Diverse, Controllable, and Keyphrase-Aware: A Corpus and Method for News Multi-Headline Generation

Dayiheng Liu[♠*] **Yeyun Gong**[†] **Yu Yan**[‡] **Jie Fu**[♢]
Bo Shao[†] **Daxin Jiang**[‡] **Jiancheng Lv**[♠] **Nan Duan**[†]
[♠]College of Computer Science, Sichuan University [†]Microsoft Research Asia
[♢] Mila [‡]Microsoft
losinuris@gmail.com

Abstract

News headline generation aims to produce a short sentence to attract readers to read the news. One news article often contains multiple keyphrases that are of interest to different users, which can naturally have multiple reasonable headlines. However, most existing methods focus on the single headline generation. In this paper, we propose generating multiple headlines with keyphrases of user interests, whose main idea is to generate multiple keyphrases of interest to users for the news first, and then generate multiple keyphrase-relevant headlines. We propose a multi-source Transformer decoder, which takes three sources as inputs: (a) keyphrase, (b) keyphrase-filtered article, and (c) original article to generate keyphrase-relevant, high-quality, and diverse headlines. Furthermore, we propose a simple and effective method to mine the keyphrases of interest in the news article and build a first large-scale keyphrase-aware news headline corpus, which contains over 180K aligned triples of ⟨news article, headline, keyphrase⟩. Extensive experimental comparisons on the real-world dataset show that the proposed method achieves state-of-the-art results in terms of quality and diversity[1].

1 Introduction

News Headline Generation is an under-explored subtask of text summarization (See et al., 2017; Gehrmann et al., 2018; Zhong et al., 2019). Unlike text summaries that contain multiple context-related sentences to cover the main ideas of a document, news headlines often contain a single short sentence to encourage users to read the news. Since one news article typically contains multiple keyphrases or topics of interest to different users, it is useful to generate multiple headlines covering different keyphrases for the news article. Multi-headline generation aims to generate multiple independent headlines, which allows us to recommend news with different news headlines based on the interests of users. Besides, multi-headline generation can provide multiple hints for human news editors to assist them in writing news headlines.

However, most existing methods (Takase et al., 2016; Ayana et al., 2016; Murao et al., 2019; Colmenares et al., 2019; Zhang et al., 2018) focus on single-headline generation. The headline generation process is treated as an one-to-one mapping (the input is an article and the output is a headline), which trains and tests the models without any additional guiding information or constraints. We argue that this may lead to two problems. Firstly, since it is reasonable to generate multiple headlines for the news, training to generate the single ground-truth might result in a lack of more detailed guidance. Even worse, a single ground-truth without any constraint or guidance is often not enough to measure the quality of the generated headline for model testing. For example, even if a generated headline is considered reasonable by humans, it can get a low score in ROUGE (Lin, 2004), because it might focus on the keyphrases or aspects that are not consistent with the ground-truth.

In this paper, we incorporate the keyphrase information into the headline generation as additional guidance. Unlike one-to-one mapping employed in previous works, we treat the headline generation process as a two-to-one mapping, where the inputs are news articles and keyphrases, and the output is a headline. We propose a keyphrase-aware news multi-headline generation method, which contains two modules: (a) Keyphrase Generation Model, which aims to generate multiple keyphrases of interest to users for the news article. (b) Keyphrase-Aware Multi-Headline Generation Model, which

[*] Work is done during internship at Microsoft Research Asia.

[1]The source code will be available at https://github.com/dayihengliu/KeyMultiHeadline.

Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, pages 6241–6250,
November 16–20, 2020.

takes the news article and a keyphrase as input and generates a keyphrase-relevant news headline. For training models, we build a first large-scale news keyphrase-aware headline corpus that contains 180K aligned triples of ⟨news article, headline, keyphrase⟩ .

