Multiple News Headlines Generation using Page Metadata

Kango Iwama, Yoshinobu Kano
Faculty of Informatics, Shizuoka University, Japan
kiwama@kanolab.net, kano@inf.shizuoka.ac.jp

Abstract

Multiple headlines of a newspaper article have an important role to express the content of the article accurately and concisely. A headline depends on the content and intent of their article. While a single headline expresses the whole corresponding article, each of multiple headlines expresses different information individually. We suggest an automatic generation method of such diverse multiple headlines in a newspaper. Our generation method is based on the Pointer-Generator Network, using page metadata on a newspaper which can change headline generation behavior. We conducted automatic evaluations for generated headlines. The results show that our method improved ROUGE-1 score by 4.32 points compared to a baseline system. This is the first trial to evaluate such multiple headlines generation as far as we know. These results suggest that our model using page metadata can generate various multiple headlines for an article with better performance.

1 Introduction

Headlines of newspaper articles have a role to express the content accurately and concisely. Newspapers have *pages*, by which the importance of an article, and sometimes an article's genre, is determined. Therefore, a headline depends on the *page* metadata. For example, the first (front) *page* is normally most important; the literary style of headlines is different depending on genres such as national current affairs and local news. The contents and corresponding headlines of articles are different by the *page*.

Generation of newspaper article headlines is a kind of summarization tasks of articles. There have been a variety of previous works of headline generation and document summarization: neural headline generation by AMR (Takase et al., 2016); Japanese news articles compression using the Dependency Tree (Hasegawa et al., 2017); summary generation by Attention-based model (Rush et al., 2015); readable summary generation by GAN (Wang and Lee, 2018). These methods normally generate a single summary from a single given document. However, a news article could have multiple headlines. Multiple headlines could have a sub headline(s) in addition to its main headline. Headlines do not share same information; main and sub headlines supplement the content of an article each other (Figure 1). Therefore, multiple headline generation requires a variety of headlines with different contents from the same article.

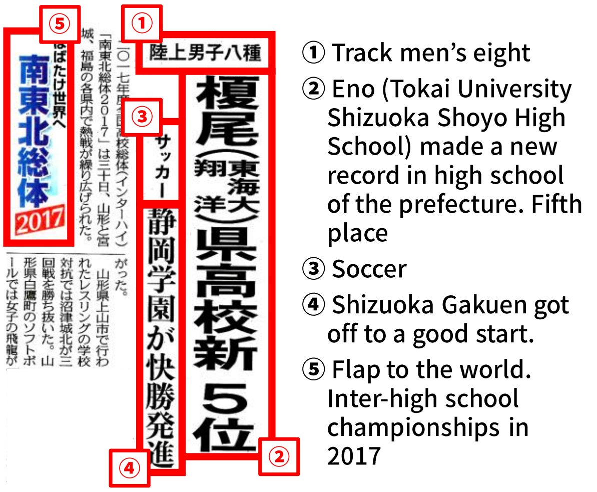
①
陸上男子八種
②
榎尾（東海大翔洋）県高校新５位
③
サッカー
④
静岡学園が快勝発進
⑤
はばたけ世界へ
南東北総体2017
① Track men's eight
② Eno (Tokai University Shizuoka Shoyo High School) made a new record in high school of the prefecture. Fifth place
③ Soccer
④ Shizuoka Gakuen got off to a good start.
⑤ Flap to the world. Inter-high school championships in 2017

Figure 1: Example of multiple headlines in newspaper. (The Chunichi Shimbun, 2017)

Wang et al. (2016) generated multiple headlines, then scored them to filter candidates out. They aimed to provide candidates of main headlines rather than to provide main and sub headlines. They used three generation models, where a single headline is generated from each model.

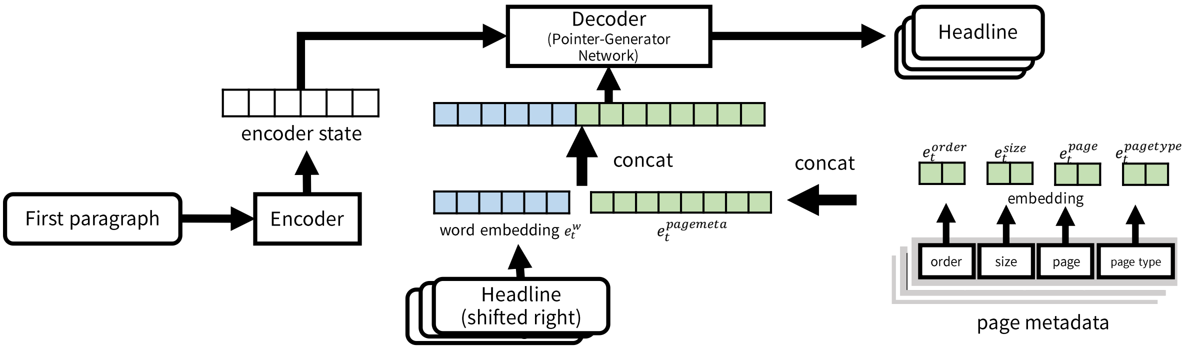

Figure 2: Our method overview.

