KSU Team's System and Experience at the NTCIR-11 RITE-VAL Task

Tasuku Kimura
Kyoto Sangyo University, Japan
i1458030@cse.kyoto-su.ac.jp

Hisashi Miyamori
Kyoto Sangyo University, Japan
miya@cse.kyoto-su.ac.jp

ABSTRACT

This paper describes the systems and results of the team KSU for RITE-VAL task in NTCIR-11. Three different systems were implemented for each of the two subtasks: Fact Validation and System Validation. In Fact Validation subtask, systems were designed respectively based on character overlap, existence of entailment result 'Y', and voting of entailment results. In System Validation subtask, systems were designed respectively using SVM, Random Forest, and Bagging, with features such as surface features, numerical expressions, location expressions, and named entities. Scores of the formal runs were 52.78% in macro F1 and 66.96% in accuracy with KSU-FV-02 in Fact Validation, and 66.96% in macro F1 and 79.84% in accuracy with KSU-SV-01 in System Validation. Also, in System Validation, scores of the unofficial runs were 67.18% in macro F1 and 76.50% in accuracy with KSU-SV-03-C.

Team Name

KSU

Subtasks

RITE-VAL FV,SV (Japanese)

Keywords

Surface features, Generalized overlap ratio, Recognizing textual entailment in top search results

1. INTRODUCTION

Textual entailment recognition is, given a text pair t_1 and t_2, a problem of recognizing whether text t_1 entails t_2. It has attracted the attention of many researchers in recent decades as one of the fundamental technologies that can be applied to various information access technologies such as question and answering, document summarization, and information retrieval.

In RITE1 at NTCIR-9, several subtasks were set, which require inference at single sentence level, deciding binary or multiple classes, given a sentence pair t_1 and t_2[8]. In RITE2 at NTCIR-10, new subtasks were introduced besides conventional subtasks, which require inference using multiple sentences retrieved from Wikipedia and textbooks, and which necessitate inference at linguistic phenomena cooccurring with entailment[9].

In RITE-VAL at NTCIR-11[6], two new subtasks were set to enhance the two subtasks newly introduced in RITE2 of NTCIR-10: Fact Validation subtask, where inference is needed using multiple sentences obtained by information retrieval, and System Validation subtask, where inference is necessary at detailed linguistic phenomena cooccurring with entailment.

This paper describes the systems and results of the team KSU for RITE-VAL task in NTCIR-11. Three different systems were implemented for each of the two subtasks: Fact Validation and System Validation. In Fact Validation subtask, systems were designed respectively based on character overlap, existence of entailment result 'Y', and voting of entailment results. In System Validation subtask, systems were designed respectively using SVM, Random Forest, and Bagging, with features such as surface features, numerical expressions, location expressions, and named entities. Scores of the formal runs were 52.78% in macro F1 and 66.96% in accuracy with KSU-FV-02 in Fact Validation, and 66.96% in macro F1 and 79.84% in accuracy with KSU-SV-01 in System Validation. Also, in System Validation, scores of the unofficial runs were 67.18% in macro F1 and 76.50% in accuracy with KSU-SV-03-C.

2. FACT VALIDATION

In Fact Validation, it is necessary, without t_1, to identify whether t_2 is entailed or not from relevant sentences obtained in search results using t_2. We designed systems based respectively on character overlap ratio, existence of entailment result 'Y', and voting of entailment results. In order to identify entailment, we referred to the system RITE2-SKL-MC-01, which gave best performance in MC subtasks at RITE2 as a base system[4]. Also, we used search results provided by organizers, obtained from a textbook of World/Japanese History using TSUBAKI search engine.

2.1 Features

2.1.1 *Surface Similarity using Generalized Overlap Ratio*

First of all, a function is defined, which calculates how many number of entities are overlapped between strings t_1 and t_2, as follows:

$$overlap(E; t_1, t_2) = \Sigma_{x \in E} min(fr(x, t_1), fr(x, t_2)) \qquad (1)$$

where E denotes a set of entities and $fr(x, s)$ represents a function calculating frequencies of x in a given string s.

