A Case Study on Neural Headline Generation for Editing Support

Kazuma Murao[*†], **Ken Kobayashi**[*†], **Hayato Kobayashi**[†‡],
Taichi Yatsuka[†], **Takeshi Masuyama**[†], **Tatsuru Higurashi**[†], **Yoshimune Tabuchi**[†]
[†]Yahoo Japan Corporation [‡]RIKEN AIP
{kmurao,kenkoba,hakobaya}@yahoo-corp.jp
{tyatsuka,tamasuya,thiguras,yotabuch}@yahoo-corp.jp

Abstract

There have been many studies on neural headline generation models trained with a lot of (article, headline) pairs. However, there are few situations for putting such models into practical use in the real world since news articles typically already have corresponding headlines. In this paper, we describe a practical use case of neural headline generation in a news aggregator, where dozens of professional editors constantly select important news articles and manually create their headlines, which are much shorter than the original headlines. Specifically, we show how to deploy our model to an editing support tool and report the results of comparing the behavior of the editors before and after the release.

(a) List of news topics including short titles.

(b) Page of news entry including headline and lead.

Figure 1: News-aggregator of Yahoo! JAPAN.

1 Introduction

A news-aggregator is a website or mobile application that aggregates a large amount of web content, e.g., online newspapers provided by different publishers. The main purpose of such a service is to help users obtain important news out of vast amounts of information quickly and easily. Therefore, it is critical to consider how to compactly show news, as well as what type of news to select, to improve service quality. In fact, the news-aggregator of Yahoo! JAPAN[1], the largest Japanese portal site, is supported by dozens of professional editors who constantly select important news articles and manually create their new headlines called *short titles*, which are much shorter than the original headline, to construct a news-topic list. Note that we use the term "title" to avoid confusion with the original news headline, although they are similar concepts.

Figure 1 shows screenshots of the news-aggregator of Yahoo! JAPAN, where the English translations of the short title, headline and lead are listed in Table 1. The left figure (a) shows the list of news topics (important news articles), which includes short titles, and the right figure (b) shows the entry page of the first topic in the list, which consists of a headline and lead. The lead is a short version of the article and can be used by users to decide whether to read the whole article. The editors' job is to create a short title from news content including the headline and lead. A short title has two advantages over a normal headline; one is quick understandability of the content and the other is saving display space by using a single line. This means that short titles can increase a user's chances of reaching interesting articles. Since the click-through rate of news articles is directly related to ad revenue, even a small improvement in short titles has a significant impact on business.

We tackle an automatic-generation task of such short titles for a news aggregator to support the

[*]Both authors contributed equally to this work.
[1]https://www.yahoo.co.jp/

Proceedings of NAACL-HLT 2019, pages 73–82
Minneapolis, Minnesota, June 2 - June 7, 2019.

	Japanese	English translation
Short title	首相 忖度ないと言い切れず	The prime minister cannot say that there is no surmise.
Headline	忖度なかったと言い切ることはできない＝加計問題で安倍首相	It cannot be said that there is no "sontaku (surmise)" with absolute certainty. The prime minister Abe said about the problem of "Kake Gakuen (Kake school)".
Lead	安倍晋三首相は１４日午後行われた参院予算委員会の集中審議で、加計疑惑などを巡り、官僚側から首相に対する忖度（そんたく）があったのではとの指摘に対して「忖度があったかどうか、忖度される側には分かりにくい面もある」と述べた。「忖度がなかったと言い切ることはできない」としつつ、「ごまをするための忖度は求めていない」などと説明した。塚田一郎委員（自民）への答弁。	Prime Minister Shinzo Abe said, in an intensive deliberation with the House of Councilors Budget Committee held on the afternoon of the 14th, as an answer to a question about whether bureaucrats surmised to the prime minister regarding the Kake suspicion, "It is difficult to understand whether there is a sontaku (surmise)". He said "It cannot be said that there was nothing wrong," while explaining that "I do not need to be obsequious". An answer to Ichiro Tsukada (LDP).

Table 1: Short title, headline, and lead in Figure 1(b) with English versions.

editorial process. Our task is a variant of news-headline generation, which has been extensively studied, as described in Section 6. A clear difference between their task and ours is that we need to generate short titles from news content including headlines. Thus, we formulate our task as an abstractive summarization from multiple information sources, i.e., headlines and leads, based on an encoder-decoder model (Section 2).

