KSU Systems at the NTCIR-14 QA Lab-PoliInfo Task

Tasuku Kimura, Ryo Tagami, Hikaru Katsuyama, Sho Sugimoto, and Hisashi Miyamori

Affiliation 京都産業大学 KYOTO SANGYO UNIVERSITY むすんで、うみだす。

Conference Workshop NTCIR Task QA Lab

ABSTRACT

In **Segmentation Task** which required extracting primary information correctly from the input data, we proposed a method based on rules and vocabulary distributions.
As a result, the team KSU achieved **third in five teams** with **the f-measure of 0.855.**

In **Summarization Task** which demanded generating a summary focused on a specific topic, we tried using a framework of the query-focused abstractive summarization.

In **Classification Task** which called for classifying stances of a certain text for a specific topic, we developed a method combining deep learning and two-stage classifiers.
As a result, the team KSU achieved **second place in 11 teams** with **the accuracy 0.934**.

SEGMENTATION TASK

Pre-processing

Pre-processing for **each sentence in minutes data**.
Before indexing, the following two pre-processings are performed.

1. Single Speech Estimation

Approach: **speech boundary is clarified** according to the pre-defined rules.
Rules: When another speaker starts talking OR when ruled line strings appear
Before: minutes data as a set of sentences.
After: minutes data with clear boundaries between consecutive speech.

2. Speech Type Classification

Approach: classification using fastText[1]
7 speech types: **QUESTION**, **ANSWER**, PROGRESS, GREETING, OPINION, REPORT, and REQUEST
Training data: four local assemblies (110 minutes data)
Accuracy: the estimated accuracy achieved 99.3%.

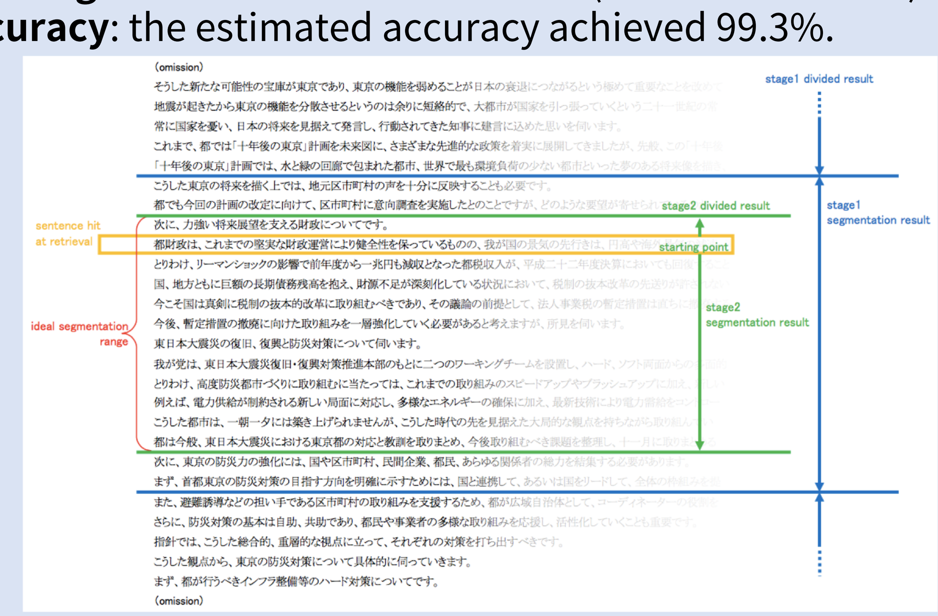


Fig. 1. Overview of two-step segmentation process.

[1] Joulin et al..: Bag of tricks for efficient text classification. In: Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers. pp. 427–431. Association for Computational Linguistics (2017)

Segmentation Process

The **QUESTION** speech and the corresponding answer segment in **ANSWER** speech is extracted from each target speech by the following processing.

