NTTCS Textual Entailment Recognition System for NTCIR-9 RITE

Yasuhiro AKIBA* Hirotoshi TAIRA Sanae FUJITA
Kaname KASAHARA Masaaki NAGATA

NTT Communication Science Laboratories
Hikaridai 2-4, Seika-cho, "Kansai Science City", Kyoto, 619-0237, JAPAN
*) akiba.yasuhiro@lab.ntt.co.jp

ABSTRACT

This paper describes initial Japanese Textual Entailment Recognition (RTE) systems that participated Japanese Binary-class (BC) and Multi-class (MC) subtasks of NTCIR-9 RITE. Our approaches are based on supervised learning techniques: Decision Tree (DT) and Support Vector Machine (SVM) learners. The employed features for the learners include text fragment based features such as lexical, syntactic, and semantic ones and new surface/deep case structure based features. These features are designed so as to assign entailment directions to a text pair in MC subtask. The authors submitted three runs to each of BC and MC subtasks. The best performance in the three runs achieves an accuracy of 0.548 in BC subtask and 0.452 in MC subtask, which were better than the averaged accuracy of all team submissions.

Team Name/ID

NTT CS Labs. / NTTCS

Subtasks/Languages

Japanese Binary-class (BC) and Multi-class (MC) subtasks

External Resources Used

As Japanese publicly available resources, (P1) ALAGIN monolingual language resources: Japanese hierarchical hypernym DB, Japanese cross-script/orthographic variation pair DB, Japanese WordNet, (P2) GoiTaikei: Japanese thesauri, (P3) NAIST Japanese Dictionary, (P4) a Japanese morphological analyzer — ChaSen, (P5) a Japanese Named Entity chunker/tagger — YamCha, (P6) a Japanese syntactic dependency parser — CaboCha. As Japanese in-house resources, (I1) Lexeed: a semantic lexicon, (I2) a base NP chunker/tagger, (I3) a surface case structure analyzer, (I4) a predicate argument structure analyzer.

Keywords

Classifiers, DT: Decision Trees, SVM: Support Vector Machines, Lexical normalization, Ratio of shared/unshared text fragments, Heuristic entailment rules, Entailment pattern superposition.

1. INTRODUCTION

This paper describes Textual Entailment Recognition (RTE) systems constructed from scratch to participate Japanese Binary-class (BC) and Multi-class (MC) subtasks of NTCIR-9 RITE [15]. This evaluation campaign of NTCIR-9 RITE is the first trial of RTE intended for Japanese. Unlike previous evaluation campaigns of PASCAL RTE Challenges (RTE1-6), the MC subtask preliminarily provides no information of entailment direction; if necessary, it requires to judge even entailment direction.

Existing approaches to English RTE systems include theorem prover based technique, transformation/similarity based technique, and supervised learning technique. Although each approach has its advantages and disadvantages, the authors took the approach based on supervised learning techniques using Decision Tree (DT) and Support Vector Machine (SVM) learners [13, 3], which are comparatively accessible.

In designing system architecture and features for the learners, the authors focus on three issues below: (1)errors by text analyzers, (2)lexical normalization, and (3) entailment direction judgment.

To encode an input of text pair into a feature space, our system utilizes NLP tools such as morphological analyzers and syntactic dependency parsers as in existing RTE systems. Because the longer text's analysis results by NLP tools are likely to contain the more analysis errors in totality, smaller text fragment are utilized in matching them in lexical, syntactic, and semantic level.

To match two different language expression with the same sense, our system refers to normalization forms of NAIST Dictionary[1] and ALAGIN's three language resources: variation pair DB[9, 12], Japanese WordNet[2], hypernymy DB[16, 10, 12]. Ratios of shared/ unshared text fragments are employed as features. Novel features are, moreover, innovated by using two in-house analyzers: a surface case structure analyzer and a predicate argument structure analyzer[17]. These features are designed so as to assign entailment directions to a text pair in MC subtask.