There are roughly three approaches for handling multiple information sources. The first approach is to merge all sources with some weights based on the importance of each source, which can be achieved by a weighted average of the context vectors, as in multimodal summarization (Hori et al., 2017). This is the most general approach since the other two can also be regarded as special cases of the weighted average. The second approach is to use one source as the main source and others as secondary ones. This is effective when the main source can be clearly determined, such as query-focused summarization (Nema et al., 2017), where the target document is main and a query is secondary. The third approach is to find the salient components of the sources. This is suitable when there are many sources including less informative ones (redundant sources), such as lengthy-document summarization that outputs a multi-sentence summary (Tan et al., 2017), where each sentence can be regarded as one source. We addressed an extension of the weighted average approach and compared our proposed model with a multimodal model (Hori et al., 2017) from the first approach and a query-based model (Nema et al., 2017) from the second approach, as well as the normal encoder-decoder model. Since we have only two sources (headlines and leads), where the headline source is clearly salient for generating a short title, the third approach can be reduced to the normal encoder-decoder model.

Our contributions are as follows.

- We report on a case study of short-title generation of news articles for a news aggregator as a real-world application of neural headline generation. This study supports previous studies based on the encoder-decoder model from a practical standpoint since most real-world news articles basically already have headlines, which means that there has been little direct application of these previous studies.

- We propose an encoder-decoder model with multiple encoders for separately encoding news headlines and leads (Section 3). Our comparative experiments with several baselines involving evaluations done by crowdsourcing workers showed the effectiveness of our model, especially using the "usefulness" measure (Section 4).

- We describe how to deploy our model to an editing support tool and show the results of comparing the editors' behavior before and after releasing the tool (Section 5), which imply that the editors began to refer to generated titles after the release.

2 Encoder-Decoder Model

An encoder-decoder model (Bahdanau et al., 2015) is a conditional language model that predicts the correct output sequence from an input sequence, which is learned from many correct pairs of input and output sequences, e.g., news articles and their headlines. To train this model, we calcu-

late the following conditional likelihood

$$p(y \mid x) = \prod_{t=1}^{T-1} p(y_{t+1} \mid y_{\leq t}, x) \quad (1)$$

with respect to each pair (x, y) of an input sequence $x = x_1 \cdots x_S$ and output sequence $y = y_1 \cdots y_T$, where $y_{\leq t} = y_1 \cdots y_t$, and maximize its mean. The model $p(y \mid x)$ in Eq. (1) is computed by a combination of two recurrent neural networks (RNNs): an encoder and decoder. The encoder reads an input sequence x to recognize its content, and the decoder predicts an output sequence y corresponding to the content.

More formally, an encoder calculates a hidden state h_s for each element x_s in a x by using the state transition function f_{enc} of the encoder: $h_s = f_{\text{enc}}(x_s, h_{s-1})$. In a similar fashion, a decoder calculates a hidden state $\hat{h}_t$ for each element y_t in a y by using the state transition function f_{dec} of the decoder after setting the last hidden state of the encoder as the initial state of the decoder ($\hat{h}_0 = h_S$): $\hat{h}_t = f_{\text{dec}}(y_t, \hat{h}_{t-1})$. Then, a prediction of outputs for each $\hat{h}_t$ is calculated using the output function g_{dec} with an attention mechanism:

$$p(y_{t+1} \mid y_{\leq t}, x) = g_{\text{dec}}(\hat{h}_t, c_t), \quad (2)$$

where c_t is a weighted average of the encoder hidden states $\{h_1, \cdots, h_S\}$, defined by

$$c_t = \sum_{s=1}^{S} a_t(s) h_s, \quad (3)$$

where $a_t(s)$ represents a weight of an encoder hidden state h_s with respect to a decoder hidden state $\hat{h}_t$. c_t represents a soft alignment (or attention weight) to the source sequence at the target position t, so it is called a *context*.

3 Proposed Method

We propose an encoder-decoder model with multiple encoders. For simplicity, we describe our model assuming two encoders for news headlines and leads. Let d_t and d'_t be contexts calculated with Eq. (3) with the headline encoder and lead encoder, respectively. Our model combines the two context vectors inspired by a gating mechanism in long-short term memory networks (Hochreiter and Schmidhuber, 1997) as follows:

$$w_t = \sigma(W[d_t; d'_t; \hat{h}_t]), \quad (4)$$

$$w'_t = \sigma(W'[d_t; d'_t; \hat{h}_t]), \quad (5)$$

$$\bar{c}_t = w_t \odot d_t + w'_t \odot d'_t, \quad (6)$$

where function σ represents the sigmoid function, i.e., $\sigma(x) = 1/(1 + e^{-x})$, and the operator $\odot$ represents the element-wise product. Eq. (4) calculates a gating weight w_t for d_t, where W represents a weight matrix for a concatenated vector $[d_t; d'_t; \hat{h}_t]$. Similarly, Eq. (5) calculates a gating weight w'_t for d'_t. Eq. (6) calculates a mixed context $\bar{c}_t$ made from the two contexts, d_t and d'_t. Finally, the output function in our model is constructed by substituting c_t with $\bar{c}_t$ in Eq. (2).