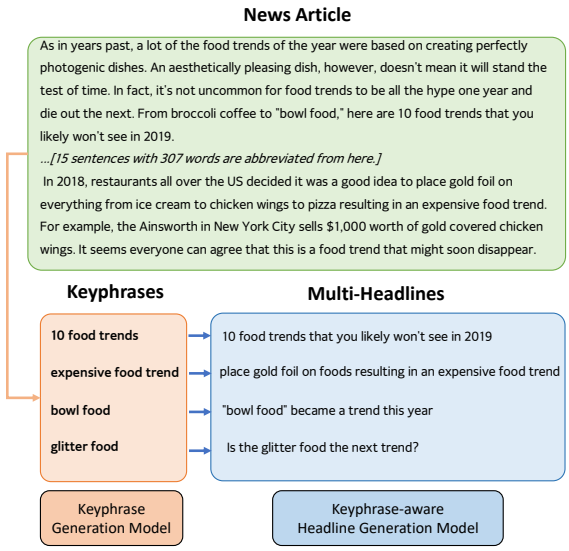

Figure 1: Keyphrase-aware multi-headline generation

The proposed approach faces two major challenges. The first one is how to build the keyphrase-aware news headline corpus. To our best knowledge, no corpus contains the news article and headline pairs, which are aligned with a keyphrase of interest to users. The second is how to design the keyphrase-aware news headline generation model to ensure that the generated headlines are keyphrase-relevant, high-quality, and diverse. For the first challenge, we propose a simple but efficient method to mine the keyphrases of interest to users in news articles based on the user search queries and news click information that are collected from a real-world search engine. With this method, we build the keyphrase-aware news headline corpus.

For the second challenge, we design a multi-source Transformer (Vaswani et al., 2017) decoder to improve the generation quality and the keyphrase sensitivity of the model, which takes three source information as inputs: (a) keyphrase, (b) keyphrase-filtered article, and (c) original article. For the proposed multi-source Transformer decoder, we further design and compare several variants of attention-based fusing mechanism. Extensive experiments on real-world dataset have shown that the proposed method can generate high-quality, keyphrase-relevant, and diverse news headlines.

2 Keyphrase-Aware Headline Corpus

Our keyphrase-aware news headline corpus called **KeyAware News** is built by the following steps:

(1) **Data Collection**. We collect 16,000,000 raw samples which contain news articles with user search query information from Microsoft Bing News search engine[2]. Each sample can be presented as a tuple $\langle Q, X, Y, C \rangle$ where Q is a user search query, X is a news article that the search engine returns to the user based on the search query Q, Y is a human-written headline for X, and C represents the number of times the user clicks on the news under the search query Q. Each news article X has 10 different queries Q on average.

(2) **Keyphrase Mining**. We mine the keyphrase of interest to users with user search queries. We assume that if many users find and click on one news article through different queries containing the same phrase, such a phrase is the keyphrase for the article. For each article, we collect its corresponding user search queries and remove the stop words and special symbols from the queries. Then we find the common phrases (4-gram, 3-gram, or 2-gram) in these queries. These common phrases are scored based on how many times they appear in these queries and normalized by length. The score is also weighted by the user click number C, which means the phrases that appear in the queries have more users click on the article are more important. Finally, we use the n-gram with the highest score as the keyphrase Z of the article X.

(3) **Article-Headline-Keyphrase Alignment**. In order to obtain the aligned article-headline-keyphrase tuple $\langle X, Y, Z \rangle$. We filter out the sample whose article or headline does not contain the Z. Moreover, we remove such pairs whose article length are greater than 600 or less than 100 tokens, or whose headline length are greater than 20 or less than 3 tokens. After the alignment and data cleaning, we obtain the **KeyAware News** which contains about 180K aligned article-headline-keyphrase triples. We split it into Train, Test, and Dev sets, each containing 165,913, 10,000, and 5,000 samples.

[2]All news articles are high-quality, real-world news and all the news headlines are written by humans.