The authors submitted three runs to each of BC and MC subtasks on test data. For the 1st and 2nd runs, DT and SVM classifier [13, 3] were, respectively, learned on the provided training data, with all the features. For the 3rd run, SVM classifier was learned on the provided training data, with all the features except some features based on surface/deep case structure. For each of BC and MC subtasks, the 3rd run performs best in the three runs, which achieves an accuracy of 0.548 in BC subtask and 0.452 in MC subtask. These accuracy are better than the averaged accuracy of all team submissions.

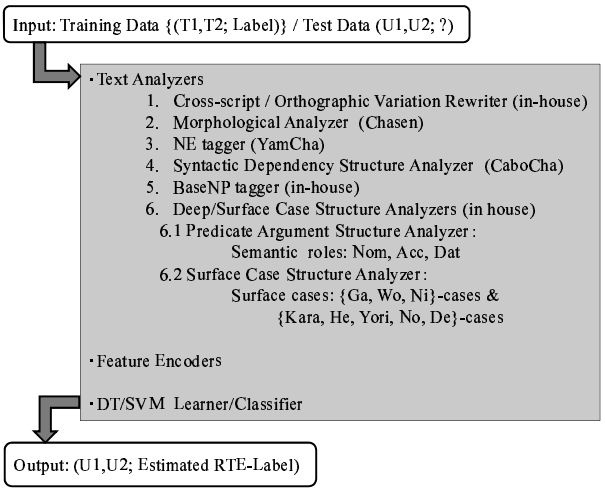

Figure 1: Overview of NTTCS RTE system for NTCIR-9 RITE

After formal run, the authors conducted ablation tests for the 1st and 2nd run to BC and MC subtasks on test data. The results of ablation tests depend on combination of a subtask and a learner. It is difficult to make unified assertions of which tools or resource contribute to the accuracy.

For the combination of BC subtask and SVM learner, in the cases of using all the features except a feature based on surface case structure or except a feature based on synset overlapping, such runs achieved a slightly better accuracy while the difference of accuracies may be in error range.

For the combination of MC subtask and DT learner, in the cases of using all the features except a unmatched predicate based feature, such runs achieved a slightly better accuracy while the difference of accuracies may be in error range.

The next section outlines our RTE systems and describes in-house resources and the employed features. Results of the runs and ablation tests are shown and discussed in Section 3. Finally, our conclusions are presented in Section 4.

2. SYSTEM DESCRIPTION

Our RTE systems are outlined in Section 2.1; and Section 2.2 presents the features utilized in our RTE systems.

2.1 Overview

As reported in the previous section, our RTE system for NTCIR-9 RITE [15] are based on supervised learning techniques: Decision Tree (DT) and Support Vector Machine (SVM) learners [13, 3]. Figure 1 sketches out rough data flow in our RTE systems. Let $(T1, T2)$ or $(U1, U2)$ denote a text pair in training or test data, respectively. To encode an input text pair into a feature space, each text in pair involved by text analyzers 1 to 6 in listed in Figure 1.

The first analyzer rewrite/transform each input text by using ALAGIN's Japanese cross-script/orthographic variation pair DB[9, 12] so as to realize lexical normalization touched in Section 1.

After this rewriting, morphological analyzer: ChaSen[11], NE tagger: YamCha[8], non-typed syntactic dependency structure analyzer: CaboCha[7] are followed as usual. YamCha in our RTE systems assigns an NE label in the IREX project[14]. Note that unlike English syntactic dependency structure analyzer, Japanese syntactic dependency structure analyzer does not label a dependency type such as subject or indirect object to each dependency relation.

As the fifth text analyzer, in-house base NP Chunker/ Tagger [18] followed the CaboCha dependency parser. Base NP Tagger assigns a supersense category in Japanese thesauri of GoiTaikei [5] to base NP Chunks.