Our model can be regarded as an extension of the multimodal fusion model (Hori et al., 2017), where multiple contexts are mixed using scalar weights, i.e., $\bar{c}_t = \alpha d_t + \beta d'_t$, where α and β are positive scalar weights calculated using an attention mechanism such as $a_t(s)$ in Eq. (3). Our model can obtain a more sophisticated mixed context than their model since that model only takes into account which encoder to weigh at a time step, while our model adjusts weights on the element level.

4 Experiments

4.1 Dataset

We prepared a dataset extracted from the news-aggregator of Yahoo! JAPAN by Web crawling. The dataset included 263K (headline, lead, short title) triples, and was split into three parts, i.e., for training (90%), validation (5%), and testing (5%). We preprocessed them by separating characters for training since our preliminary experiments showed that character-based training clearly performed better than word-based training.

The statistics of our dataset are as follows. The average lengths of headlines, leads, and short titles are 24.87, 128.49, and 13.05 Japanese characters, respectively. The dictionary sizes (for characters) of headlines, leads, and short titles are 3618, 4226, and 3156, respectively. Each news article has only one short title created by a professional editor. The percentage of short titles equal to their headlines is only 0.13%, while the percentage of extractively solvable instances, in which the characters in each short title are completely matched by those in the corresponding headline, was about 20%. However, the average edit distance (Levenshtein, 1966) between short titles and headlines was 23.74. This means that short titles cannot be easily created from headlines.

Hyper-parameter	Value
# of layers (RNN, CNN)	3
# of units (embedding)	200
# of units (RNN, CNN)	400
# of units (context)	400
Window width of CNN	7
Dropout rate	0.3
Learning rate	0.05
Momentum rate	0.8
Learning_decay rate	0.85
# of epochs	20
Batch size	64
Beam width	5

Table 2: Hyper-parameter settings.

4.2 Training

We implemented our model on the OpenNMT[2] toolkit. We used a convolutional neural network (CNN) (Kim, 2014), instead of an RNN, to construct the lead encoder since leads are longer than headlines and require much more computational time. Since the CNN encoder outputs all hidden states for an input sequence in the same format as the RNN encoder, we can easily apply these states to Eq. (3). Our headline encoder still remains as an RNN (i.e., bidirectional LSTM) for fair comparison with the default implementation. We used a stochastic gradient descent algorithm with Nesterov momentum (Nesterov, 1983) as an optimizer, after initializing parameters by uniform sampling on $(-0.1, 0.1)$. Table 2 lists the details of the hyper-parameter settings in our experiment. Other settings were basically the same as the default implementation of OpenNMT.

4.3 Evaluation

We conducted two crowdsourcing tasks to separately measure readability and usefulness. The readability task asked ten workers how readable each short title was on a four-point scale (higher is better), while the usefulness task asked them how useful the short title was compared to the corresponding article. The score of each generated short title was calculated by averaging the scores collected from the ten workers.

4.4 Compared Models

We prepared four models, our model `GateFusion` and three baselines `MultiModal`, `QueryBased`, and `OpenNMT`, listed below. We implemented the fusion mechanisms of `MultiModal` and `QueryBased` on OpenNMT using an RNN encoder for headlines and CNN encoder for leads (see Appendix A for detailed definitions).

- `GateFusion`: Our model with a gating mechanism described in Section 3. This is a fusion based on vector weights.
- `MultiModal`: A multimodal model proposed by (Hori et al., 2017), which can handle multimodal information such as image and audio as well as text by using separate encoders. The model combines contexts obtained from the encoders via an attention mechanism such as $a_t(s)$ in Eq. (3). This is a fusion based on scalar weights.
- `QueryBased`: A query-based model proposed by (Nema et al., 2017), which can finetune the attention on a document by using a query for query-focused summarization. We regard a headline as a document and a lead as a query since the headline is more similar to its short title. Specifically, the model finetunes an attention weight $a_t(s)$ for calculating a headline context d_t by using a pre-computed lead context d'_t. This is a fusion based on cascade connection.
- `OpenNMT`: An encoder-decoder model with a single encoder implemented in OpenNMT, whose input is a headline only, because a variant using a lead did not perform better than this setting.

