STARS at the NTCIR-14 QA Lab-PoliInfo Classification Task

Daiki Shirafuji[1], Sho Takishita[1], Patrycja Swieczkowska[12], Rafal Rzepka[12], and Kenji Araki[1]

[1] Graduate School of Information Science and Technology,
Hokkaido University, Japan
{d_shirafuji, takishita.sh, swieczkowska, rzepka, araki}@ist.hokudai.ac.jp
[2] RIKEN Center for Advanced Intelligence Project (AIP)

Abstract. The STARS team participated in the Classification task of Question Answering Lab for Political Information (QA Lab-PoliInfo) subtask of the NTCIR-14. This report describes our methods for solving the task and discusses the results. We identify whether the policy and remarks are relevant or not, whether they contain a verifiable fact or not, and predict the stance (positive, negative or neutral) mainly with machine learning approach.

Team Name. STARS

Subtasks. Classification Task (Japanese)

Keywords: assembly minutes · argument mining · argumentative relation · fact checking

1 Introduction

In recent years, the area of argument mining has become popular [1–3]. One of the main tasks of argumentation processing is to identify argument stance to the theme, i.e. recognizing if it is an attack or support [4]. Usually works considering this task focus on political discussions or news articles [5], while our research target are political minutes written in Japanese. One of the problems of measuring arguments is the fact that even if politicians clearly display their stance in a statement, they are less meaningful if no evidence is mentioned. Therefore, in addition to identifying the stance, it is necessary to implement fact checking ability to properly asses a given statement. Because it is important to evaluate the truthfulness of the increasing volume of arguments, in our opinion it is essential to develop and implement fact checking methods.

2 Related Work

In the area of argument mining, fact-checking and fact-checkability discovery have been recently becoming more and more popular [6, 7]. Vlachos and Riedel

Table 1. Relations between Subtasks 1-3 results and class result.

Subtask 1	Subtask 2	Subtask 3	class
relevant	capable to fact-check	positive	support
relevant	capable to fact-check	negative	against
relevant	capable to fact-check	neutral	other
relevant	incapable to fact-check	any result	other
not relevant	any result	any result	other

[8] have shown that constructing manually fact-checked datasets is important for fact-checking. Shu et al. [9] introduced datasets useful for distinguishing fake news from true news automatically, however detecting fake news remains a world-wide problem that is difficult to solve. Various researchers work on this problem, and many fake news recognition challenges were proposed [10]. The area of argument mining focuses also on identifying argument stances (positive, negative or neutral) [11].

There are several studies for identifying argument stance and fact-checking for Japanese language. In the statement map which is made for studying and developing technology to analyze semantic relationships among various Japanese text information existing on the Web, the authors divided arguments into six types: “agreement”, “conflict”, “confinement”, “evidence” and “other” [12, 13].

However, automatic categorization into such types is not able to identify a given argument stance and decide whether a supporting evidence exists or not.

Therefore, for the purpose of the QA Lab-PoliInfo Classification task [14], we have developed methods for identifying argument stances and discovering supporting evidences.

3 Classification task in QA Lab-PoliInfo

The Classification Task in QA Lab-PoliInfo is to determine the class (*support*, *against* or *other*) of a statement in assembly minutes containing discussions on local politics. “*Tsukiji* market should be transferred to *Toyosu*” is an example topic where politicians discuss in order to make decision on the transfer or not. The minutes contain statements divided into three classes: *support*, *against* and *other*. In this task, *support* is defined as opinions that agree to a given topic with reasons stated, *against* include opinions that disagree with stated reasons, and *other* is defined as statements other than *support* and *against*.

To address these differences we divided the task into three dependent tasks (see Table 1): Subtask 1 is to identify whether statements are relevant with a topic or not, Subtask 2 is to identify whether statements are supported by a verifiable fact or not, and Subtask 3 is to identify which stance (positive, negative or other) remarks have. The class of a statement is judge with these tasks, and examples of statements and their classes are shown in Table 2.