Finally, the two in-house deep/surface case structure analyzers are conducted. The former is a predicate argument structure (PAS) analyzer[17]; and the latter is a surface case structure (SCS) analyzer. Because the PAS analyzer were trained on NAIST Text Corpus [4], the labeled semantic/case roles are restricted to Ga-case (nominative), Wo-case (accusative), and Ni-case (dative). To extract information corresponding to other semantic roles such as locative or time case, the SCS analyzer extracts surface cases based on eight case markers: {Ga, Wo, Ni, Kara, He, Yori, No, De}-particles.

2.2 Features

This section presents the features utilized in our RTE systems. which include text fragment based features such as lexical, syntactic, and semantic ones and new surface/deep case structure based features. These features are designed so as to assign entailment directions to a text pair in MC subtask. Table 1 shows a summary of the features below.

Lexical match / unmatch features in surface level

Lexical match features in surface level are defined as normalized number of *shared tokens* with certain POS tags. Tokens are compared between normalized/base forms. The normalized form are preliminarily selected from base form variations, which are defined in NAIST Japanese Dictionary [1].

The normalized number of shared tokens in this paper is calculated as the number of shared tokens divided by the number of all tokens in either of each text pair, while usual denominator is the number of tokens only in each hypothesis.

In finding shared tokens, their POS is restricted to one of four choice below: (0) All token except Particles, Auxiliary verbs, and Symbols, (1) Nouns, (2) Verbs, or (3) Adjectives.

Lexical unmatch features in surface level are defined as with the above lexical match features in surface level. In the definition of lexical unmatch features in surface level, Phrase "shared tokens" are substituted with Phrase "unshared tokens". Moreover, in the calculation of the normalized number of unshared tokens, the denominator is the number of tokens only in each hypothesis.

Lexical match features in semantic level

Lexical match features in semantic level are defined normalized numbers of shared tokens with certain POS tags, as with the above lexical match features in surface level.

In these features, tokens are compared between synsets. Each Token belongs to one or more synsets. If compared tokens share a certain synset, such tokens successes to match. Their POS restricted is one of three choice below: (1) Nouns, (2) Verbs, or (3) Adjectives.

Bigram/Trigram match features

Bigram/Trigram match features are defined as with above lexical match features in surface level. In the definition of bigram/trigram match features, Phrase "shared tokens" are substituted with Phrase "shared N-grams ($N = 2$ *or* 3)".

Title	Brief description	Forcused types	Normalized by	Value range	# of features
Lexical match features in surface level	# of shared token in surface level	Content Words, Nouns, Verbs, Ajectives	# of token in T1 or T2	[0,1]	8 (=4*2)
Lexical unmatch features in surface level	# of unshared token in surface level	Content Words, Nouns, Verbs, Ajectives	# of token in T2	[0,1]	4 (=4*1)
Lexical match features in semantic level	# of shared token in semantic level	Nouns, Verbs, Ajectives	# of token in T1 or T2	[0,1]	6 (=3*2)
Bigram/Trigram match features	# of shared Ngram in surface level	N=2,3 (Bigram, Trigeram)	# of Ngram in T1 or T2	[0,1]	4 (=2*2)
Chunk match Features	# of shared Chunks	BaseNP, NE	# of chunks in T1 or T2	[0,1]	4 (=2*2)
Chunk unmatch Features	# of unshared Chunks	BaseNP, NE	# of chunks in T2	[0,1]	2 (=2*1)
Syntactic dependency relation match features	# of shared dependency bigram	With/Without ignoring particles	# of dependency bigram in T1 or T2	[0,1]	2 (=2*2)
Case structure match features	5-dim bit vector	SCS, PAS	-	{0,1}^5	10 (=5*2)
Predicate unmatch features (1)	Difference between numbers of predicates	SCS, PAS	-	Integer	2
Predicate unmatch features (2)	# of unshared predicates	SCS, PAS		Integer	2
Predicate unmatch features (3)	# of predicate pairs in antonym relation	SCS, PAS	-	Integer	2

Table 1: Summary of features

There are no restriction of POS tags.