[2] `https://github.com/OpenNMT/OpenNMT-py`

	Readablity	Usefulness	Average
`Editor`	3.62	3.18	3.40
`Prefix`	2.72	2.38	2.55
`OpenNMT`	3.53	3.16	3.35
`MultiModal`	3.51	3.12	3.32
`QueryBased`	3.52	3.11	3.32
`GateFusion`	3.50	†**3.22**	3.36
`HybridFusion`	†**3.55**	†**3.22**	†**3.39**

Table 3: Mean scores of readability (r), usefulness (u), and their average $\frac{r+u}{2}$ based on crowdsourcing. The "†" mark shows a statistical significance from all three baselines `OpenNMT`, `MultiModal`, and `QueryBased` on a one-tailed, paired t-test ($p < 0.01$).

4.5 Results

Table 3 lists the results from the crowdsourcing tasks for readability and usefulness (see Appendix B for the details of these scores). `Editor` and `Prefix` in the top block of rows show the results of correct short titles created by editors

and a naive model using the first 13.5 Japanese characters[3], respectively. The middle and bottom blocks represent the three baselines and our models, respectively. We explain our hybrid model HybridFusion later. Each model was prepared as an ensemble of ten models by random initialization, aiming for robust performance. Our GateFusion clearly performed better than the three baselines regarding usefulness and interestingly outperformed even Editor. This implies that GateFusion tends to aggressively copy elements from source sequences. However, this seemed to result in complicated expressions; thus, GateFusion performed the worst with respect to readability. To overcome this weakness, we developed a hybrid model HybridFusion that consists of GateFusion and another fusion model QueryBased, which performed relatively well in terms of readability. The results indicate that HybridFusion performed the best regarding readability and usefulness. It can be considered that QueryBased helps GateFusion generate headline-style outputs since QueryBased mainly uses the headline source.

Table 4 lists output examples generated by the best model OpenNMT from the three baselines and our best model HybridFusion (see Appendix C for more examples). In this case, the difference between OpenNMT and HybridFusion is easily comprehensible. The former selected "進化 (evolution)", and the latter selected "ダルビッシュ (Darvish)" from the headline. In Japanese headlines, the last word tends to be important, so using the last word is basically a good strategy. However, the lead indicates that "Darvish" is more important than "evolution" (actually, there is no word "evolution" in the lead); thus, HybridFusion was able to correctly select the long name "Darvish" and abbreviate it to "ダル (Dar)". In addition, it forcibly changed the style to the short title's style by putting the name into the forefront to easily get users' attention. This suggests that our neural-headline-generation model HybridFusion can successfully work even in this real-world application.

5 Deployment to Editing Support Tool

We deployed our short-title-generation model to an editing support tool in collaboration with the news service, as shown in Figure 2. In the tool, when an editor enters the URL of an article, the tool can automatically fetch the headline and lead of the article and display up to five candidates next to the edit form of a short title, as shown in the dotted box in the figure. These candidates are hypotheses (with high probabilities) generated by the beam search based on the model. Then, the editor can effectively create a short title by referring to the generated candidates. This supporting feature is expected to be useful especially for inexperienced editors since the quality of short titles is heavily dependent on editors' experience.

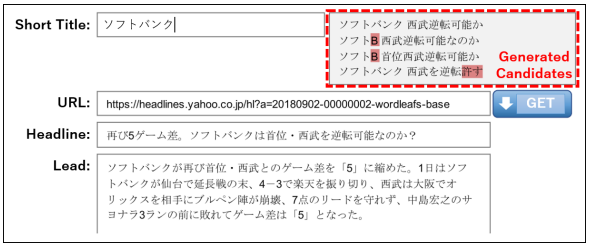

Figure 2: Screenshot of editing support tool displaying generated candidates for creating a short title.

From now on, we briefly describe three features of the tool to improve its usability when displaying candidates: cutoff of unpromising candidates, skipping redundant candidates, and highlighting unknown characters. After that, we discuss the effect of the deployment analyzing user behavior before and after releasing the tool.

5.1 Cutoff of Unpromising Candidates

The quality of displayed candidates is one of the main factors that affect the usability of the tool. If the tool frequently displays unpromising candidates, editors will gradually start ignoring them. Therefore, we cutoff unpromising candidates whose perplexity scores are higher than a certain threshold, where the perplexity score of a candidate is calculated by the inverse of the geometric mean of the generation probabilities for all characters in the candidate. We set the threshold considering the results of the editors' manual evaluation, where they checked if each candidate was acceptable or not. Specifically, we used 1.47 (=1/0.68) as the threshold, which means that the (geometric) mean character likelihood in the candidate should be higher than 0.68. If all candidates are judged as unpromising, the tool displays a message like "No promising candidates."

[3] 13.5 is the limit in the news-aggregator, where space, numbers, and alphabet characters are counted as 0.5.