Using the above function, two kinds of overlap ratios are defined as follows:

$$overlap_D(E; t_1, t_2) = \frac{overlap(E; t_1, t_2)}{\Sigma_{x \in E} fr(x, t_2)} \quad (2)$$

$$overlap_B(E; t_1, t_2) = \frac{2overlap(E; t_1, t_2)}{\Sigma_{x \in E} fr(x, t_1) + \Sigma_{x \in E} fr(x, t_2)} \quad (3)$$

where $overlap_D$ is a directional function used when identifying entailment, and $overlap_B$ means a bidirectional function used in detecting contradictions.

Using these generalized overlap ratios, character overlap ratio, character bigram overlap ratio, and kanji-katakana character overlap ratios are respectively defined as follows:

$$cor_D(t_1, t_2) = overlap\_ratio_D(C; t_1, t_2) \quad (4)$$

$$bor_D(t_1, t_2) = overlap\_ratio_D(C^2; t_1, t_2) \quad (5)$$

$$kor_D(t_1, t_2) = overlap\_ratio_D(K; t_1, t_2) \quad (6)$$

where C denotes a set of all characters in Japanese texts, and K expresses a union of Kanji and Katakana character sets.

2.1.2 Named Entity Mismatch

$NE\_mismatch()$ returns mismatch of named entities in two sentences t_1 and t_2. It returns true when t_2 contains named entities not included in t_1, and returns false otherwise. Named entities were obtained using JUMAN as morphemes with syntax categories ''named entities" or with feature labels ''automatically retrieved from Wikipedia"

2.1.3 Number Expression Mismatch

$Num\_mismatch()$ returns mismatch of numerical expressions in two sentences t_1 and t_2. It returns true when t_2 contains numerical expressions not included in t_1, and returns false otherwise. Numerical expressions were extracted using JUMAN as morphemes with ''numerical quantities" in bunsetsu features.

2.1.4 String Decomposition into Three Parts

From a sentence pair t_1 and t_2, the longest common prefix h and the longest common suffix t are identified, decomposing the pair into three parts as follows:

$$t_1 = h + b_1 + t \quad (7)$$

$$t_2 = h + b_2 + t \quad (8)$$

where b_1 and b_2 represent the body parts subtracted h and t from t_1 and t_2, respectively.

ht_ratio is defined as follows:

$$ht\_ratio = \frac{2\,(|h| + |t|)}{|t1| + |t2|} \quad (9)$$

2.2 KSU-JA-FV-01

KSU-JA-FV-01 is based on character overlap ratio using top documents obtained from search results. Figure 1 shows a pseudo-code for KSU-JA-FV-01.

For each of top l sentences of top k documents obtained from search results, the character overlap ratios are calculated between t_2 and the sentence which is seen as t_1, and identified entailment according to the ratio. The threshold $thresh$ were set to 0.6, and k and l were set as follows: $k = 5, l = 5$.

Algorithm 1 KSU-JA-FV-01

```
Require: top_docs, t2
  label = 'N'
  max_cor = 0
  for doc in top_docs do
    for t1 in top_sentences[doc] do
      cor = cor_D(t1, t2)
      if cor > max_cor and cor > thresh then
        max_cor = cor
        label = 'Y'
      end if
    end for
  end for
  return label
```

2.3 KSU-JA-FV-0[2,3]

First, figure 3 shows a pseudo-code of the MC system which bases KSU-JA-FV-02 and KSU-JA-FV-03.

Algorithm 2 Base-MC

```
Require: t1, t2
  if contradict(t1, t2) then
    return 'C'
  else if Base-BC(t1, t2) = 'Y' then
    if Base-BC(t2, t1) = 'Y' then
      return 'B'
    else
      return 'F'
    end if
  else
    return 'I'
  end if
```

Algorithm 3 Base-BC

```
Require: t1, t2
  if cor_D(t1, t2) ≥ 0.73 or (kor_D(t1, t2) > cor_D(t1, t2) ≥
  0.69) or ((0.69 > cor_D(t1, t2) > 0.65) and (kor_D(t1, t2) −
  0.1 > cor_D(t1, t2))) then
    if NE_mismatch(t1, t2) or Num_mismatch(t1, t2)
    then
      return 'N'
    else
      return 'Y'
    end if
  else
    return 'N'
  end if
```

KSU-JA-FV-02 is based on existence of entailment result 'Y' for top documents obtained from search results. Figure 4 shows a pseudo-code for KSU-JA-FV-02.