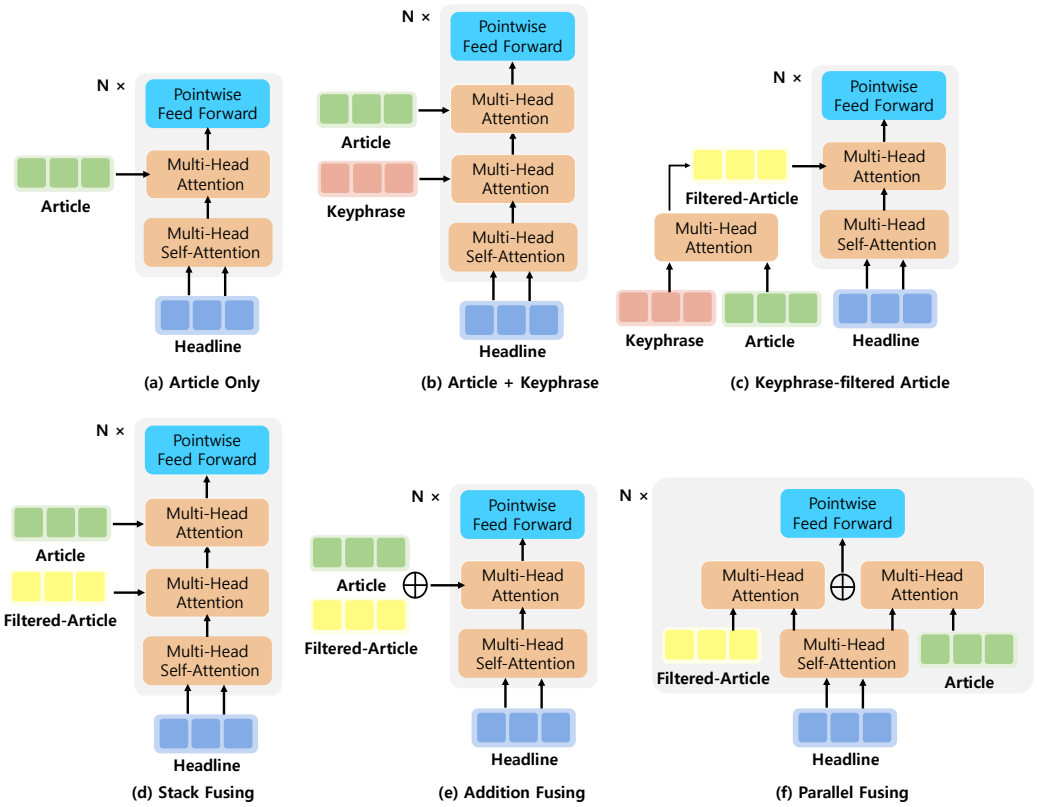

Figure 2: Visualization of the computational steps in each block of our multi-source Transformer decoders.

3 Methodology

3.1 Overview

The overall keyphrase-aware multi-headline generation procedure is shown in Figure 1, which involves two modules: (a) **keyphrase generation model** generates multiple keyphrases of interest to users for the news article. (b) **keyphrase-aware headline generation model** takes the news article and each generated keyphrase as input, and generates multiple keyphrase-relevant news headlines.

3.2 Headline Generation

The headline generation can be formalized as a sequence-to-sequence learning (Sutskever et al., 2014) task. Given an input news article X and a specific keyphrase Z, we aim to produce a keyphrase-relevant headline Y.

3.2.1 Headline Generation BASE Model

We first introduce the basic version of our headline generation model (we call BASE), which is keyphrase-agnostic. BASE is built upon the Transformer Seq2Seq model (Vaswani et al., 2017), which has made remarkable progress in sequence-to-sequence learning. Transformer contains a multi-head self-attention encoder and a multi-head self-attention decoder. As discussed in Vaswani et al. (2017), an attention function maps a query and a set of key-value pairs to an output as:

$$\textbf{Attention}(\bar{Q}, \bar{K}, \bar{V}) = \textbf{Softmax}(\frac{\bar{Q}\bar{K}^T}{\sqrt{d_k}})\bar{V},$$

where the queries $\bar{Q}$, keys $\bar{K}$, and values $\bar{V}$ are all vectors, and d_k is the dimension of the key vector. Multi-head attention mechanism projects queries, keys, and values to h different subspaces and calculates corresponding attention as:

$$\textbf{MultiHead}(\bar{Q}, \bar{K}, \bar{V}) = \textbf{Concat}(\mathrm{h_1}, ..., \mathrm{h_h})\mathrm{W^O},$$
$$\text{where } \mathrm{h_i} = \textbf{Attention}(\bar{\mathrm{Q}}\mathrm{W_i^Q}, \bar{\mathrm{K}}\mathrm{W_i^K}, \bar{\mathrm{V}}\mathrm{W_i^V}).$$

The encoder is composed of a stack of N identical blocks. Each block has two sub-layers: multi-head self-attention mechanism and a position-wise fully connected feed-forward network. All sub-layers are interconnected with residual connections (He et al., 2016) and layer normalization (Ba et al., 2016).

Similarly, the decoder is also composed of a stack of N identical block. In addition to the two sub-layers in each encoder block, the decoder contains a third sub-layer which performs multi-head

attention over the output of the encoder. Figure 2 (a) shows the architecture of the block in the decoder.

BASE uses the pre-trained BERT-base model (Devlin et al., 2018) to initialize the parameters of the encoder. Also, it uses the transformer decoder with a copy mechanism (Gu et al., 2016), whose hidden size, the number of multi-head h, and the number of blocks N are the same as its encoder.

3.2.2 Keyphrase-Aware Headline Generation Model

In order to explore more effective ways of incorporating keyphrase information into BASE, we design 5 variants of multi-source Transformer decoders.