	Input and generated title (Japanese)	English translation
Headline	逆境をチャンスに変えたダルビッシュの進化	Evolution of Darvish; turning adversity into opportunity.
Lead	レンジャーズのダルビッシュ有（29）が 28 日、本拠地で行われたパイレーツ戦で [...]	Yu Darvish (29) in Rangers took a mound for the first time in 1 year and 9 months with Pirates [...]
Editor	術前より進化 ダルの肉体改造	Dar sculpted his body better than before surgery.
OpenNMT	逆境をチャンスに変えた進化	Evolution; turning adversity into opportunity.
HybridFusion	ダル 逆境をチャンスに変えた	Dar turned adversity into opportunity.

Table 4: Examples of generated titles. Headline and Lead denote headline and lead as input. Editor is reference title created by an editor. OpenNMT and HybridFusion are the OpenNMT model and our hybrid model.

5.2 Skipping Redundant Candidates

The purpose of the tool is to give editors some new ideas for creating short titles, so it is not useful to display redundant candidates similar to others. Therefore, we skip candidates whose edit distance (Levenshtein, 1966) to the other candidates is lower than a threshold when selecting hypotheses in descending order of probability. Formally, the edit distance between two texts is defined as the minimum number of single-character edits (insertions, deletions, or substitutions) required to change one text into the other. We set the threshold to 2 so as to restrict variations of Japanese particles as there are many particles with a similar meaning in Japanese[4], e.g., "は (ha)" and "が (ga)". Although we used a unit cost for the edit distance, we can adjust the cost of each edit operation so that the tool can ignore variations of prepositions if we want to use English texts.

5.3 Highlighting Unknown Characters

One difficulty of neural models is that there is a possibility of generating incorrect or fake titles, which do not correspond to the article. This is a serious issue for news editing support since displayed candidates can mislead editors. For example, if the tool displays "藤波 (Fujinami)" for the news about "藤浪 (Fujinami)", where they are different names with the same pronunciation, editors might choose the incorrect one. As a simple solution, we highlighted unknown characters that do not appear in both headline and lead in red. In Figure 2, two phrases ("B" and "許す") are highlighted since they do not appear in the headline and lead. When a candidate includes highlighted characters, editors can carefully check if the candidate is semantically correct. Note that we did not exclude candidates with unknown characters so that the model can aggressively generate paraphrases and abbreviations. For example, the tool suggests "ソフト B(Soft B.)" as an abbreviation of "ソフトバンク (Softbank)" in the figure.

[4] https://en.wikipedia.org/wiki/Japanese_particles

	ROUGE-L (± SE)	# articles
Before	52.71% (± 0.56)	1773
After	57.65% (± 0.53)	1959

Table 5: Sequence matching rates (ROUGE-L) of editors' titles and generated titles, which are averaged over articles over three weeks before/after releasing tool.

5.4 Effect of Deployment

To investigate the effect of the deployment, we compared the sequence matching rates between editors' correct titles and generated candidates before and after releasing the tool. The sequence matching rate is basically calculated by ROUGE-L (Lin, 2004), which is defined as the rate of the length of the longest common subsequence between two sequences, i.e., a correct title and a generated candidate. Because we have multiple candidates for each article, we calculate the sequence matching rate as the maximum of their ROUGE-L scores, assuming that editors may refer to the most promising candidate. Note that the candidates were filtered by the aforementioned features, so we omitted a few articles without candidates.

Table 5 shows the results of the sequence matching rates averaged over the articles over three weeks before and after releasing the tool. The results indicate that the ROUGE-L score increased by about 5 percentage points after the release. This implies that editors created their titles by referring to the displayed candidates to some extent. In fact, the ratio of the exact matched titles (ROUGE-L = 100%) in all articles (before/after the release) increased after the release by a factor of 1.62(i.e., from 3.78% to 6.13%). Similarly, the ratio of the 80% matched titles (ROUGE-L $\geq$ 80%) also increased by a factor of 1.32 (i.e., from 14.04% to 18.53%). This suggests that professional editors obtained new ideas from generated titles of the tool.

6 Related Work

We briefly review related studies from three aspects: news headline generation, editing support, and application of headline generation. In summary, our work is the first attempt to deploy a neural news-headline-generation model to a real-world application, i.e., news editing support tool.

