Detecting Contradiction in Text by Using Lexical Mismatch and Structural Similarity

Daniel Andrade Masaaki Tsuchida Takashi Onishi Kai Ishikawa
Knowledge Discovery Research Laboratories, NEC Corporation, Nara, Japan
{s-andrade@cj, m-tsuchida@cq,
t-onishi@bq, k-ishikawa@dq}.jp.nec.com

ABSTRACT

Detecting contradicting statements is a crucial sub-problem for filtering textual entailment pairs. Given two text pairs T1 and T2 that are topically related, the task is to find whether the two statements are in contrast to each other, or not. In many situations a mismatch in named entities or numerical expressions is a strong clue for a contradiction between T1 and T2. However, if the dependency parse trees are quite different, and the words of T1 and T2 cannot be aligned well, then this indicates that a lexical mismatch is not sufficient to conclude contradiction of T1 and T2. We present a new method that assumes the higher the structural similarity of two sentences, the higher the chance that a contradiction on the word level indicates contradiction on the sentence level. We participated in two subtasks of RITE2 at NTCIR-10 which contain many contradicting statements in a real world setting. Our system became second place in the ExamBC subtask (formal run, official result), and first place in the ExamSearch subtask (formal run, unofficial result). We show that our proposed method contributed to the improvement in both tasks.

Team Name

KDR

Subtasks

ExamBC, ExamSearch (Japanese)

Keywords

Textual Entailment, Contradiction Detection, Tree Edit Distance

1. INTRODUCTION

Here in this paper we describe our system participating in the ExamBC and ExamSearch subtasks of the NTCIR-10 Recognizing Inference in Text (RITE) task [6]. Finding contradiction is an important subtask of recognizing textual entailment. Especially in the domain of university entrance examinations (ExamBC, ExamSearch), we found that non-entailment is often due to clear contradicting statements in two texts T1 and T2. We therefore propose a new method for detecting contradiction in texts, and show that it contributes to the improvement of our system.

In many situations a mismatch in named entities or numerical expressions is a strong clue for a contradiction between T1 and T2. However, if the dependency parse trees are quite different, and the words of T1 and T2 cannot be aligned well, then this indicates that a lexical mismatch is not sufficient to conclude contradiction of T1 and T2. We present a new method that assumes the higher the structural similarity of two sentences, the higher the chance that a contradiction on the word level indicates contradiction on the sentence level.

In the following section, we briefly describe existing work on detecting contradiction and explain the differences to our approach. In Section 2, we describe our proposed method to detect contradiction that is incorporated in the systems that we submitted to the ExamBC and ExamSearch subtasks. In Section 3, we describe the other features that were used for judging textual entailment. In Section 4, we evaluate the impact of our proposed method and the other features that were used by our system. Finally, in Section 5, we summarize our contributions.

Alignment Cost (range)	Case Description
0.0	if w1 and w2 are the same.
0.0 - 0.2	if w1 and w2 are synonyms according to WordNet or Wikipedia redirect, or if they are antonyms according to Kougaku-Research dictionary (言語工学研究所類語辞書 2005 版)
0.2 - 1.0	if the words belong to similar categories in Bunruigoi-Hyo, or if they have the same category according to Wikipedia.
3.0	otherwise.

Figure 1: The costs for aligning two chunks (bunsetsus) as determined by the similarity of the head content words w1 and w2.

1.1 Related Work

The work in [1], extracts various features that indicate contradiction between text T1 and T2. In the first step, they align the words in T1 and T2 by using their dependency trees. In the second step, they extract from each pair of aligned words features that indicate contradiction between T1 and T2. They extract a structural contradiction feature that is set if the grammatical role of two aligned words is different. For instance, if the aligned word in T1 is an object, but in T2 it has the role of a subject, it is considered as structural contradiction. However, in Japanese due to the topic marker は (wa) it is not trivial to detect the grammatical role of a word. Furthermore, they also extract an antonym feature, that detects whether the aligned words are antonyms using WordNet and VerbOcean. We will call the latter a type of lexical contradiction.