For each of top l sentences of top k documents obtained from search results, the sentence is seen as t_1 and the base MC is used to identify entailment. If one of the results include 'F' or 'B', it returns 'Y', and 'N' otherwise. When the system label is 'Y', the first document returning 'Y' is output as the t_1 documents. When the system label is 'N', the t_1 documents are decided as follows: If one of the results from the base system contains 'C', the documents returning 'C' are set as the t_1 documents. Otherwise, all the documents

Algorithm 4 KSU-JA-FV-02

```
Require: top_docs, t2
  decision_flag = false
  for doc in top_docs do
    for t1 in top_sentences[doc] do
      label = Base-MC(t1, t2)
      if label == 'F'or label == 'B' then
        decision_flag = true
        break
      end if
    end for
    if decision_flag == true then
      break
    end if
  end for
  if decision_flag == true then
    return 'Y'
  else
    return 'N'
  end if
```

returning 'I' are set as the t_1 documents.

KSU-JA-FV-03 is based on voting of entailment results for top documents obtained from search results. Figure 5 shows a pseudo-code for KSU-JA-FV-03.

For each of top l sentences of top k documents obtained from search results, the sentence is seen as t_1 and the base MC is used to identify entailment. If one of the results include 'F' or 'B', it votes for 'Y', and if it contains 'C', it votes for 'N'. Otherwise it votes for nothing. When the system label is 'Y', the first document returning 'Y' is output as the t_1 documents. When the system label is 'N', the t_1 documents are decided as follows: If one of the results from the base system contains 'C', the documents returning 'C' are set as the t_1 documents. Otherwise, all the documents returning 'I' are set as the t_1 documents.

Algorithm 5 KSU-JA-FV-03

```
Require: top_docs, t2
  decision_flag = false
  for doc in top_docs do
    initializefreq
    for t1 in top_sentences[doc] do
      label = Base-MC(t1, t2)
      if label == 'F'or label == 'B' then
        freq['Y'] += 1
      else if label == 'C' then
        freq['N'] += 1
      end if
    end for
    if freq['Y'] >= freq['N'] then
      label = 'Y'
    else
      label = 'N'
    end if
    if label == 'Y' then
      decision_flag = true
      break
    end if
  end for
  if decision_flag == true then
    return 'Y'
  else
    return 'N'
  end if
```

3. SYSTEM VALIDATION

In System Validation, it is necessary to identify whether t_2 is entailed or not in linguistic phenomena related to entailment. We referred to the system RITE2-FLL-JA-UnitTest-01, which gave best performance in UnitTest subtasks in RITE2[5]. We designed systems respectively based on SVM, Random Forest, and Bagging, with features such as surface features, numerical expressions, location expressions, and named entities.

3.1 Features

Below shows the features used in our system.

3.1.1 Surface Features

The following surface features were used for a given sentence pair t_1 and t_2.

Cosine similarity of content words Let w_1 and w_2 be the sets of content words included in t_1 and t_2 respectively. The cosine similarity of content words are calculated as follows:

$$cos\_sim\_w = \frac{|w_1 \cap w_2|}{|w_1||w_2|}$$

Cosine similarity of characters Let c_1 and c_2 be the sets of characters included in t_1 and t_2 respectively. The cosine similarity of characters are calculated as follows:

$$cos\_sim\_c = \frac{|c_1 \cap c_2|}{|c_1||c_2|}$$

Jaccard coefficient of content words Let w_1 and w_2 be the sets of content words in t_1 and t_2 respectively. The Jaccard coefficient of content words are calculated as follows:

$$jaccard\_coeff\_w = \frac{|w_1 \cap w_2|}{|w_1| \cup |w_2|}$$

Longest common subsequence The longest common subsequence is the longest substrings common to t_1 and t_2. Here, the value of LCS is normalized by the length of t_2.