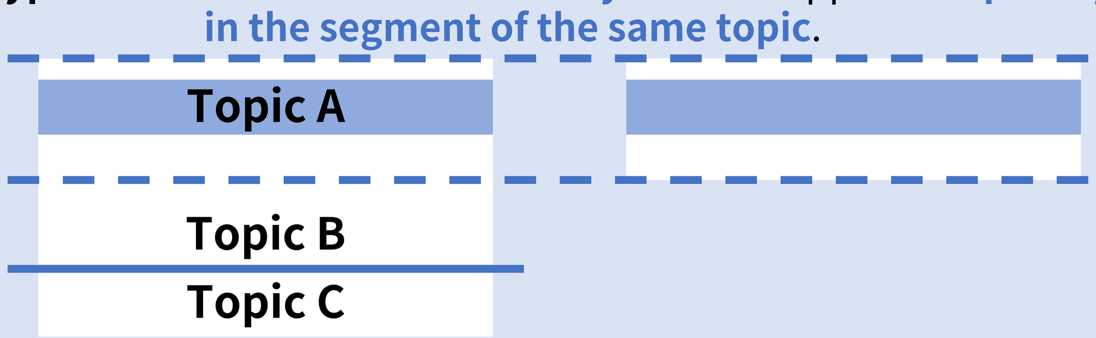
1. Document Retrieval of Single Speech.

Approach: **obtaining the speech section** that contains the sentence most relevant to the query
Query: the strings composing the question theme and those composing the subtopic
Filter: by date, and by **speaker(SF)**

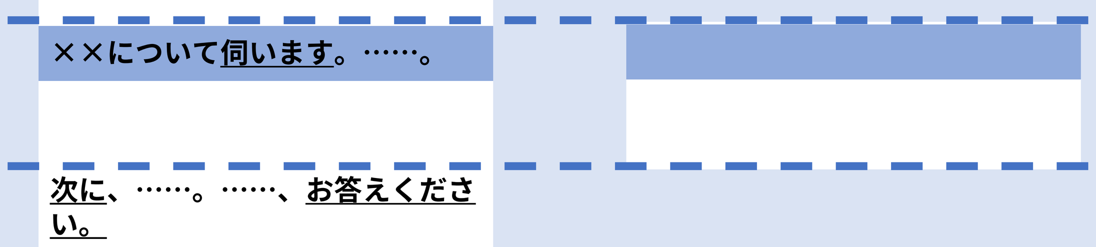
QUESTION: filtering by agreement of speaker's name
ANSWER: filtering by the agreement with the answer immediately after the obtained QUESTION

2. Speech Segmentation

Approach 1: segmentation based on the distribution of word frequency(WF)[2]
Hypothesis: **the similar vocabulary** tends to appear **frequently in the segment of the same topic**.



Approach 2: rule-based segmentation(RB)
Hypothesis: **the boilerplate language frequently appears** at the topic break.



[2] Utiyama et al. : A statistical model for domain-independent text segmentation. In: In Proceedings of the 9th Conference of the European Chapter of the Association for Computational Linguistics. pp. 491–498 (2001)

Results

We tried **eight conditions** corresponding to the different combination of the presence or absence of SF, WF and RB.
Table 1 shows the results.

Table 1. Conditions for experiment

#	Priority	SF	WF	RB
C1		–	–	–
C2		✓	–	–
C3	KSU-01	–	✓	–
C4		–	–	✓
C5	KSU-03	✓	✓	–
C6	KSU-02	–	✓	✓
C7		✓	–	✓
C8	KSU-04	✓	✓	✓

Discussion

The recall of C1 and C2 is expected to be as close to 1 as possible.
➢ **SF improved the recall of "all"**
The appropriate segmentation is expected to improve the precision and F-measure.
➢ Both the two segmentation processings contribute to improve the accuracy. Especially, **RB improves F-measure more greatly** than WF.

Table 2. Result of segmentation

#	all			QUESTION			ANSWER		
	P	R	F	P	R	F	P	R	F
C1	.087	.940	.160	.065	.920	.122	.275	.982	.430
C2	.112	.991	.202	.088	1	.162	.278	.973	.433
C3	.243	.779	.370	.191	.834	.311	.841	.661	.740
C4	.294	.906	.444	.221	.879	.353	.797	.964	.873
C5	.660	.819	.731	.612	.898	.728	.857	.654	.742
C6	.267	.759	.395	.208	.806	.331	.949	.660	.778
C7	.857	.952	.902	.881	.953	.916	.812	.950	.875
C8	.922	.796	.854	.905	.865	.885	.973	.651	.780

SUMMARIZATION TASK

Training data set

Training data were constructed from the Assembly minutes collected from the Web and the Newsletters which contain highlights of the Assembly minutes.
Note that the minutes of Tokyo Metropolitan Assembly in the fiscal years used in the formal run were excluded from the training data.