The work in [7] also suggests a similar strategy for Japanese that uses a predicate relation database to detect antonyms. They first align all chunks in T1 and T2, and assign an "opposite" label for contradicting meaning like antonyms, or "normal" label for no contradiction. Finally, using the assumption of compositionality, they combine all labels to one label that indicates contradiction or entailment of T1 and T2. However, some alignments might cause the false detection of contradiction. For example, "Mary buys a car from Tom"(T1) entails "Tom sells a car to Mary"(T2). Using the predicate relation database, the system might be able to align "buy" and "sell" and detect the lexical contradiction ("opposite" label). The system then combines the results of all alignments (labeled as "opposite" or "normal") and might conclude an "opposite" label, meaning contradiction of T1 and T2. However, the structural similarity of T1 and T2's dependency trees is low, suggesting that the contradiction of the words "buy" and "sell" does not necessarily mean that T1 and T2 are contradicting. Our proposed method tries to overcome this problem by weighting lexical contradiction by the degree of T1 and T2's structural similarity.

2. DETECTING CONTRADICTION ON SENTENCE LEVEL

If the dependency trees of T1 and T2 are quite different, we assume that lexical contradiction of T1 and T2 has no meaning. On the other hand, if the dependency graphs are very similar, then, we assume that, lexical contradiction implies contradiction of the sentences T1 and T2.

Given two texts T1 and T2, our proposed method for detecting contradiction can be separated into the following steps:

1. Split complex sentences into simple sentences that contain only one predicate each.
2. Calculate the minimum cost alignment for each pair of simple sentences from T1 and T2.
3. Calculate the total minimum cost alignment of all simple sentences in T2 with the simple sentences in T1.
4. Calculate the degree of lexical contradiction of the aligned chunks (bunsetsus) in the aligned simple sentences.
5. Calculate the degree of contradiction between T1 and T2.

The next sub-sections explain in more detail each of the steps.

2.1 Extracting simple sentences from complex sentence

In ExamBC and ExamSearch, T1 often contains long complex sentences like the ones that occur frequently in the history text book and on Wikipedia. Also T2 itself sometimes is a complex sentence which evidence is spread over different sub-sentences of T1. Therefore, in the first step we split T1 and T2 into simple sentences. A simple sentence is a sentence that contains only one predicate. The splitting in simple sentences is done in such a way that a simple sentence contains exactly one predicate and all predicate arguments. For the identification of predicates, its arguments and necessary co-reference resolution of zero-pro-nouns, we used Syncha [2]. For example, consider the sentence of pair 9 in the training data of ExamBC shown in Figure 2. T1 is split into seven and T2 is split into two simple sentences.

T1: 一揆をおこした農民は徴兵以外にも、新政のいろいろに不満をもっていたが、1872年に施行された学制に対するそれも大きく、学制から始まった義務教育推進運動は、当初は授業料徴収があったためになかなか効果を上げなかった。
T2: 学校の建設費や授業料が民衆の負担とされたため，学制の実施にあたっては民衆の反対運動もみられた。

T11: 一揆をおこした農民
T12: 農民は徴兵以外にも、 新政のいろいろに不満をもっていたが、
T13: 1872 年に施行された学制
T14: 学制に対するそれも大きい教育推進運動は、
T15: 学制から始まった義務教育推進運動は、
T16: 授業料徴収があったために
T17: 義務教育推進運動は、当初はなかなか効果を上げなかった。

T21: 学校の建設費や授業料が民衆の負担とされたため，
T22: 学制の実施にあたっては民衆の反対運動もみられた。

Figure 2: The upper part shows T1 and T2 from pair 9 of the training data of ExamBC, the label is "Y" saying that T1 entails T2. The lower part shows that our system splits T1 and T2 into simple sentences T11, T12,...,T17 and T21, T22, respectively.

Degree of Lexical Contradiction	Case Description
1.0	if b1 and b2 are both place names but their surface forms are different (place names are extracted from Wikipedia).
1.0	if b1 and b2 contain antonyms according to Kougaku-Research dictionary.
0.5	if the head content word is the same but all other words (morphemes) are different and are not listed as synonyms (in any of the used resources).
0.0	otherwise.