3.1.2 Numerical Expression-based Features

The following numerical expression-based features were used for a given sentence pair t_1 and t_2.

numexp_exact It represents whether all the numerical expressions N_2 in t_2 are exactly included in the numerical expressions N_1 in t_1. If N_2 expresses ranges, the ranges should be the same as those in N_2.

numexp_n2subset It represents whether the numerical expressions N_2 in t_2 are partially included in N_1 in t_1. This feature is used when some of the numerical expressions N_2 are partially contained in N_1 and the numerical values in N_2 are exactly included in N_1.

numexp_n1subset It expresses whether all the numerical expressions N_1 are contained in N_2.

Table 1: Results of our runs for FV subtask

System	Macro F1	Accuracy
NUL-JA-FV-03 (1st)	61.47	62.84
NUL-JA-FV-01 (2nd)	59.94	61.67
NUL-JA-FV-05 (3rd)	59.67	61.87
KSU-JA-FV-02	52.78	63.42
KSU-JA-FV-03	52.42	63.23
KSU-JA-FV-01	50.61	50.97

Table 2: Results of our runs for SV subtask

System	Macro F1	Accuracy
NUL-JA-SV-04 (1st)	69.59	77.81
NUL-JA-SV-05 (2nd)	68.94	77.96
NUL-JA-SV-01 (3rd)	68.73	77.81
KSU-JA-SV-01	66.96	79.84
KSU-JA-SV-03	65.72	75.78
KSU-JA-SV-02	64.87	76.00

numexp_diff It describes whether one or more numerical expressions exist in N_2 which do not match with the numerical expressions in N_1.

The value of each feature above was set as "missing", if either the numerical expressions N_1 in t1 or N_2 in t2 became an empty set.

3.1.3 Location Features

The following location feature was used for a given sentence pair t_1 and t_2. The value of the feature was set as "missing", if either the location names in t1 or those in t2 were found empty.

location It represents whether location names described in t_2 are also referred to in t_1.

3.1.4 Named Entity Features

The following named entities features were used for a given sentence pair t_1 and t_2.

ne_n2subset It indicates whether all the named entities NE_2 in t_2 are included in NE_1 in t_1.

ne_diff It represents whether a named entity exist in NE_2 which is not included in NE_1.

ne_cos_sim It represents the cosine similarity between NE_1 and NE_2.

The value of each feature above was set as "missing", if either the named entities NE_1 in t1 or NE_2 in t2 were found an empty set.

3.2 System Description

The systems were implemented using the above features with the following learning methods:

KSU-SV-JA-01 SVM[3] were applied to the system using poly kernel.

KSU-SV-JA-02 Random forest[2] were applied to the system. The number of trees were set to 150.

KSU-SV-JA-03 Bagging[1] were applied to the system. REPTree[7] were used as a classifier.

4. RESULTS AND DISCUSSION

The results of our systems for Fact Validation subtask and System Validation subtask were shown in tables 1 and 2.

Table 3: Degree of coincidence between correct t1 documents and those selected as t1 during each run or by TSUBAKI

System	Precision	Recall	F-measure	MAP
FV-01	0.00783	0.00783	0.00006	0.00392
FV-02	0.01364	0.01957	0.00012	0.00740
FV-03	0.01364	0.01957	0.00012	0.00740
TSUBAKI	0.01364	0.02677	0.00014	0.00854

4.1 Fact Validation

In FV subtask, our systems showed less favorable results compared to the top three results. Our systems were found to be weak in identifying 'Y' correctly, considering that the accuracies of our systems were similar to those of the top three systems, and that the number of 'N' samples were larger than that of 'Y' both in training and test sets.