- Minutes of Tokyo Metropolitan Assembly from 2001 to 2017.
- Minutes of Itabashi City Assembly from 2009 to 2017.

Problems of training data set

1. It is difficult to **deal with unknown words**, since the data set is constructed from the minutes of the specific Assemblies.
2. It is difficult to say that the data set of 19,689 minutes were **sufficient amount for deep learning**.

Solution using Byte Pair Encoding (BPE)

1. BPE Subword tokenizer treats **high frequency words in the training data as one word** and divides **low frequency words into shorter units** such as substrings and characters.
2. **SentencePiece[3]** which can provide unigram-based tokenizers can output **multiple segmentation candidates with confidence degrees for the same input**.
   ➢ **The training data can be augmented** by sampling dynamically from the corpus.

[3] Kudo et al. : Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations. pp. 66–71 (2018)

Proposed model

a. Generating the summary in accordance with the topic
b. Controlling the output length
c. Solving the problem of the Recurrent Neural Network (RNN)

a. Extended attention mechanism

Global attention mechanism is extended so that the attention of document vector is generated based on **the topic vector**.

The vector indicates the content of the summary.

b. LenEmb mechanism

LenEmb[4] is a method to introduce **the length embedding vector** to the input of LSTM in the decoder. The model can generate a summary according to the remaining length information.

c. Diversity cell mechanism

The mechanism[5] **transforms the input vector into vectors orthogonal to each other in each decoding step** by extending the implementation of LSTM.

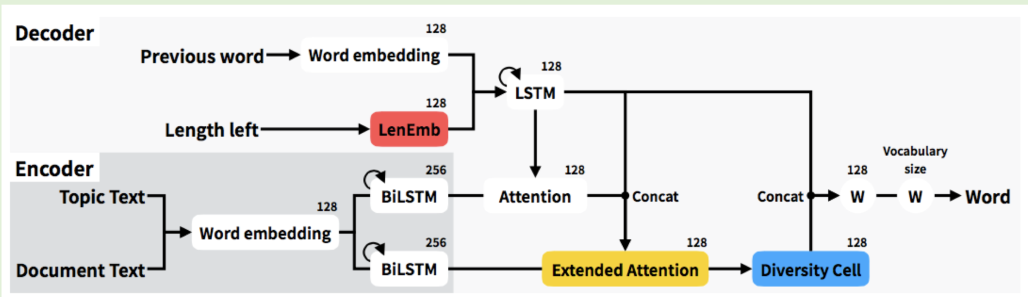


Fig. 2. Model configuration of priority 5.

[4] Kikuchi et al. : Controlling Out- put Length in Neural Encoder-Decoders. In: Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing. pp. 1328–1338 (2016)
[5] Nema et al. : Diversity driven attention model for query-based abstractive summarization. In: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). pp. 1063–1072. Association for Computational Linguistics (2017)

Results

We constructed **six models by combining the three mechanisms**, namely, the tokenizer, the diversity cell, and LenEmb.
Also, the configuration of the model of priority 5 is shown in **Fig.2** as an example of the proposed model.

Table 3. Comparison of the system configuration and the result of summarization

Priority	Tokenizer	Diversity cell	LenEmb	ROUGE-N		ROUGE-L	all-topic			
							content		formed	total
				N=1	N=2		X=0	X=2		
KSU-01	MeCab	✓	–	.158	.028	.009	.043	.043	1.955	.048
KSU-02	MeCab	–	–	.185	.043	.021	.076	.121	1.745	.071
KSU-03	BPE	✓	–	.172	.036	.008	.091	.157	1.715	.104
KSU-04	BPE	–	–	.171	.044	.013	.111	.167	1.419	.093
KSU-05	MeCab	✓	✓	.227	.029	.010	.048	.078	1.692	.048
KSU-06	BPE	✓	✓	.221	.038	.013	.078	.169	1.535	.091

Discussion

BPE
content↑ : The model could deal with unknown words appropriately.
formed↓ : The possibility of outputting a summary with grammatical errors increased.