Table 2. Example of remark and class.

topic	*Tsukiji-shijou no Toyosu iten ni tsuite* (*Tsukiji* market should be transferred to *Toyosu*)
statement	*Toyosu ha, shin-shijou iten niyori senkyaku-banrai no shisetsu ga dekiru nado, kongo, kankoukyaku no shuukyaku ga ooini kitai dekiru eria de ari masu.* (*Toyosu* must become a good facility that will attract many tourists, such as a facility capable to host thousands of customers could be created after a new market is constructed.)
Subtask1	relevant
Subtask2	capable to fact-check
Subtask3	positive
class	support

In similar tasks to QA Lab-PoliInfo Classification Task, existing researches work on those tasks only with Support Vector Machine [15] because they have small data (e.g. less than 2,000 statements) [16]. However, in the QA Lab-PoliInfo Classification Task, the statements are provided over 10,000, so that we applied Long Short-Term Memory (LSTM) [17] and Bidirectional LSTM (BiLSTM) [18] to the tasks as shown in the next section.

4 Our Methods

We process the Classification task with three steps: Subtask 1, 2 and 3. First, our algorithm judges relevance between the topic and statements (Subtask1), finds a verifiable fact in statements (Subtask 2) and then classifies the stance (Subtask 3) mainly with machine learning. Then, it predicts the class by integrating results of each Subtask according to relations shown in Table 1.

Before applying machine learning, we applied two preprocessing. First, we made a correct answer label for each statement from the given QA Lab-PoliInfo Classification Task data [14]. The data was annotated by three or five (depending on the topics) Japanese native speakers, therefore we regard the mode value as the correct answer. Second, we perform morphological analysis to the topics and statements with JUMAN++ parser [19] and obtain the distributed representation. In the machine learning phase, we train 80% of the data provided by the task organizers and perform tests on remaining 20% of the dataset.

4.1 Subtask 1 Algorithm

In Subtask 1, we utilize three methods to identify the relevance.

Common words between policy and remark (Common Words) We regard statements as relevant to the topic. When the number of common words between a topic and a statement (except for *hiragana*[1] words, which may be stop words) is larger than the threshold: 2 points are given. If we set the threshold at one word, the system makes a false judgement and a statement less relevant to the topic becomes relevant (e.g. tuna + *Tsukiji*). From the experiments, we observed that higher thresholds positively influence precision, otherwise, the recall increases. In this subtask, we focus on the number of correct outputs, so we set the threshold to 2.

Similarity between policy and remark (Similarity) When the mean of similarities between word vectors of the topic and statements is over the threshold, the statement is marked as relevant to the topic. We set the threshold with 0.02, 0.04, 0.06 and 0.08.

LSTM with statements and topics (LSTM) We employ LSTM and BiLSTM with the word vector of a remark and policy as features, and predict whether remarks and the policy is associated together or not.

4.2 Subtask 2 Algorithm

In Subtask 2, we utilize three methods to identify whether statements are supported by a verifiable fact or not.

Whether Numbers are Included in Statements (Number) Statistical information is easy to be fact-checked because those numbers are usually opened to the public and can be checked. Therefore, we regard a statement as fact-checkable when numbers are used. Otherwise, we regard a statement as not fact-checkable.

Semantic Features of Statements using Morphological Analysis (Semantics) Annotating fact-checkable or not is based on whether information which can be confirmed is included or not. And those annotations are following some features, e.g. number phrase and organization name. Therefore, we estimate that following five features can provide useful clues:

- Place Name (e.g. *Tsukiji*, *Toyosu*)
- Person's Name (e.g. *Abe*, *Ueda*)
- Organization Name (e.g. House of Representatives, Cabinet)
- Number Phrase (e.g. 1, one)
- Counter Suffix (e.g. Yen, *Nin (number of people))*

Using these features, we performed six experiments. First, we regard a statement as fact-checkable when any of above five features are included. For investigating which features are useful for the Subtask 2, we perform the same experiment excluding one feature at a time.

[1] Japanese words are written in the italic form.

BiLSTM with statements and topics (BiLSTM) We employ BiLSTM with the word vector of a statement as a feature, and predict whether statements are fact-checkable or not. We experimentally set the number of epochs to 2, which performs the best accuracy.

4.3 Subtask 3 Algorithm

In Subtask 3, we utilize two methods to identify the stance (positive, negative or other).