To evaluate the validity of the selected documents as t_1, we calculated the degree of coincidence between the correct t_1 documents provided by the organizers and those selected as t_1 during each run. The degree of coincidence between the correct t_1 documents and those obtained from a textbook of World/Japanese History by TSUBAKI search engine was also estimated. The results are shown in table 3. We also computed the degrees of agreement with 129 documents labeled as 'Y' among the total of the 132 correct t_1 documents provided by the organizers. The result is given in table 4. The degrees of agreement with three correct documents labeled as 'N' were zeros both during each run and with TSUBAKI search engine.

Tables 3 and 4 indicate that most of the correct t_1 documents obtained by TSUBAKI were successfully identified as correct t_1 both in KSU-JA-FV-02 and KSU-JA-FV-03.

Note that the further error analysis turned out that some documents were considered to be missing in the correct t_1 documents provided by the organizers. The examples of the selected t_1 in table 5 were not included in the correct t_1 documents provided, but should be judged that they entail

Table 4: Degree of coincidence between correct t1 documents labeled as 'Y' and those selected as t1 during each run or by TSUBAKI

System	Precision	Recall	F-measure	MAP
FV-01	0.00801	0.00801	0.00006	0.00401
FV-02	0.01395	0.02003	0.00013	0.00757
FV-03	0.01395	0.02003	0.00013	0.00757
TSUBAKI	0.01395	0.02739	0.00014	0.00873

Table 5: Examples of documents considered to be missing as correct t_1 documents

t_2		selected t_1	
id	text	id	text
13	国際連合は，パレスティナを分割する案を採択した。	WBS-59	４７年，国連はパレスティナを，ユダヤ人国家とアラブ人国家に分割する決議案を採択した。
41	20世紀前半にイランでは，レザー＝ハーンが，カージャール朝を廃してパフレヴィー朝を開いた。	WB-69	大戦中にイギリス・ロシア両軍に占領されていたイランでは，１９２１年にレザー＝ハーン（レザー＝シャー）がクーデタで政権を握り，イギリスから独立を回復し，２５年にはトルコ系のカージャール朝を廃してパフレヴィー朝を創始した。
84	パリ条約で，イギリスは北アメリカ植民地の独立を認めた。	WBS-42	イギリスは１７８３年にパリ条約で北米植民地の独立をみとめ，ミシシッピ川以東の広い土地をゆずった。

the corresponding t_2. Thus, we need to remember that the degrees of coincidence in tables 3 and 4 are estimated lower than those with the truly correct documents.

4.2 System Validation

In SV subtask, our systems gave results which were close to the top three results. In the development phase, systems using bagging and random forest showed better results than one with SVM. However, in the formal run, the system using SVM showed best performance among our systems as a result. After submitting the results of the formal runs, however, errors were found in calculating some features used in SV subtask. Thus, we run the experiments after correcting them and generating the training data again. The results of the formal runs and the unofficial, corrected ones for SV subtask are shown in table 6. After correction, KSU-JA-SV-03-C which is based on Bagging showed best performance, followed by KSU-JA-SV-01-C with SVM.

To clarify the degree of contribution of each feature, we carried out the ablation analysis for each run. The results by Macro-F1 are shown in tables 7, 8 and 9. The results by Accuracy are shown in tables 10, 11 and 12.

Tables 7, 8, and 9 show that the macro-F1 of each run was decreased when removing surface features. In Random Forest, however, it was found that the macro-F1 was increased when removing lcs feature. It is presumed that there were many sensitive branches in Random Forest that cannot handle the decision properly because the values of lcs change in a very wide range regardless of labels 'Y' and 'N'.

Table 6: Results of our formal and unofficial runs for SV subtask

Runs	System	Macro F1	Accuracy
formal runs (submitted)	KSU-JA-SV-01	66.96	79.84
	KSU-JA-SV-02	64.87	76.00
	KSU-JA-SV-03	65.72	75.78
unofficial runs (corrected)	KSU-JA-SV-01-C	66.01	79.48
	KSU-JA-SV-02-C	63.80	75.56
	KSU-JA-SV-03-C	67.18	76.50

Table 7: Result of ablation test by Macro-F1 for KSU-JA-SV-01-C (SVM)