Article + Keyphrase. The basic idea is to add the keyphrase into the decoder directly. The keyphrase X_{key} is represented as a sequence of word embeddings. As shown in Figure 2 (b), we add an extra sub-layer that performs multi-head attention over the X_{key} in each block of the decoder.

$$X_{dec}^{(n+1)} = \textbf{MultiHead}(X_{dec}^{(n)}, X_{key}, X_{key}), \quad (1)$$

where $X_{dec}^{(n)}$ is the output of the n-th block in the decoder. Since the original article has contained sufficient information for the model to learn to generate the headline, the model may tend to mainly use the article information and ignore the keyphrase and become less sensitive to keyphrases. As a byproduct, the generated headlines may lack diversity and keyphrase relevance.

Keyphrase-Filtered Article. Intuitively, when people read news articles, they tend to focus on the parts of the article that are matched to the keyphrases of their interests. Inspired by this, before inputting the original article representation into the decoder, we use the attention mechanism to filter the article with the keyphrase (see Figure 2 (c)).

$$\hat{X}_{enc} = \textbf{MultiHead}(X_{key}, X_{enc}, X_{enc}), \quad (2)$$

where X_{enc} is the output of the last block in the encoder. The resulting representation $\hat{X}_{enc}$ can be seen as the keyphrase-filtered article, which mainly keeps the article information that is related to the keyphrase. Since the decoder cannot directly access the representation of the original article, the model is forced to utilize the information of the keyphrase. Therefore, the sensitivity of the model to keyphrase is improved.

Fusing Keyphrase-Filtered Article and Original Article. Although feeding the keyphrase-filtered article representation $\hat{X}_{enc}$ instead of the original article representation X_{enc} to the decoder can improve the sensitivity of the model to keyphrase, some useful and global information in the original article may also be filtered out. It might reduce the quality of the generated headlines. To further balance the keyphrase sensitivity and headline quality of the model, we use X_{enc} and $\hat{X}_{enc}$ as two input sources for the decoder and fuse them. As shown in Figure 2 (d)-(f), we design three decoder variants based on different fusing mechanism to fuse the X_{enc} and the $\hat{X}_{enc}$.

(I) **Addition-Fusing Mechanism**. We directly perform a point-wise addition between the X_{enc} and the $\hat{X}_{enc}$. Then we feed it into the decoder.

(II) **Stack-Fusing Mechanism**. We perform a multi-head attention on $\hat{X}_{enc}$ and X_{enc} one by one in each block of the decoder. All of the sub-layers are interconnected with residual connections.

$$\hat{X}_{dec}^{(n)} = \textbf{MultiHead}(X_{dec}^{(n)}, \hat{X}_{enc}, \hat{X}_{enc}) \quad (3)$$

$$X_{dec}^{(n+1)} = \textbf{MultiHead}(\hat{X}_{dec}^{(n)}, X_{enc}, X_{enc}) \quad (4)$$

(III) **Parallel-Fusing Mechanism**. For each block of the decoder, we perform a multi-head attention in parallel on $\hat{X}_{enc}$ and X_{enc}. Then, we perform a point-wise addition between them. Similarly, all of the sub-layers are interconnected with residual connections.

$$\hat{X}_{dec}^{(n)} = \textbf{MultiHead}(X_{dec}^{(n)}, \hat{X}_{enc}, \hat{X}_{enc}) \quad (5)$$

$$\bar{X}_{dec}^{(n)} = \textbf{MultiHead}(X_{dec}^{(n)}, X_{enc}, X_{enc}) \quad (6)$$

$$X_{dec}^{(n+1)} = \bar{X}_{dec}^{(n)} + \hat{X}_{dec}^{(n)} \quad (7)$$

3.3 Keyphrase Generation

In this subsection, we show how to generate the keyphrases for a given news article X. Here we briefly describe three methods for keyphrase generation. It should be noted that in this paper, we mainly focus on news headline generation rather than keyphrase generation.

(1) **TF-IDF Ranking**. We use Term Frequency Inverse Document Frequency (TF-IDF) (Zhang et al., 2007) to weight all n-grams (n = 2, 3, and 4) in the news article X. Then we filter out n-grams with TF-IDF below the threshold or containing any punctuation or special character. For different n of the n-gram, we set different thresholds for filtering. We take this unsupervised method as a baseline.