News-headline-generation tasks have been extensively studied since early times (Wang et al., 2005; Soricut and Marcu, 2006; Woodsend et al., 2010; Alfonseca et al., 2013; Sun et al., 2015; Colmenares et al., 2015). In this line of research, Rush et al. (2015) proposed a neural model to generate news headlines and released a benchmark dataset for their task, and consequently this task has recently received increasing attention (Chopra et al., 2016; Takase et al., 2016; Kiyono et al., 2017; Zhou et al., 2017; Suzuki and Nagata, 2017; Ayana et al., 2017; Raffel et al., 2017; Cao et al., 2018; Kobayashi, 2018). However, their approaches were basically based on the encoder-decoder model, which is trained with a lot of (article, headline) pairs. This means that there are few situations for putting their models into the real world because news articles typically already have corresponding headlines, and most editors create a headline before its content (according to a senior journalist). Therefore, our work can strongly support their approaches from a practical perspective.

Considering technologies used for editing support, there have been many studies for various purposes, such as spelling error correction (Farra et al., 2014; Hasan et al., 2015; Etoori et al., 2018), grammatical error correction (Dahlmeier and Ng, 2012; Susanto et al., 2014; Choshen and Abend, 2018), fact checking (Baly et al., 2018; Thorne and Vlachos, 2018; Lee et al., 2018), fluency evaluation (Vadlapudi and Katragadda, 2010; Heilman et al., 2014; Kann et al., 2018), and so on. However, when we consider their studies on our task, they are only used after editing (writing a draft). On the other hand, the purpose of our tool is different from theirs since our tool can support editors before or during editing. The usage of (interactive) machine translation systems (Denkowski et al., 2014; González-Rubio et al., 2016; Wuebker et al., 2016; Ye et al., 2016; Takeno et al., 2017) for supporting manual post-editing are similar to our purpose, but their task is completely different from ours. In other words, their task is a translation without information loss, whereas our task is a summarization that requires information compression. We believe that a case study on summarization is still important for the summarization community.

There have been several studies reporting case studies on headline generation for different real services: (a) question headlines on question answering service (Higurashi et al., 2018), (b) product headlines on e-commerce service (Wang et al., 2018), and (c) headlines for product curation pages (Mathur et al., 2018; Camargo de Souza et al., 2018). The first two (a) and (b) are extractive approaches, and the last one (c) is an abstractive approach, where the input is a set of slot/value pairs, such as "color/white." That is, our task is more difficult to use in the real-world. In addition, application to news services tends to be sensitive since news articles contain serious contents such as incidents, accidents, and disasters. Thus, our work should be valuable as a rare case study applying a neural model to such a news service.

7 Conclusion

We addressed short-title generation from news articles for a news aggregator to support the editorial process. We proposed an encoder-decoder model with multiple encoders for separately encoding multiple information sources, i.e., news headlines and leads. Comparative experiments using crowdsourcing showed that our hybrid model performed better than the baselines, especially using the usefulness measure. We deployed our model to an editing support tool and empirically confirmed that professional editors began to refer to the generated titles after the release. Future research will include verifying how much our headline generation model can affect practical performance indicators, such as click-through rate. In this case, we need to develop a much safer model since our model sometimes yields erroneous outputs or fake news titles, which cannot be directly used in the commercial service.

Acknowledgements

We would like to thank editors and engineers in the news service who continuously supported our experiments. We are also in debt to Chahine Koleejan who helped with proofreading. Finally, special thanks go to the anonymous reviewers for their insightful comments.

References

Enrique Alfonseca, Daniele Pighin, and Guillermo Garrido. 2013. HEADY: News headline abstraction through event pattern clustering. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (ACL 2013)*, pages 1243–1253. Association for Computational Linguistics.

Ayana, Shi-Qi Shen, Yan-Kai Lin, Cun-Chao Tu, Yu Zhao, Zhi-Yuan Liu, and Mao-Song Sun. 2017. Recent Advances on Neural Headline Generation. *Journal of Computer Science and Technology*, 32(4):768–784.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *Proceedings of the 3rd International Conference on Learning Representations (ICLR 2015)*.

Ramy Baly, Mitra Mohtarami, James Glass, Lluís Màrquez, Alessandro Moschitti, and Preslav Nakov. 2018. Integrating Stance Detection and Fact Checking in a Unified Corpus. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT 2018))*, pages 21–27. Association for Computational Linguistics.

Ziqiang Cao, Furu Wei, Wenjie Li, and Sujian Li. 2018. Faithful to the Original: Fact Aware Neural Abstractive Summarization. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence (AAAI 2018)*, pages 4784–4791. AAAI Press.

Sumit Chopra, Michael Auli, and Alexander M. Rush. 2016. Abstractive Sentence Summarization with Attentive Recurrent Neural Networks. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT 2016)*, pages 93–98. Association for Computational Linguistics.

Leshem Choshen and Omri Abend. 2018. Automatic Metric Validation for Grammatical Error Correction. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (ACL 2018)*, pages 1372–1382. Association for Computational Linguistics.