Feature	System Description	Macro-F1	Δ
	Baseline	66.01	
Surface features	w/o cos_sim_c	65.57	-0.44
	w/o cos_sim_w	60.34	-5.67
	w/o jc_coef_w	63.18	-2.83
	w/o lcs	64.18	-1.83
Location	w/o location	65.98	-0.03
Named entities	w/o ne_cos_sim	65.91	-0.10
	w/o ne_diff	66.01	0
	w/o ne_n2subset	66.08	0.07
Numerical expressions	w/o numexp_diff	66.08	0.07
	w/o numexp_exact	66.01	0
	w/o numexp_n1subset	66.01	0
	w/o numexp_n2subset	66.01	0

Table 8: Result of ablation test by Macro-F1 for KSU-JA-SV-02-C (Random Forest)

Feature	System Description	Macro-F1	Δ
	Baseline	63.80	
Surface features	w/o cos_sim_c	63.43	-0.37
	w/o cos_sim_w	61.98	-1.82
	w/o jc_coef_w	63.17	-0.63
	w/o lcs	64.60	0.80
Location	w/o location	64.50	0.70
Named entities	w/o ne_cos_sim	64.51	0.71
	w/o ne_diff	64.41	0.61
	w/o ne_n2subset	63.64	-0.16
Numerical expressions	w/o numexp_diff	63.96	0.16
	w/o numexp_exact	62.95	-0.85
	w/o numexp_n1subset	64.70	0.90
	w/o numexp_n2subset	64.13	0.33

Table 9: Result of ablation test by Macro-F1 for KSU-JA-SV-03-C (Bagging)

Feature	System Description	Macro-F1	Δ
	Baseline	67.18	
Surface features	w/o cos_sim_c	64.51	-2.67
	w/o cos_sim_w	63.61	-3.57
	w/o jc_coef_w	62.89	-4.29
	w/o lcs	64.40	-2.78
Location	w/o location	66.91	-0.27
Named entities	w/o ne_cos_sim	66.31	-0.87
	w/o ne_diff	67.18	0
	w/o ne_n2subset	67.18	0
Numerical expressions	w/o numexp_diff	67.18	0
	w/o numexp_exact	67.18	0
	w/o numexp_n1subset	67.31	0.13
	w/o numexp_n2subset	67.18	0

Table 10: Result of ablation test by Accuracy for KSU-JA-SV-01-C (SVM)

Feature	System Description	Accuracy	Δ
	Baseline	79.48	
Surface features	w/o cos_sim_c	78.90	-0.58
	w/o cos_sim_w	77.96	-1.52
	w/o jc_coef_w	78.97	-0.51
	w/o lcs	78.90	-0.58
Location	w/o location	79.55	0.07
Named entities	w/o ne_cos_sim	79.26	-0.22
	w/o ne_diff	79.48	0
	w/o ne_n2subset	79.55	0.07
Numerical expressions	w/o numexp_diff	79.55	0.07
	w/o numexp_exact	79.48	0
	w/o numexp_n1subset	79.48	0
	w/o numexp_n2subset	79.48	0

Table 11: Result of ablation test by Accuracy for KSU-JA-SV-02-C (Random Forest)

Feature	System Description	Accuracy	Δ
	Baseline	75.56	
Surface features	w/o cos_sim_c	72.37	-3.19
	w/o cos_sim_w	76.94	1.38
	w/o jc_coef_w	76.58	1.02
	w/o lcs	74.33	-1.23
Location	w/o location	75.20	-0.36
Named entities	w/o ne_cos_sim	75.49	-0.07
	w/o ne_diff	75.85	0.29
	w/o ne_n2subset	75.27	-0.29
Numerical expressions	w/o numexp_diff	75.34	-0.22
	w/o numexp_exact	74.47	-1.09
	w/o numexp_n1subset	75.71	0.15
	w/o numexp_n2subset	75.34	-0.22

Table 12: Result of ablation test by Accuracy for KSU-JA-SV-03-C (Bagging)