We suggest a method to generate such a variety of multiple headlines using page metadata. In contrast to the previous work (Wang et al., 2016), we use a single model to generate multiple headlines. Our system generates individual main/sub headlines separately, which allows partial re-generation upon users' requests e.g. when users wish to change a generated headline and/or its style. Our generation method is based on the Pointer-Generator Network, using page metadata on a newspaper which can change headline generation behavior. This is the first trial to evaluate such multiple headlines generation as far as we know. Our evaluation results show better ROUGE-1 score in 4.32 points than a baseline.

2 Model

A news article often picks up new topics which include new named entities such as person names, requiring unknown word processing. Pointer-Generator Network (See et al., 2017) is a hybrid model of Attention-based Seq2Seq (Nallapati et al., 2016) and Pointer Networks (Vinyals et al., 2015) for automatic summarization. Pointer-Generator Network temporarily gives a word ID to an unknown word, outputs a probabilistic distribution of its given lexicon including the unknown words. Our method is based on Pointer-Generator Network, using page metadata as one of its inputs. This page metadata includes a headline *order* within multiple headlines for an article, a headline *size* indicating the font size, an article's page number, and an article's page type. We explain details of these metadata in the Dataset section later. We describe our model assuming that the input is in the Japanese language, but the system architecture can be applied to any other languages.

Our method overview is shown in Figure 2. Given a target article's body text, we use words of the first paragraph with the article's page metadata as inputs to our model. Our encoder takes morphemes (tokens) w_i of the first paragraph, then we obtain the hidden state of our encoder. We define the input x_t to our decoder LSTM at time step t as follows:

$$x_t = e_t^w \oplus e^{pagemeta}$$
$$e^{pagemeta} = e^{order} \oplus e^{size} \oplus e^{page} \oplus e^{pagetype}$$

where $\oplus$ indicates concatenation. CLSTM (Contextual LSTM) (Ghosh et al., 2016) generates sentences which are related to input topics, using concatenation of a word vector and a topic vector as an input to LSTM. We input an input vector x_t to the decoder LSTM, which is a concatenation of a word vector e_t^w and a vector of page metadata $e^{pagemeta}$. We define e^{order}, e^{size}, e^{page}, and $e^{pagetype}$ as vectors in which the order of the headline, the headline's *size*, the page number, and the page type is embedded, respectively.

Final word distribution P(w) is calculated as follows:

$$\mathrm{P(w)} = P_{vocab}(\mathrm{w}) + (1 - P_{gen}) \sum\nolimits_{i:w_i=w} a_i^t$$

P_{vocab} is calculated by feeding a vector that concatenates a decoder state s_t and a context vector h_t^*, through linear layers. P_{gen} is calculated by feeding a vector that sums h_t^*, s_t, x_t, through sigmoid function. a_i^t indicates attention distribution. During training, the loss at the time step t is a negative log likelihood of a target word, and an entire loss is an average of these losses.

3 Dataset

We use a newspaper corpus provided by Chunichi Shimbun, which is one of the major Japanese newspaper companies. This newspaper article

Model type	ROUGE			distinct-n	
	1	2	L	1	2
(1) word (baseline)	19.52	8.18	17.70	0.0135	0.1740
(2) + order	22.45	9.57	20.24	0.0145	0.1669
(3) + order + size	23.77	10.13	21.55	0.0141	0.1660
(4) + order + size + page	23.74	**10.19**	21.51	0.0143	0.1672
(5) + order + size + page + page type	**23.84**	10.17	**21.60**	0.0142	0.1651

Table 1: ROUGE F1 and distinct-n(n=1,2) scores on the test set.

Page type	Data size	ROUGE			distinct-n	
		1	2	L	1	2
Variable page	2473k	24.09	8.87	22.01	0.0234	0.2133
Local page	3425k	**26.68**	**12.49**	**24.05**	0.0173	0.1755
Special page	537k	22.63	9.01	20.40	0.0638	0.3292

Table 2: ROUGE F1 and distinct-n(n=1,2) scores on the test set by page type (model (5)).