Carlos A. Colmenares, Marina Litvak, Amin Mantrach, and Fabrizio Silvestri. 2015. HEADS: Headline Generation as Sequence Prediction Using an Abstract Feature-Rich Space. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT 2015)*, pages 133–142. Association for Computational Linguistics.

Daniel Dahlmeier and Hwee Tou Ng. 2012. Better Evaluation for Grammatical Error Correction. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics (NAACL-HLT 2012)*, pages 568–572. Association for Computational Linguistics.

Michael Denkowski, Alon Lavie, Isabel Lacruz, and Chris Dyer. 2014. Real Time Adaptive Machine Translation for Post-Editing with cdec and TransCenter. In *Proceedings of the EACL 2014 Workshop on Humans and Computer-assisted Translation*, pages 72–77. Association for Computational Linguistics.

Pravallika Etoori, Manoj Chinnakotla, and Radhika Mamidi. 2018. Automatic Spelling Correction for Resource-Scarce Languages using Deep Learning. In *Proceedings of ACL 2018, Student Research Workshop*, pages 146–152. Association for Computational Linguistics.

Noura Farra, Nadi Tomeh, Alla Rozovskaya, and Nizar Habash. 2014. Generalized Character-Level Spelling Error Correction. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (ACL 2014)*, pages 161–167. Association for Computational Linguistics.

Jesús González-Rubio, Daniel Ortiz Martinez, Francisco Casacuberta, and Jose Miguel Benedi Ruiz. 2016. Beyond Prefix-Based Interactive Translation Prediction. In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning (CoNLL 2016)*, pages 198–207. Association for Computational Linguistics.

Saša Hasan, Carmen Heger, and Saab Mansour. 2015. Spelling Correction of User Search Queries through Statistical Machine Translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP 2015)*, pages 451–460. Association for Computational Linguistics.

Michael Heilman, Aoife Cahill, Nitin Madnani, Melissa Lopez, Matthew Mulholland, and Joel Tetreault. 2014. Predicting Grammaticality on an Ordinal Scale. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (ACL 2014)*, pages 174–180. Association for Computational Linguistics.

Tatsuru Higurashi, Hayato Kobayashi, Takeshi Masuyama, and Kazuma Murao. 2018. Extractive headline generation based on learning to rank for community question answering. In *Proceedings of the 27th International Conference on Computational Linguistics (COLING 2018)*, pages 1742–1753. Association for Computational Linguistics.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Computation*, 9(8):1735–1780.

Chiori Hori, Takaaki Hori, Teng-Yok Lee, Ziming Zhang, Bret Harsham, John R. Hershey, Tim K. Marks, and Kazuhiko Sumi. 2017. Attention-based multimodal fusion for video description. In *ICCV*.

Katharina Kann, Sascha Rothe, and Katja Filippova. 2018. Sentence-Level Fluency Evaluation: References Help, But Can Be Spared! In *Proceedings of the 22nd Conference on Computational Natural Language Learning (CoNLL 2018)*, pages 313–323. Association for Computational Linguistics.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1746–1751. Association for Computational Linguistics.

Shun Kiyono, Sho Takase, Jun Suzuki, Naoaki Okazaki, Kentaro Inui, and Masaaki Nagata. 2017. Source-side Prediction for Neural Headline Generation . *CoRR*, abs/1712.08302.

Hayato Kobayashi. 2018. Frustratingly Easy Model Ensemble for Abstractive Summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP 2018)*, pages 4165–4176. Association for Computational Linguistics.

Nayeon Lee, Chien-Sheng Wu, and Pascale Fung. 2018. Improving Large-Scale Fact-Checking using Decomposable Attention Models and Lexical Tagging. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP 2018)*, pages 1133–1138. Association for Computational Linguistics.

Vladimir Iosifovich Levenshtein. 1966. Binary Codes Capable of Correcting Deletions, Insertions and Reversals. *Soviet Physics Doklady*, 10(8):707–710.

Chin-Yew Lin. 2004. ROUGE: A Package for Automatic Evaluation of Summaries. In *Proceedings of the ACL Workshop on Text Summarization Branches Out*.

Prashant Mathur, Nicola Ueffing, and Gregor Leusch. 2018. Multi-lingual neural title generation for e-Commerce browse pages. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT 2018)*, pages 162–169. Association for Computational Linguistics.

Preksha Nema, Mitesh M. Khapra, Anirban Laha, and Balaraman Ravindran. 2017. Diversity driven attention model for query-based abstractive summarization. In *ACL*, pages 1063–1072.

Yurii Nesterov. 1983. A method of solving a convex programming problem with convergence rate o(1/k2). *Soviet Mathematics Doklady*, 27.