Figure 3: Determining the degree of lexical contradiction of two aligned chunks (bunsetsus) b1 and b2.

2.2 Calculating minimum alignment costs between two simple sentences

Similar to the work of [5], we use the costs of aligning the parse trees of T1 and T2 as an approximation of two sentences' structural (grammatical) similarity. The costs for aligning two chunks are determined by the similarity of the head content words as shown in Figure 1. Note that our

Using Top-1 and Top-2 Search Result from Text Book				
	CV Training Data		Test Data	
Features	Accuracy	Macro-F1	Accuracy	Macro-F1
BC	60.47	54.82	63.39	58.59
BC - Contradiction	59.66	51.80	62.28	53.23
BC + Tsubaki Score	57.93	51.76	63.17	57.39
BC + Tsubaki Score + Word Overlap	58.32	53.13	63.17	57.55
BC + Tsubaki Score + Word Overlap + Named Entity	59.48	55.53	63.84	58.24
Using Top-1 and Top-2 Search Result from Text Book and Wikipedia				
	CV Training Data		Test Data	
Features	Accuracy	Macro-F1	Accuracy	Macro-F1
BC	59.84	55.22	62.50	57.91
BC - Contradiction	58.89	54.00	60.27	55.55
BC + Tsubaki Score	61.03	56.97	61.61	56.46
BC + Tsubaki Score + Word Overlap	60.60	55.12	60.94	55.31
BC + Tsubaki Score + Word Overlap + Named Entity (KDR-JA-ExamSearch-02)	64.13	59.18	64.51	58.12

Table 1: Results for ExamSearch. BC corresponds to the system KDR-JA-ExamBC-02 but trained on the search results from Tsubaki, instead of T1 from ExamBC. BC - X means all features of BC without feature X. BC + X means all features of BC additionally including feature X. The last line shows the result of the system KDR-JA-ExamSearch-02.

	CV Training Data		Test Data	
Features	Accuracy	Macro-F1	Accuracy	Macro-F1
BC (KDR-JA-ExamBC-02)	73.69	72.59	68.75	66.90
BC - Contradiction	70.38	69.27	68.08	66.77
BC - Tree-edit distance	72.93	71.32	68.75	66.50
BC - Character Overlap	59.73	50.09	61.16	49.68
BC - Temporal Expressions	71.94	70.81	67.63	65.83

Table 2: Results for ExamBC. BC corresponds to the system KDR-JA-ExamBC-02 that was submitted to the formal run. BC - X means all features of BC without feature X.

alignment strategy is that words that have a high degree of selectional preference in common should be aligned. This is similar to the work in [7] which suggests to use distributional similarity for word alignment. Note that our alignment strategy also includes antonyms like 売る ("sell") and 買う ("buy"), or named entities that belong to the same category but refer to different entities, like the countries イギリス ("England") and ドイツ ("Germany"). The cost of deleting nodes in T1 that cannot be matched to T2 are set to 1.0. We calculate the minimum alignment costs for each pair of T1 and T2's simple sentences. This way, we get a cost matrix which has the size of the number of simple sentence in T1 times the number of simple sentences in T2.

2.3 Calculating total minimum alignment costs between T1 and T2

We use the cost matrix from the previous step to find the globally best alignment of the simple sentences in T1 and T2. We consider this problem as the problem of finding the minimum cost alignment in a bipartite graph. The computation can be done using the Hungarian Algorithm. We denote the total minimum alignment cost as a_{total}.