Chunk match / unmatch features

Chunk match features are defined as with above lexical match features in surface level. In the definition of Chunk match features, Phrase "shared tokens" are substituted with Phrase "shared chunks (Base NPs / NEs)". Chunks are compared between sequences of normalized/base forms and their categories. Base NP chunks are assigned to a supersense category in Japanese thesauri of GoiTaikei [5] by the in-house base NP Tagger [18]. NE chunks are assigned to an NE category [14] by Yamcha NE Tagger [8]. Number of shared tokens are normalized as with above lexical match features in surface level

Chunk unmatch features are defined as with the lexical unmatch features. In the definition of chunk unmatch features, Phrase "unshared tokens" are substituted with Phrase "unshared chunks". Moreover, in the calculation of the normalized number of unshared chunks, the denominator is the number of chunks only in each hypothesis.

Syntactic dependency relation match feature

Chunk match features are defined as normalized numbers of shared dependency bigram. Dependency bigram is a modifier-modifiee pair of Bunsetu-chunks. A Bunsetsu-chunk consists of one or more content words followed by zero or more function words. Japanese syntactic dependency parser finds modifier-modifiee pairs of Bunsetu-chunks.

Dependency bigrams are compared between sequences of all normalized/base forms, or between sequences of all normalized/base forms except functional words. Note that dependency types are not used in this comparison.

In the calculation of the normalized number of unshared chunks, the denominator is the number of dependency bigrams in either of each text pair.

Case Structure match features

By using the resulting case structures from the PAS/SCS analyzers, five-dimensional bit vectors are calculated bellow. Each bit in a bit vector corresponds to one of the five entailment labels: B, F, R, C, I in BC subtask.

(1) Find the corresponding predicate pairs $(P1, P2)$ in input text pair $(T1, T2)$ such that predicates $P1$ from $T1$ and $P2$ from $T2$ have the same normalized/base forms, or synonym/hyponymy relations [2, 16, 10, 12].

(2) For each corresponding predicate pairs $(P1, P2)$, compare slots for each case and assign a bit vector as follows:

- **(2.1)** if noun phrases in the case slots has the same normalized/base forms, assign a bit vector whose only the bit for entailment label 'B' is 1 and the other bits are 0,
- **(2.2)** if the case slot of predicate $P2$ is empty, assign a bit vector whose only the bit for entailment label 'F' is 1 and the other bits are 0,
- **(2.3)** if the case slot of predicate $P1$ is empty, assign a bit vector whose only the bit for entailment label 'R' is 1 and the other bits are 0,
- **(2.4)** if noun phrases in the case slot has a antonym relation [6] assign a bit vector whose only the bit

BC Subtask		MC Subtask	
Runs	Accuracy	Runs	Accuracy
Best	0.580	Best	0.511
Run 3	0.548	Run 3	0.452
Run 1	0.532	Run 1	0.448
Average	0.521	Average	0.407
Run 2	0.520	Run 2	0.405

Table 2: Formal run results of our RTE system

for entailment label 'C' is 1 and the other bits are 0,

(2.5) if noun phrase in the case slot of predicate $P1$ has a hyponym of noun phrase in the case slot of predicate $P2$ assign a bit vector whose only the bit for entailment label 'F' is 1 and the other bits are 0,

(2.6) if noun phrase in the case slot of predicate $P2$ has a hyponym of noun phrase in the case slot of predicate $P1$ assign a bit vector whose only the bit for entailment label 'R' is 1 and the other bits are 0,

(2.7) if none of the above cases (2.1)–(2.6) are applied, assign a bit vector whose only the bit for entailment label 'I' is 1 and the other bits are 0,

(3) For each corresponding predicate pairs $(P1, P2)$, calculate logical OR of the bit vectors assigned to all the cases,

(4) For test pair $(T1, T2)$, calculate logical OR of the bit vectors assigned to all corresponding predicate pairs.

Case Structure match features are bits in the resulting vector.

Predicate unmatch features

Three predicate unmatch features are defined as follows:
(1)Difference between numbers of predicates,
(2)Number of unmatched predicates in T1 / T2, and
(3)Number of predicate pairs in antonym relation [6].

