikashi Nobata, Kiyotaka Uchimoto, and Hitoshi Isahara. 2002. NYU/CRL QA system, QAC question analysis and CRL QA data. In *Working Notes of NTCIR Workshop 3.*

Kiyoaki Shirai. 2002. Construction of a word sense tagged corpus for SENSEVAL-2 Japanese dictionary task. In *Third International Conference on Language Resources and Evaluation (LREC-2002)*, pages 605–608.

Melanie Siegel and Emily M. Bender. 2002. Efficient deep processing of Japanese. In *Proceedings of the 3rd Workshop on Asian Language Resources and International Standardization at the 19th International Conference on Computational Linguistics.* Taipei.

Lucy Vanderwende, Deborah Coughlin, and Bill Dolan. 2005. What syntax can contribute in entailment task. In *Proceedings of the PASCAL Challenges Workshop on Recognising Textual Entailment.*