Page Type	distinct-n	
	1	2
Variable page	0.0333	0.3511
Local page	0.0320	0.3164
Special page	0.0465	0.3562

Table 3: distinct-n(n=1,2) scores on true data.

corpus includes almost all of the published articles for 30 years. An article in this corpus includes a page number, a page type, a body text of the article, and corresponding multiple headlines. A page type is one of *variable page*, *local page, special page, radio page, or additional page*. Each headline has *size* information of that headline. Size information indicates the physical size in five grades. The smaller the value, the larger the physical size. Size information normally shows the importance of the article and headlines.

We used about 15 years from the original corpus because older data does not include the size information. In order to exclude headlines which are fixed regardless of the article content, e.g. the "editorial" and "column", we excluded the top 200 frequent headlines from the data. Then we excluded articles of which the number of words in the first paragraph is more than 10 and less than 150, also excluded articles which headlines include more than 10 words. We finally excluded articles with headlines which *order*, i.e. the number of appearances among multiple headlines for an article, is fifth or after. The final dataset used in our experiments include 6,435,774 articles.

4 Experiments

We compared five models which use different page metadata. Our evaluation was performed by 3-fold cross validation, using ROUGE and distinct-n (Li et al., 2016), which is a metric to evaluate diversity of outputs. We used ROUGE-1, ROUGE-2, ROUGE-L and distinct-n (n=1, 2) scores.

We used MeCab 0.996[1] with the mecab-ipadic-NEologd[2] dictionary to tokenize sentences. We implemented our model using PyTorch 1.0.1[3]. Dimensions of each embedding vector were set to 128. Each layer of our encoder LSTM and our decoder LSTM has 256 dimensions. The vocabulary size was 50,000. Word vector representations are same between the encoder and the decoder. Adam as an optimizer, batch size was 64. All numerals were regarded as unknown words. Japanese letters of numerals in the first paragraph were converted into Arabic numerals. Consecutive numerals were concatenated into a single word.

We used an early stopping to avoid overfitting in our training. Our validation data was 2.5% of the training data, randomly extracted. The remaining of the data was used as our training data. The loss for the validation data was calculated for every 1000 iterations, and training was stopped if the loss did not decrease within 1 epoch at the longest.

5 Results

Table 1 shows our results of the automatic evaluation. *word*, *order, size, page*, and *page type*

[1] https://taku910.github.io/mecab/

[2] https://github.com/neologd/mecab-ipadic-neologd

[3] https://pytorch.org/

F: 年金記録不備問題で社会保険庁は十一日、二十四時間態勢でオペレーターが対応する電話相談「ねんきんあんしんダイヤル」を始めたが、午前八時半から電話が殺到。午後になってもほとんどつながらない状態で、同庁担当者は「ご不便をおかけして申し訳ない。今後はスタッフを増員したい」と、前日のシステム障害に続く不手際に平謝りだった。(The Social Insurance Agency started working on a telephone call "Pension Relief Dial", which the operator responded in 24 hours on Monday, due to the problem of inadequate pension record, but the telephone was flooded at 8:30 am. With no connection in the afternoon, the agency official said, "I am sorry for the inconvenience. We would like to increase the number of staff in the future," they said sorry for the trouble following the system failure the day before.)

H: 年金記録　フリーダイヤル相談 (Pension record, Free dial consultation)
『不安』鳴りっぱなし (“anxiety” ringing)

B: 社保庁 (Social Insurance Agency)
『ねんきんあんしんダイヤル』 (“Pension Relief Dial”)

P: 社保庁 (Social Insurance Agency)
社保庁 『ねんきん』(Social Insurance Agency, Pension)

PA: 社保庁 年金電話相談 (Social Insurance Agency, Pension telephone consultation)
電話殺到 (a flood of calls)
年金電話相談 電話殺到(Pension telephone consultation, a flood of calls)
『今後はスタッフ増員』(“We will increase the number of staff in the future.”)
年金記録不備 (Pension record deficiencies)
どうなる年金(Whither Pension?)

Figure 3: Example of automatically generated headlines by our method. **F:** first paragraph, **H:** human written original headlines, **B:** baseline (word), **P:** word + all page metadata, **PA:** all parameters combinations of all page metadata (Excerpts)

indicate the word vector, e^{order}, e^{size}, e^{page}, and $e^{pagetype}$, respectively. Model (1) outputs words that concatenate multiple headlines, but the outputs are subdivided into individual headlines when in the evaluation. Model (5) used all of page metadata. Comparing with the baseline, Model (5) performed better 4.32 points in ROUGE-1, 1.99 points in ROUGE-2, 3.9 points in ROUGE-L. ROUGE-1 and ROUGE-L scores were the highest in model (5). Regarding the distinct-1 score, which is the metric to evaluate diversity of outputs, any model using page metadata performed better than model (1) that uses words only.