3. FORMAL RUN

This section shows formal run setup and discusses the formal run results and ablation tests.

3.1 Setup and Submission results

The authors submitted three runs to each of BC and MC subtasks on test data. For the 1st and 2nd runs, DT and SVM classifier were, respectively, learned on the provided training data, with all the features. For the 3rd run, SVM classifier was learned on the provided training data, with all the features except some features based on surface/deep case structure. The DT classifiers were trained by C4.5 [13] with the default options. The SVM classifiers were trained by nu-svm of LIBSVM [3]. with 2nd polynomial kernel

Table 2 shows the formal run results of our RTE system. For each of BC and MC subtasks, the 3rd run performed best in the three runs, which achieves an accuracy of 0.548 in BC subtask and 0.452 in MC subtask. These accuracy are better than the averaged accuracy of all team submissions.

3.2 Ablation tests and their discussion

After formal run, the authors conducted ablation tests for the 1st and 2nd run to BC and MC subtasks on test data. Table **??** shows sample results of ablation tests.

The results of ablation tests depend on combination of a subtask and a learner. It is difficult to make unified assertions of which tools or resource contribute to the accuracy.

For the combination of BC subtask and SVM learner, in the cases of using all the features except a feature based on surface case structure or except a feature based on synset overlapping, such runs achieved a slightly better accuracy while the difference of accuracies may be in error range.

For the combination of MC subtask and DT learner, in the cases of using all the features except a unmatched predicate based feature, such runs achieved a slightly better accuracy while the difference of accuracies may be in error range.

4. CONCLUSIONS

This paper describes initial Japanese Textual Entailment Recognition (RTE) systems that participated Japanese Binary-class (BC) and Multi-class (MC) subtasks of NTCIR-9 RITE. Our approaches are based on supervised learning techniques: Decision Tree (DT) and Support Vector Machine (SVM) learners. These features are designed so as to assign entailment directions to a text pair in MC subtask. These accuracy of our formal run results are better than the averaged accuracy of all team submissions.

After formal run, the authors conducted ablation tests for the 1st and 2nd run to BC and MC subtasks on test data. The results of limited ablation tests depend on combination of a subtask and a learner. It is difficult to make unified assertions of which tools or resource contribute to the accuracy. The authors need to conducts further experiment and elaborate analysis in future.

5. ACKNOWLEDGMENTS

The authors give special thanks to Mr. Eiki FUJIMOTO for cooperating the development of our RTE system.

6. REFERENCES

[1] M. Asahara and Y. Matsumoto. NAIST Japanese dictionary version 0.4.0 users manual. Technical report, NAIST, 2008.

[2] F. Bond, H. Isahara, S. Fujita, K. Uchimoto, T. Kuribayashi, and K. Kanzaki. Enhancing the Japanese WordNet. In *Proc of The 7th Workshop on Asian Language Resources, in conjunction with ACL-IJCNLP2009*, 2009.

[3] C. C. Chang and C. J. Lin. Libsvm – a library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2(3), 2011.

[4] R. Iida, M. Komachi, K. Inui, and Y. Matsumoto. Annotating a japanese text corpus with predicate-argument and coreference relations. In *Proc. of ACL 2007 Workshop on Linguistic Annotation*, pages 132–139, 2007.

[5] S. Ikehara, M. Miyazaki, S. Shirai, A. Yokoo, H. Nakaiwa, K. Ogura, Y. Ooyama, and Y. Hayashi.