2.4 Calculating degree of lexical contradiction

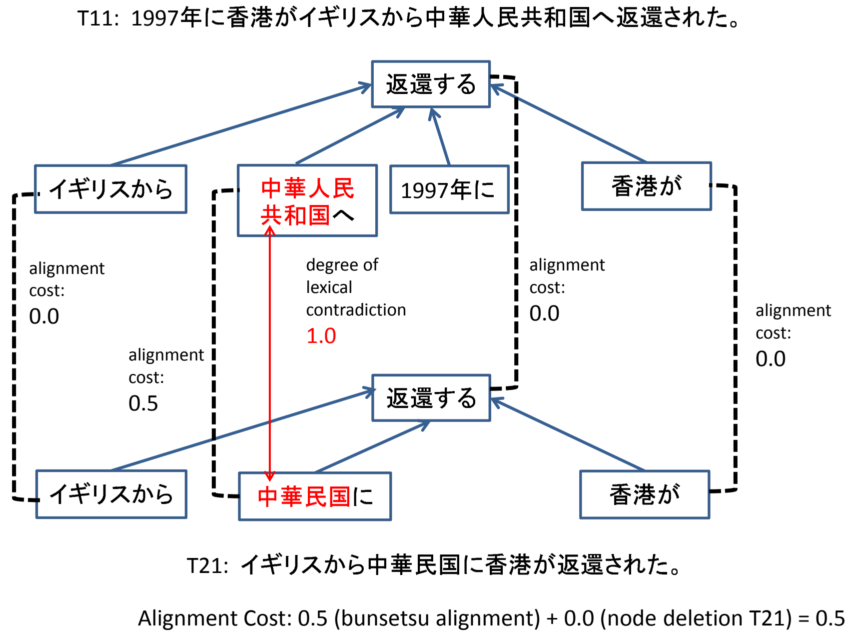
For each aligned chunks (bunsetsus), in each aligned simple sentence pair, we calculate the degree of lexical contradiction. The degree of lexical contradiction is defined in Table 3. For example, named entities, like イギリス ("England") and ドイツ ("Germany") have the degree of lexical contradiction 1.0. Furthermore, chunks where the head is the same but all other words are different like マンデラ大統領 ("President Mandela") and デクラーク大統領 ("President de Klerk") are considered to have degree of lexical contraction of 0.5. Obviously, the latter case does not always imply different entities consider for example the two spelling variations ラッダイト運動 ("Luddite movement") and ラダイト運動 ("Luddite movement").

2.5 Calculate the degree of contradiction between T1 and T2

Finally, we add up the degree of lexical contradiction for each chunk in each simple sentence pair. This way we get the total degree of lexical contradiction c_{total}. Obviously, lexical contradiction does not necessarily imply contradiction between the whole texts T1 and T2. We therefore adjust the total degree of lexical contradiction to get the degree of contradiction between T1 and T2, denoted as c_{text}.

$$c_{text} = \frac{c_{total}}{a_{total} + d}, \quad (1)$$

where $d > 0$ is set using the training data. For all of our experiments (ExamBC, ExamSearch) d was set to 10. The intuition for Formula (1) is that high structural similarity between text T1 and T2, in combination with (high) lexi-



<pair id="351" label="N">
<t1>1997 年に香港がイギリスから中華人民共和国へ返還された。</t1>
<t2>イギリスから中華民国に香港が返還された。</t2>

Figure 4: Shows the alignment of the simple sentence T11 (from T1) and T21 (from T2) from the pair 351 of the training data.

cal contradiction, implies, in general, that T1 and T2 are contradicting. Note that the closer the structural similarity between text T1 and T2 is, the lower the costs a_{total}. On the other hand, if the structural similarity is low, then lexical contradiction caused by a poor alignment is only little evidence for the contradiction between T1 and T2.

For example, consider the example given in Figure 4. The structural similarity between the simple sentences T11 and T21 is very similar suggesting that bunsetsus were aligned without changing the grammatical structure of the sentences. In this case, the lexical contrast between 中華人民共和国 ("People's Republic of China") and 中華民国 ("Republic of China") is likely to imply that T1 and T2 are contrasting. In this case $c_{text} = \frac{1.0}{0.5+10} = 0.10$.