Japanese Sentiment Polarity Analysis (Sentiment) We assumed that statements stances depend on the sentiment polarity. If the stance is *positive*, favorable words (such as *sansei* (support)) will appear in the statement in order to express the speaker support the theme. Otherwise, opposing words, such as *hantai* (disagree) or *warui* (bad), will be used in statements expressed when a speaker takes the against stance to the theme. Therefore, we calculate statements semantic polarity with Japanese Sentiment Polarity Dictionary [20, 21]. To calculate the statement's polarity, we compute the mean of the polarity of all words in the statement.

For estimating statements' stances from the polarity, we set thresholds between 0.1 to 0.9. When the polarity value is higher than the threshold, we define the stance as *positive*. When the polarity value is lower than the opposite number of the threshold, we define the stance is *negative*. Otherwise, we define the stance as *neutral*.

BiLSTM with statements and topics (BiLSTM) We employ BiLSTM with the word vector of a statement and topic as a feature, and predict which stance (positive, negative or other) statements represent. We experimentally set the number of epochs to 4, which performs the best accuracy.

5 Evaluation

We set a baseline for comparison with all subtasks regarding all statements as relevant (Subtask 1) / capable to fact-check (Subtask 2) / other (Subtask 3).

6 Results

Table 3 shows the accuracy and p-value of each system described in Section 4. As it is presented in Table 3, the Common Words method achieve over 90% accuracy even though it is a simplistic method. Therefore it can be said that the number of common words has a significant impact on the relevance between a topic and a statement.

In the Similarity method, we calculate similarities in all combinations between topics and statements, and it seems that low similarities cause a decrease

Table 3. Accuracy of each method.

	Methods		Accuracy
Subtask 1	Baseline		91.23%
	Common Words		91.12%*
	Similarity	threshold=0.02	91.27%
		threshold=0.04	91.09%**
		threshold=0.06	90.80%
		threshold=0.08	89.53%
	LSTM	LSTM	91.69%****
		BiLSTM	**92.57%****
Subtask 2	Baseline		71.55%
	Number		61.38%****
	Semantics	All features	52.77%****
		Except Place Name	64.73%****
		Except Person's Name	54.39%****
		Except Organization Name	53.34%****
		Except Number Phrase	53.91%****
		Except Counter Suffix	52.93%****
	BiLSTM		**90.47%****
Subtask 3	Baseline		85.33%
	Sentiment	threshold=0.1	42.92%*
		threshold=0.2	43.32%**
		threshold=0.3	43.34%*
		threshold=0.4	46.77%
		threshold=0.5	48.19%
		threshold=0.6	48.70%
		threshold=0.7	49.01%
		threshold=0.8	49.18%
		threshold=0.9	49.19%
	BiLSTM		**87.83%****

* : p-value < 0.05
** : p-value < 0.01
*** : p-value < 0.001
**** : p-value < 0.0001

in overall similarity even when they are relevant. This problem can be solved by setting a lower threshold, however, the system then classifies almost all statements as relevant. This method does not vary much from the baseline. For improving the results, it could be better to use Word Mover's Distance (WMD) [23]. With WMD, we could focus on better detection of important words, which in turn, could lead to improved results.

It can be considered that words in topics should also be considered. Therefore, in LSTM method, statements and topics are taken into account, thus common words in the topic and the statement are both processed by the system.

BiLSTM also obtained better results than baselines in Subtasks 2 and 3. In Subtask 2, the difference between BiLSTM method and the baseline accuracy is over 20 points. This means the system can reliability predict whether statements contain a verifiable fact or not in the most cases. For improving the results even further, it could be necessary to employ numerals (e.g. population or financial data) as features. In Subtask 3, the difference between BiLSTM method and the baseline is not substantial (approximately 2 points). Our method classified almost all of statements as "other". To solve this problem, we should employ automatic discovery of agreement and disagreement phrases (e.g. "*sansei* (I agree with)" and "*hantai* (I disagree with)") and utilize its results as another feature.

However, other proposed methods did not achieve satisfying results not exceeding even the baselines.

In Subtask 2, Number method obtained results similar to the baseline, but some statements which are fact-checkable could not be recognized as such. For example, the system recognized semantic features correctly, but annotators tagged well known facts, such as "*Tokyo Olympic to Paralympic ha 2020-nen ni kaisai sareru* (Tokyo Olympic and Paralympic being held in 2020)", as not fact-checkable or rather not worthy checking. Those differences decreased the results.