6 Discussion

The result of the automatic evaluation shows that page metadata improves ROUGE scores. Table 2 shows the results for each page type. We discarded page types of *radio page* and *additional page* because articles of these types were 0.01% of the entire data. Among these page types, *local page* was the best (ROUGE-1 and ROUGE-2 are $p<0.001$).

Table 3 shows the distinct-n scores that are calculated from the original headlines of the newspaper articles for each page type. We used 500,000 headlines extracted randomly to calculate scores in Table 3. The distinct-n scores of *local page*s are lower than the other page types. The ROUGE scores of *local page*s would have been higher than other page types because the vocabulary patterns in *local page* type were limited.

Figure 3 shows a part of a generated headline example of using all of page metadata. Our model generated more variety of headlines than original human written headlines and the baseline headlines. On the other hand, while the same information does not appear repeatedly in multiple headlines of an article in the original newspaper, our model sometimes generated headlines with the same information. The distinct-n scores in Table 1 do not show an increase in the diversity of the vocabulary. We currently assume that humans will select final candidates from our system output, but automatic selection excluding information overlaps would be our future work.

Using other evaluation metrics could be another future work. Even if a generated headline is different from with a corresponding human written headline, some of such headlines are acceptable because the human written headlines are not the unique available gold standard. Therefore, manual evaluations to measure quality of the headlines are also meaningful. Because larger vocabulary is required to generate diverse headlines, we would like to handle omitted words and unknown words which even do not appear in articles in future.

7 Conclusion

We suggested an automatic generation method of multiple headlines of news articles. We can control generated headlines by configuring page metadata manually if needed. While we used newspaper corpus, our method can be applied to any other media e.g. journals and electronic articles that could have multiple headlines. This is the first trial to generate such multiple articles as far as we know. Our model using page metadata performed better than the baseline 4.32 points in ROUGE-1 score.

References

The Chunichi Shimbun, The Chunichi Shimbun (Shizuoka) on July 31, 2017. page 12.

Shalini Ghosh, Oriol Vinyals, Brian Strope, Scott Roy, Tom Dean, Larry Heck, 2016. Contextual LSTM (CLSTM) models for Large scale NLP tasks. arXiv preprint arXiv:1602.06291.

Shun Hasegawa, Yuta Kikuchi, Hiroya Takamura, Manabu Okumura, 2017. Japanese Sentence Compression with a Large Training Dataset. *In Proceedings ofthe 55th Annual Meeting ofthe Association for Computational Linguistics*, pages 281-286.

Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, Bill Dolan, 2016. A Diversity-Promoting Objective Function for Neural Conversation Models. *In Proceedings of NAACL-HLT 2016*, pages 110-119.

Ramesh Nallapati, Bowen Zhou, Cicero Nogueira dos Santos, Caglar Gulcehre, Bing Xiang, 2016. Abstractive Text Summarization Using Sequence-to-Sequence RNNs and Beyond. *In Proceedings ofthe 20th SIGNLL Conference on Computational Natural Language Learning (CoNLL)*, pages 280-290.

Alexander M. Rush, Sumit Chopra, Jason Weston, 2015. A Neural Attention Model for Abstractive Sentence Summarization. *In Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 379-389.

Abigail See, Peter J. Liu, Christopher D. Manning, 2017. Get To The Point: Summarization with Pointer-Generator Networks. *In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, pages 1073-1083.

Sho Takase, Jun Suzuki, Naoaki Okazaki, Tsutomu Hirao, Masaaki Nagata, 2016. Neural Headline Generation on Abstract Meaning Representation. *In Proceedings ofthe 2016 Conference on Empirical Methods in Natural Language Processing*. pages 1054-1059.

Oriol Vinyals, Meire Fortunato, Navdeep Jaitly, 2015. Pointer Networks. *Neural Information Processing Systems*.

Shuguang Wang, Eui-Hong (Sam) Han, Alexander M. Rush, 2016. Headliner: An integrated headline suggestion system. *Computation + Journalism Symposium*.

Yau-Shian Wang, Hung-Yi Lee, 2018. Learning to Encode Text as Human-Readable Summaries using Generative Adversarial Networks. *In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4187-4195.