BC Subtask by SVM			MC Subtask by DT		
Runs	Accuracy	Difference	Runs	Accuracy	Difference
Best	0.580		Best	0.511	
Run2 - SynsetOverlap	0.554	0.034	Run 1 - PAS Unmatch	0.484	0.036
Run2 - SCS Match	0.552	0.032	Run 1 - PAS Match	0.468	0.020
Run3	0.548		Run 1 - SCS Unmatch	0.461	0.013
Run2 - SCS Analyzer	0.546	0.026	Run 3	0.452	
Run2 - PAS Unmatch	0.532	0.012	Run 1	0.448	
Average	0.521		Run 1 - SCS Analyzer	0.441	-0.007
Run2	0.520		Run 1 - SynsetOverlap	0.436	-0.012
Run2 - PAS Analyzer	0.514	-0.006	Run 1 - SCSMatch	0.432	-0.016
Run2 - PAS Match	0.510	-0.010	Run 1 - PAS	0.425	-0.023
Run2 - SCS Unmatch	0.506	-0.014	Average	0.407	

Table 3: Ablation test results of our RTE system

Goi-Taikei — A Japanese Lexicon. Iwanami Shoten, Tokyo, 1997.

[6] K. Kasahara, H. Sato, F. Bond, T. Tanaka, S. Fujita, T. Kanasugi, and S. Amano. Construction of a Japanese semantic lexicon: Lexeed (in Japanese). In *IPSJ SIG Technical Reports*, volume NL-159, pages 75 – 82, 2004.

[7] T. Kudo and Y. Matsumoto. Japanese dependency analysis using cascaded chunking. In *Proc. of CoNLL2002: The 6th Conference on Natural Language Learning 2002 (COLING 2002 Post-Conference Workshops)*, pages 63–69, 2002.

[8] T. Kudo and Y. Matsumoto. Fast methods for kernel-based text analysis. In *Proc. of ACL-2003: The 41st Annual Meeting of the Association for Computational Linguistics*, pages 24–31, 2003.

[9] K. Kuroda, J. Kazama, M. Murata, and K. Torisawa. Standards for certifying Japanese cross-script variation pairs capable of handling even web documents, a5-6 (in Japanese). In *Proc. of NLP-2010: The Sixteenth Annual Meeting of The Association for Natural Language Processing*, 2002.

[10] K. Kuroda, M. Murata, and K. Torisawa. When nouns need co-arguments: A case study of semantically unsaturated nouns. In *Proc. of the 5th International Workshop on Generative Approaches to the Lexicon*, pages 193–200, 2009.

[11] Y. Matsumoto, A. Kitauchi, T. Yamashita, Y. Hirano, O. Imaichi, , and T. Imamura. Japanese morphological analysis system chasen manual. Technical Report NAIST-IS-TR97007, NAIST, 1997.

[12] S. Nakamura, K. Torisawa, H. Kawai, and E. Sumita. Nict speech and language resources and corpora. In *Proc. of Oriental COCOSDA 2010*, 2010.

[13] J. R. Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann Publishers, 1993.

[14] S. Sekine and H. Isahara. Irex: Ir and ie evaluation project in Japanese. In *Proc. of LREC2000: The 2nd International Conference on Language Resources and Evaluation*, 2000.

[15] H. Shima, H. Kanayama, C.-W. Lee, C.-J. Lin, T. Mitamura, Y. Miyao, S. Shi, and K. Takeda. Overview of ntcir-9 rite: Recognizing inference in TExt. In *Proc. of NTCIR9: The 9th NTCIR Workshop Meeting Evaluation of Information Access Technologies: Information Retrieval, Question Answering, and Cross-Lingual Information Access*, 2011.

[16] A. Sumida, N. Yoshinaga, and K. Torisawa. Boosting precision and recall of hyponymy relation acquisition form hierarchical layouts in wikipedia. In *Proc. of LREC2008: The 6th International Lexical Resources and Evaluation*, 2008.

[17] H. Taira, S. Fujita, and M. Nagata. A Japanese predicate argument structure analysis using decision lists. In *Proc. of EMNLP2008: Conference on Empirical Methods in Natural Language Processing*, pages 523–532, 2008.

[18] H. Taira, S. Yoshida, and M. Nagata. Basenp supersense tagging for Japanese texts. In *Proc. of PACLIC23: The 23rd Pacific Asia Conference on Language, Information and Computation*, volume 2, pages 819 – 826, 2009.