Diversity cell
content↓ : The predicted word vectors should not necessarily be orthogonal in each decoding step.
formed↑ : The problem of repeated generation of the same words has been alleviated.

LenEmb
content↓／formed↓ :The content of the summary tends to change according to the remaining length, not the topic.

CLASSIFICATION TASK

Classification of Relevance

Input: A text obtained by concatenating a topic and an utterance
Output: A probability value
Config.: One-layered neural network [1]

Classification of Fact-checkability

Input: An utterance, **Output**: A binary probability value
Config.: Two-layered NN composed of LSTM and fully connected layer

Classification of Opinion

a. Construction of the classifier

The following **binary classifiers** are constructed **in two stages** to classify three kinds of labels more accurately.

1. The classifier to identify "no opinion" or "having opinion".
2. The classifier to identify "support" or "against".

b. Selection of the features

The occurrence frequency histogram of word N-grams(N=1,2,3) was made from the utterances in the development data per each label.
The top-K word N-grams (K=200,400,600) **having the largest difference in frequency** were selected as a feature for each label.
Table 4 shows the combinations of features determined from the preliminary experiment.

Table 4. Comparison of the opinion classifier configuration

Model	"no opinion" or "having opinion"		"support" or "against".	
	Features	Dimension	Features	Dimension
St1	1-gram	600	1-gram	600
St2	1-gram	600	1-gram	400
St3	1-gram, 2-gram, 3-gram	200+200+200	1-gram	600
St4	1-gram, 2-gram, 3-gram	200+200+200	1-gram	400

Dataset

Type1 : The correct labels were decided by taking a majority vote of labels attached by annotators.
Type2 : The correct label were decided as "unrelated" if even one annotator attached "unrelated". (used for classification of relevance only)

Result of Relevance

Table 5 shows the classification accuracy. Here, "Model:Re1" represents the model learned from Type 1, and "Model:Re2" learned from Type 2.
It showed that training by Type 2 **improved R0 greatly** whereas it **P0** decreased slightly.

Table 5. Test result of classifying relevance

Model	Acc	P0	P1	R0	R1
Re1	.790	.373	.966	.823	.785
Re2	.873	.567	.893	.257	.969

Result of Fact-checkability

Table 6 shows that the proposed method **tends to judge "not fact-checkable"**, because **R1** was lower than **R0**.
This is considered to be because there is a large proportion of "not Fact-checkable" labels in the training data and that the correct labels were biased.

Table 6. Test result of classifying fact-checkability

Model	Acc	P0	P1	R0	R1
Fc1	.735	.738	.722	.914	.407

Result of Opinion

It can be observed from **Table 7** that **R1** and **R2** are much lower than **R0** in each model.
It is considered to be because the correct labels in the training set were biased to "no opinion".

Table 7. Test result of classifying opinion

Model	Acc	P0	P1	P2	R0	R1	R2
St1	.802	.829	.683	.402	.961	.230	.237
St2	.799	.829	.724	.370	.961	.201	.254
St3	.801	.820	.720	.420	.973	.171	.202
St4	.799	.820	.732	.404	.973	.153	.214

Result of Class

It was confirmed that each proposed model has **high ability to correctly estimate the final stance as Other**, whereas they have **low ability to accurately decide whether it is Fact-checkable Support or Fact-checkable Against**. It is considered that both the recall of Fact-checkable Support and that of Fact-checkable Against in the final classification results were affected, because both the classification accuracy of "fact checkable" and that of "Support" and "Against" were low.

Table 8. Comparison of the system configuration and the result of classifying class

Priority	Rl	FC	St	Acc	P0	P1	P2	R0	R1	R2
1	Re1	Fc1	St1	.932	.937	.579	.056	.995	.075	.008
2	Re1	Fc1	St2	.932	.937	.689	.042	.995	.071	.008
3	Re1	Fc1	St3	.934	.937	.738	.083	.998	.071	.008
4	Re1	Fc1	St4	.934	.937	.738	.083	.998	.071	.008
5	Re2	Fc1	St1	.932	.937	.579	.111	.995	.075	.019
6	Re2	Fc1	St2	.932	.937	.689	.088	.995	.071	.019
7	Re2	Fc1	St3	.934	.937	.738	.100	.997	.071	.011
8	Re2	Fc1	St4	.934	.937	.738	.100	.997	.071	.011