Feature	System Description	Accuracy	Δ
	Baseline	76.50	
Surface features	w/o cos_sim_c	73.24	-3.26
	w/o cos_sim_w	78.03	1.53
	w/o jc_coef_w	76.14	-0.36
	w/o lcs	74.18	-2.32
Location	w/o location	76.29	-0.21
Named entities	w/o ne_cos_sim	76.43	-0.07
	w/o ne_diff	76.50	0
	w/o ne_n2subset	76.50	0
Numerical expressions	w/o numexp_diff	76.50	0
	w/o numexp_exact	76.50	0
	w/o numexp_n1subset	76.65	0.15
	w/o numexp_n2subset	76.50	0

Meanwhile, tables 7 and 9 indicate that only slight differences were observed when removing either numerical expression-based features, location features or named entity features, with the method using SVM or Bagging. Tables 8 show that in Random Forest, it turned out that that some of the macro-F1 and accuracy were decreased as much when removing either location features or named entity features, as when removing surface features. It was also confirmed that some of the numerical expression-based features, location features and named entity features bear an inverse relation, where one feature becomes 'true' when the other one is 'false': for example, a relation between $numexp\_diff$ and $numexp\_n2subset$. Therefore, it was found that removing one of those features didn't help decreasing the macro-F1 or accuracy and rather increased them.

The ablation analysis seemed to show that the contributions to the classification of numerical expression-based features, location features and named entity features are low compared to that of surface features. This is because the rates of docuemnt pairs including missing values in these features were high in the test data: 28% in numerical expression-based features, 40% in location features, and 72% in named entity features. Actually, it was confirmed that numerical expression-based features contribute to the classification strongly in SVM and contribute supplementarily in Random Forest and in Bagging, when combining with other features such as location features and named entity features.

5. CONCLUSION

The systems and results of the team KSU for RITE-VAL task were described in this paper. Three different systems were implemented for each of the two subtasks: Fact Validation and System Validation. In Fact Validation subtask, systems were designed respectively based on character overlap, existence of entailment result 'Y', and voting of entailment results. In System Validation subtask, systems were designed respectively using SVM, Random Forest, and Bagging, with features such as surface features, numerical expressions, location expressions, and named entities. Scores of the formal runs were 52.78% in macro F1 and 66.96% in accuracy with KSU-FV-02 in Fact Validation, and 66.96% in macro F1 and 79.84% in accuracy with KSU-SV-01 in System Validation. Also, in System Validation, scores of the unofficial runs were 67.18% in macro F1 and 76.50% in accuracy with KSU-SV-03-C.

6. REFERENCES

[1] L. Breiman. Bagging predictors. *Mach. Learn.*, 24(2):123–140, Aug. 1996.

[2] L. Breiman. Random forests. *Mach. Learn.*, 45(1):5–32, Oct. 2001.

[3] C. Cortes and V. Vapnik. Support-vector networks. *Mach. Learn.*, 20(3):273–297, Sept. 1995.

[4] S. Hattori and S. Sato. Team skl's strategy and experience in rite2. In *Proceedings of the 10th NTCIR Conference on Evaluation of Information Access Technologies*, pages 435–442, 2013.

[5] T. Makino, S. Okajima, and T. Iwakura. Fll: Local alignments based approach for ntcir-10 rite-2. In *Proceedings of the 10th NTCIR Conference on Evaluation of Information Access Technologies*, pages 518–523, 2013.

[6] S. Matsuyosh, Y. Miyao, T. Shibata, C.-J. Lin, C.-W. Shih, Y. Watanabe, and T. Mitamura. Overview of the ntcir-11 recognizing inference in text and validation (rite-val) task. In *Proceedings of the 11th NTCIR Conference on Evaluation of Information Access Technologies*, 2014.

[7] J. R. Quinlan. Simplifying decision trees. *Int. J. Man-Mach. Stud.*, 27(3):221–234, Sept. 1987.

[8] 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 *Proceedings of the 9th NTCIR Conference on Evaluation of Information Access Technologies*, pages 291–301, 2011.

[9] 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 on Evaluation of Information Access Technologies*, pages 385–404, 2013.