Other than the Except Place Name method, Semantics methods got the accuracy around 50%. This is because place names appear in statements which cannot be fact-checked. For example, almost all of statements in the theme "Tsukiji market should be transferred to Toyosu" mentioned Toyosu or Tsukiji but it does not necessarily mean these statements are fact-checkable. Therefore, we conclude that place name should not be used as a feature in the fact-checking task with political texts.

In Subtask 3, Sentiment methods were not able to exceed 50% accuracy. The main reason is that 85% of statements classified as "other" and those statements also include positive or negative words. Other reason is that some statements which stance is positive or negative have the opposite stance words. For example, the statement "*... he hantaisuru iken ni taishite sansei desu* (I agree in the opinion against ...)" take the negative stance, but it includes a positive word, i.e. "*sansei* (agree)". In such cases, Sentiment method cannot predict stance correctly, but BiLSTM method predict stance in almost all of the cases.

Regarding p-value shown on the Table 3, the methods with LSTM and BiLSTM produced the significantly different in comparison with the each baseline because the p-values of those methods are smaller than 0.0001.

In Subtask 1 and 3, it can be clearly said that LSTM methods have the significantly difference comparing to other methods. The p-values of other methods produce 0.01 at most, however LSTM and BiLSTM methods have the p-values which are less than 0.0001.

In Subtask 2, BiLSTM has the significantly different in comparison with the each baseline, but other methods also have the difference. However, because those methods produces lower accuracy than the baseline, this high p-value is due to those bad accuracy.

7 Conclusions and Future Work

We have proposed a set of methods for NTCIR-14 QA Lab-PoliInfo Classification Task and BiLSTM using word vectors of topics and statements achieved the best accuracy in all subtasks. Methods used in Subtask 1 tend to classify data as "relevant", thus it can be observed that machine learning persistently yields good results, showing that it is important to design better features. However, in Subtasks 1 and 3, almost all of the results were the same ("relevant" in Subtask 1 or "other" in Subtask 3).

BiLSTM method achieved satisfying results in Subtask 2 as well. However, in this method we did not use digits and counter suffixes which could indicate fact-checkability of the analyzed statement. As a next step, we plan to employ those features into BiLSTM.

BiLSTM performed well in Subtask 3, however, in this method we did not consider phrases like "I agree with" and "I disagree with" which could easily indicate a stance of the analyzed statement. As a next step, we plan to employ those phrases as a feature into BiLSTM.

In future, we will also use trained distributed word representations in order to consider synonyms, and perform machine learning with word embeddings. In addition, because the ratio of labels is biased (i.e. there is many more examples annotated as "relevant" than "not relevant"), we will need to reduce the uneven distribution.

References

1. Nancy Green, Kevin Ashley, Diane Litman, Chris Reed, and Vern Walker: Proceedings of the First Workshop on Argumentation Mining. In: Proceedings of the First Workshop on Argumentation Mining, 2014.
2. Elena Cabrio and Serena Villata.: Five Years of Argument Mining: a Data-driven Analysis. In: Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence (IJCAI-18), pp. 5427–5433. 2016.
3. Stab, Christian and Miller, Tristan and Schiller, Benjamin and Rai, Pranav and Gurevych, Iryna.: Cross-topic argument mining from heterogeneous sources. In: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pp. 3664–3674, 2018.