On the other hand, consider the example given in Figure 5. The tree structures of the simple sentences T15 and T21 are dissimilar, suggesting that the bunsetsu alignment is little meaningful. As a consequence, the lexical contradiction between 義務教育推進運動 ("promotion campaign for compulsory education") and 反対運動 ("opposition campaign") is not likely to imply that T1 and T2 are contrasting. In this case $c_{text} = \frac{1.0}{3.0+10} = 0.08$.

We include c_{text} as one feature in our system. We name this feature Contradiction.

3. OTHER FEATURES

Here in this section we describe the other features that were used by our system.

We calculate the clipped precision between single characters analogously to [4], we name this feature Character Overlap. Furthermore we calculate also the clipped precision of the words (morphemes) that we were recognized by MeCab[1]. This feature is named Word Overlap.

Furthermore ExamBC and ExamSearch contains many statements about history. The match of a temporal expression in T1 and T2 can be an important clue that T2 is entailed by T1 (see [5]). There are many varieties to express temporal expressions in Japanese, including the name of the era or the use of the Western calendar. We therefore first normalize temporal expressions using normalizeNumexp[2]. In the second step we test for matching or mismatch of temporal expressions and include it as a feature. We name this feature Temporal Expressions.

Moreover, the feature Tree-edit Distance is calculated by using the method described in [8]. This method assumes that the order of siblings in the dependency tree is important. However, since Japanese is a free-order language we account for this effect by also calculating all permutations of the siblings in the dependency tree, and then take the minimum tree edit distance.[3]

For ExamSearch we also included as feature the overlap ratio of named entities (Named Entity). The named entities in T1 and T2 are recognized by using Cabocha [3]. The ratio corresponds to the percentage of T2's named entities that are found in T1.

Finally, for ExamSearch we also used the score of the retrieved sentences from Tsubaki (Tsubaki Score). The Tsubaki search results were provided by the organizers.

[1] http://mecab.googlecode.com/svn/trunk/mecab/doc/index.html
[2] http://www.cl.ecei.tohoku.ac.jp/katsuma/software/normalizeNumexp/
[3] In situations, where this is computationally too expensive we do not calculate all permutations.

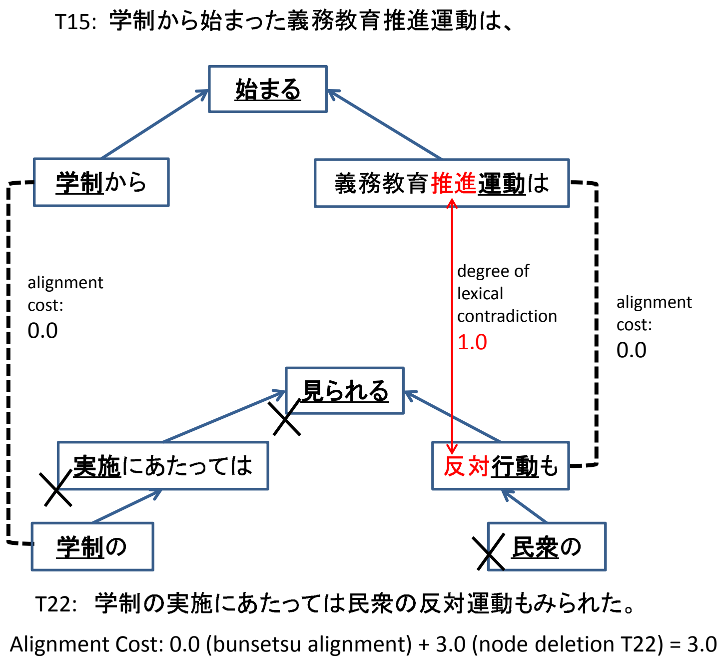


Figure 5: Shows the alignment of the simple sentence T15 and T22 from the pair 9 of the training data.

4. EXPERIMENTS

All three systems that were submitted to the ExamBC-Task use only 4 features:
Contradiction, Tree-edit distance, Character Overlap, Temporal Expressions.
Our best system BC (KDR-JA-ExamBC-02) combines these features using SVM with a radial basis kernel. For all of our ablation experiments we left the degree of regularization constant.[4] The impact of each of the four features is investigated in Table 2. The other two systems that we submitted to the ExamBC-Task, namely KDR-JA-ExamBC-01, and KDR-JA-ExamBC-03, use the same features as BC, but use as a machine learner Naive Bayes and the combination of Naive Bayes and SVM, respectively.