Colin Raffel, Minh-Thang Luong, Peter J. Liu, Ron J. Weiss, and Douglas Eck. 2017. Online and Linear-Time Attention by Enforcing Monotonic Alignments. In *Proceedings of the 34th International Conference on Machine Learning (ICML 2017)*, pages 2837–2846.

Alexander M. Rush, Sumit Chopra, and Jason Weston. 2015. A Neural Attention Model for Abstractive Sentence Summarization. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP 2015)*, pages 379–389. Association for Computational Linguistics.

Radu Soricut and Daniel Marcu. 2006. Stochastic Language Generation Using WIDL-Expressions and its Application in Machine Translation and Summarization. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics (COLING-ACL 2006)*, pages 1105–1112. Association for Computational Linguistics.

José G. Camargo de Souza, Michael Kozielski, Prashant Mathur, Ernie Chang, Marco Guerini, Matteo Negri, Marco Turchi, and Evgeny Matusov. 2018. Generating E-Commerce Product Titles and Predicting their Quality. In *Proceedings of the 11th International Conference on Natural Language Generation*, pages 233–243. Association for Computational Linguistics.

Rui Sun, Yue Zhang, Meishan Zhang, and Donghong Ji. 2015. Event-Driven Headline Generation. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (ACL-IJCNLP 2015)*, pages 462–472. Association for Computational Linguistics.

Raymond Hendy Susanto, Peter Phandi, and Hwee Tou Ng. 2014. System Combination for Grammatical Error Correction. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP 2014)*, pages 951–962. Association for Computational Linguistics.

Jun Suzuki and Masaaki Nagata. 2017. Cutting-off redundant repeating generations for neural abstractive summarization. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics (EACL 2017)*, pages 291–297. Association for Computational Linguistics.

Sho Takase, Jun Suzuki, Naoaki Okazaki, Tsutomu Hirao, and Masaaki Nagata. 2016. Neural Headline Generation on Abstract Meaning Representation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing (EMNLP 2016)*, pages 1054–1059. Association for Computational Linguistics.

Shunsuke Takeno, Masaaki Nagata, and Kazuhide Yamamoto. 2017. Controlling target features in neural machine translation via prefix constraints. In *Proceedings of the 4th Workshop on Asian Translation (WAT2017)*, pages 55–63. Asian Federation of Natural Language Processing.

Jiwei Tan, Xiaojun Wan, and Jianguo Xiao. 2017. Abstractive Document Summarization with a Graph-Based Attentional Neural Model. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (ACL 2017)*, pages 1171–1181. Association for Computational Linguistics.

James Thorne and Andreas Vlachos. 2018. Automated Fact Checking: Task Formulations, Methods and Future Directions. In *Proceedings of the 27th International Conference on Computational Linguistics (COLING 2018)*, pages 3346–3359. Association for Computational Linguistics.

Ravikiran Vadlapudi and Rahul Katragadda. 2010. On Automated Evaluation of Readability of Summaries: Capturing Grammaticality, Focus, Structure and Coherence. In *Proceedings of the NAACL HLT 2010 Student Research Workshop*, pages 7–12. Association for Computational Linguistics.

Jingang Wang, Junfeng Tian, Long Qiu, Sheng Li, Jun Lang, Luo Si, and Man Lan. 2018. A Multi-Task Learning Approach for Improving Product Title Compression with User Search Log Data. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence (AAAI 2018)*, pages 451–458.

Ruichao Wang, John Dunnion, and Joe Carthy. 2005. Machine Learning Approach to Augmenting News Headline Generation. In *Proceedings of the Second International Joint Conference on Natural Language Processing (IJCNLP 2005)*. Association for Computational Linguistics.

Kristian Woodsend, Yansong Feng, and Mirella Lapata. 2010. Title Generation with Quasi-Synchronous Grammar. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing (EMNLP 2010)*, pages 513–523. Association for Computational Linguistics.

Joern Wuebker, Spence Green, John DeNero, Sasa Hasan, and Minh-Thang Luong. 2016. Models and Inference for Prefix-Constrained Machine Translation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL 2016)*, pages 66–75. Association for Computational Linguistics.

Na Ye, Guiping Zhang, and Dongfeng Cai. 2016. Interactive-predictive machine translation based on syntactic constraints of prefix. In *Proceedings of the 26th International Conference on Computational Linguistics (COLING 2016)*, pages 1797–1806. The COLING 2016 Organizing Committee.

Qingyu Zhou, Nan Yang, Furu Wei, and Ming Zhou. 2017. Selective encoding for abstractive sentence summarization. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (ACL 2017)*, pages 1095–1104. Association for Computational Linguistics.