4. Lucas Carstens and Francesca Toni.: Identifying attack and support argumentative relations using deep learning. ACM Transactions on Internet Technology (TOIT) 17.3 (2017): 30.
5. Oana Cocarascu and Francesca Toni.: Using Argumentation to improve classification in Natural Language problems. In: Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, pages 1374-1379 Copenhagen, Denmark, September 7-11, 2017.
6. Dusmanu, Mihai, Elena Cabrio, and Serena Villata. "Argument mining on Twitter: Arguments, facts and sources." Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing. 2017.
7. Zubiaga, Arkaitz. "Mining social media for newsgathering." arXiv preprint arXiv:1804.03540 (2018).
8. Andreas Vlachos and Sebastian Riedel.: Fact Checking: Task definition and dataset construction. In: Proceedings of the ACL 2014 Workshop on Language Technologies and Computational Social Science, pages 18-22, Baltimore, Maryland, USA, June 26, 2014.
9. Kai Shu, Amy Sliva, Suhang Wang, Jiliang Tang, and Huan Liu.: Fake News Detection on Social Media: A Data Mining Perspective. ACM SIGKDD Explorations Newsletter 19.1 (2017): 22-36.
10. Baird, Sean and Sibley, Doug and Pan, Yuxi.: Talos targets disinformation with fake news challenge victory. Fake News Challenge. 2017.
11. Brian Xu, Mitra Mohtarami; James Glass.: Adversarial Domain Adaptation for Stance Detection. arXiv preprint arXiv:1902.02401, 2019.
12. Koji Murakami, Eric Nichols, Suguru Matsuyoshi, Asuka Sumida, Shouko Masuda, Kentaro Inui and Yuji Matsumoto.: Statement map: assisting information credibility analysis by visualizing arguments. In: Proceedings of the 3rd workshop on Information credibility on the web. ACM, 2009. p. 43-50.
13. Koji Murakami, Eric Nichols, Junta Mizuno, Yotaro Watanabe, Shouko Masuda, Hayato Goto, Megumi Ohki, Chitose Sao, Suguru Matsuyoshi, Kentaro Inui and Yuji Matsumoto.: Statement Map: Reducing Web Information Credibility Noise through Opinion Classification. In: Proceedings of the fourth workshop on Analytics for noisy unstructured text data. ACM, 2010. p. 59-66.
14. Yasutomo Kimura, Hideyuki Shibuki, Hokuto Ototake, Yuzu Uchida, Keiichi Takamaru, Kotaro Sakamoto, Madoka Ishioroshi, Teruko Mitamura, Noriko Kando, Tatsunori Mori, Harumichi Yuasa, Satoshi Sekine and Kentaro Inui.: Overview of the NTCIR-14 QA Lab-PoliInfo Task. In: Proceedings of the 14th NTCIR Conference, 2019.
15. V. Vapnik and A. Lerner, ?Pattern recognition using generalized portrait method,? Automation and Remote Control, 24, pp. 774-780, 1963.
16. Divyank Barnwal, Siddharth Ghelani, Rohit Krishna, Moumita Basu and Saptarshi Ghosh. Identifying fact-checkable microblogs during disasters: a classification-ranking approach. In: Proceedings of the 20th International Conference on Distributed Computing and Networking. ACM, 2019. p. 389-392.
17. Hochreiter, Sepp, and Jürgen Schmidhuber. (1997). Long short-term memory. Neural computation, 9(8), 1735-1780.
18. Mike Schuster, and Kuldip K Paliwal. "Bidirectional recurrent neural networks." IEEE Transactions on Signal Processing 45.11 (1997): 2673-2681.
19. Hajime Morita, Daisuke Kawahara and Sadao Kurohashi: Morphological Analysis for Unsegmented Languages using Recurrent Neural Network Language Model, Proceedings of EMNLP 2015: Conference on Empirical Methods in Natural Language Processing, pp.2292-2297, (2015.9.17).

20. Nozomi Kobayashi, Kentaro Inui, Yuji Matsumoto, Kenji Tateishi. Collecting Evaluative Expressions for Opinion Extraction, Journal of Natural Language Processing 12(3), 203-222, 2005.
21. Masahiko Higashiyama, Kentaro Inui, Yuji Matsumoto. Learning Sentiment of Nouns from Selectional Preferences of Verbs and Adjectives, Proceedings of the 14th Annual Meeting of the Association for Natural Language Processing, pp.584-587, 2008.
22. Sadao Kurohashi and Daisuke Kawahara, 2009. Japanese Morphological Analysis System JUMAN 6.0 Users Manual. http://nlp.ist.i. kyoto-u.ac.jp/EN/index.php?JUMAN.
23. Matt J. Kusner, Yu Sun, Nicholas I. Kolkin and Kilian Q. Weinberger. (2015, June). From word embeddings to document distances. In International Conference on Machine Learning (pp. 957-966).