In Table 1 we show the results of the ExamSearch-Task. The first half shows the results when extracting the features only from search results of the Japanese history text book provided by the organizers. The second half shows the results when extracting the features from both the text book and the Japanese Wikipedia. We used the search results provided by the organizers. Our best system KDR-JA-ExamSearch-02, uses the same features as BC plus additionally the Tsubaki Score, Word Overlap and Named Entity. The other two systems, that were submitted, namely KDR-JA-ExamSearch-01 and KDR-JA-ExamSearch-03, differ only in the degree of regularization.

As can be seen in Table 2 and Table 1, the use of feature Contradiction constantly improves the textual entailment accuracy and macro F1-score. The improvement in F1-score (accuracy), for the systems that were submitted, ranges from 0.13 percent points, in ExamBC, to 2.36 percent points, in ExamSearch.

[4]We plan to provide a more thorough analysis in a future publication.

5. CONCLUSIONS

We introduced our systems, KDR-JA-ExamBC-02 and KDR-JA-ExamSearch-02, that became second and first place in the RITE2-ExamBC-Task (formal run, official result) and RITE2-ExamSearch-Task (formal run, unoffical result), respectively. Both systems employed a new method for detecting contradicting statements. We showed that this method contributes to the improvement in textual entailment recognition. Our method first aligns the chunks in text T1 and T2, and then measures the degree of lexical contradiction for each pair of chunks that were aligned. We then measure the degree of how likely it is that the lexical contradiction, observed on the chunk level, implies the contradiction of the texts T1 and T2. Our method assumes that the closer the dependency tree structures of T1 and T2 are, the likelier it is that (local) lexical contradiction propagates to (global) contradiction of the whole two texts.

6. ACKNOWLEDGMENT

We thank Luo Yanyan who greatly helped us implementing a baseline system during her internship at NEC Research.

7. REFERENCES

[1] M.-C. De Marneffe, A. N. Rafferty, and C. D. Manning. Finding contradictions in text. Proceedings of ACL-08: HLT, pages 1039–1047, 2008.

[2] R. Iida and M. Poesio. A cross-lingual ilp solution to zero anaphora resolution. In Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics, pages 804–813, 2011.

[3] T. Kudo and Y. Matsumoto. Fast methods for kernel-based text analysis. In Proceedings of the 41st Annual Meeting on Association for Computational

Linguistics-Volume 1, pages 24–31. Association for Computational Linguistics, 2003.

[4] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu. Bleu: a method for automatic evaluation of machine translation. In Proceedings of the 40th annual meeting on association for computational linguistics, pages 311–318. Association for Computational Linguistics, 2002.

[5] Y. Tsuboi, H. Kanayama, M. Ohno, and Y. Unno. Syntactic difference-based approach for the ntcir-9 rite task. In Proceedings of the 9th NII Test Collection for Information Retrieval Workshop (NTCIR ' 11), pages 404–411, 2011.

[6] Y. Watanabe, Y. Miyao, J. Mizuno, T. Shibata, H. Kanayama, C.-W. Lee, C.-J. Lin, S. Shi, T. Mitamura, N. Kando, H. Shima, and K. Takeda. Overview of the Recognizing Inference in Text (RITE-2) at NTCIR-10. In Proceedings of the 10th NTCIR Conference, 2013.

[7] Y. Watanabe, J. Mizuno, E. Nichols, K. Narisawa, K. Nabeshima, N. Okazaki, and K. Inui. Leveraging diverse lexical resources for textual entailment recognition. ACM Transactions on Asian Language Information Processing (TALIP), 11(4):18, 2012.

[8] K. Zhang and D. Shasha. Simple fast algorithms for the editing distance between trees and related problems. SIAM journal on computing, 18(6):1245–1262, 1989.