Method	EM@1	EM@3	EM@5	R@1	R@3	R@5
TF-IDF	18.63	42.05	52.60	30.13	53.82	63.91
SEQ2SEQ	57.27	**78.45**	**84.26**	59.60	81.32	87.04
SLOT	**60.75**	76.94	83.18	**65.13**	**84.05**	**89.08**

Table 1: Keyphrase Generation Results

(2) **Seq2Seq**. Since our **KeyAware News** corpus contains the article-keyphrase pairs, we treat the keyphrase generation as a sequence-to-sequence learning task. We train the model BASE with article-keyphrase pairs. During inference, we use beam search with length penalty to generate n-grams ($n = 2$, 3, and 4) as the keyphrases.
(3) **Slot Tagging**. Because the keyphrases also appear in the news articles, we can formulate the keyphrase generation task as a slot tagging task (Zhang et al., 2016; Williams, 2019). We fine-tune the BERT-base model to achieve that. Concretely, we use the output sequence of the model to predict the beginning and end position of the keyphrase in the article. During inference, we follow the answer span prediction method used in Seo et al. (2017) to predict n-grams ($n = 2$, 3, and 4) with the highest probabilities as the keyphrases.

4 Experiments

4.1 Keyphrase Generation

In the first experiment, we evaluate the performance of three keyphrase generation methods: (a) unsupervised TF-IDF Ranking, (b) supervised sequence-to-sequence model (SEQ2SEQ), and (c) supervised slot tagging model (SLOT).
Implementation and Hyperparameters. The SEQ2SEQ has the same architecture hyperparameters as BASE model. And the architecture hyperparameters of SLOT are the same as those of the BERT-base[3]. We use article-keyphrase pairs in the train set of **KeyAware News** to train SEQ2SEQ and SLOT.
Metrics. For evaluation, each method generates top-K keyphrases for every news article in the test set. We use a top-K exact-match rate (EM@K) as an evaluation metric, which tests whether one of the K generated keyphrases matches the golden keyphrase exactly. Some of the generated key phrases may not exactly match the golden keyphrase but have overlapping tokens with it (it may be a sub-sequence of the golden keyphrase or vice versa). We thus report the Recall@K (R@K), which tests the percentage of the tokens in golden keyphrase covered by the K generated keyphrases.
Results. The results are shown in Table 1. We can see that the EM@1 of TF-IDF is only 18.63%, but SLOT achieves 60.75%. Both of SEQ2SEQ and SLOT significantly outperform the TF-IDF in all metrics. SEQ2SEQ achieves comparable performances in EM@K, but performs worse than SLOT in R@K. SLOT achieves 83.18% EM@5 and 89.08% R@5. In the following experiments, we use SLOT to generate keyphrases for our keyphrase-aware news headline generation models.

4.2 News Headline Generation

Baselines. In the following experiments, we compare various variants of the proposed keyphrase-aware models we introduced in Section 3.2.1 as follows: (1) **BASE**, as shown in Figure 2 (a), which is keyphrase-agnostic and only takes the news article as input. (2) **BASE + KEY**, as shown in Figure 2 (b), which takes keyphrase and article as input. (3) **BASE + Filter**, as shown in Figure 2 (c), which takes keyphrase-filtered article as input. (4) **BASE + StackFuse**, (5) **BASE + AddFuse**, and (6) **BASE + ParallelFuse** as shown in Figure 2 (d-f), which take the keyphrase-filtered article and the original article as inputs with stack-fusing, addition-fusing, and parallel-fusing mechanism, respectively. Based on **BASE + StackFuse**, **BASE + AddFuse**, and **BASE + ParallelFuse**, we further use the keyphrase as their additional inputs, like **BASE + KEY**. Then we obtain three additional variants (7) **BASE + StackFuse + KEY**, (8) **BASE + AddFuse + KEY**, and (9) **BASE + ParallelFuse + KEY**. In addition to **BASE**, We also compare four other keyphrase-agnostic baselines as follows. (10) **PT-NET**, the original pointer-generator network (See et al., 2017) , which are widely used in text summarization and headline generation tasks. (11) **SEASS** (Zhou et al., 2017b), the GRU-based (Cho et al., 2014) sequence-to-sequence model with selective encoding mechanism, which is widely used in text summarization. (12) **Transformer + Copy** (Vaswani et al., 2017; Gu et al., 2016), which has the same architecture hyperparameters as **BASE**, the only difference is that it does not use BERT to initialize the encoder. (13) **BASE + Diverse**, which applies diverse decoding (Li et al., 2016b) in beam search to **BASE** during inference to improve the generation diver-

[3] https://github.com/google-research/bert.

Method	ROUGE-1			ROUGE-2			ROUGE-L			Distinct-1		Distinct-2	
	K=1	K=3	K=5	K=1	K=3	K=5	K=1	K=3	K=5	K=3	K=5	K=3	K=5
PT-GEN	35.66	39.82	41.59	19.80	22.60	23.84	32.96	36.73	38.33	0.125	0.076	0.215	0.143
SEASS	31.20	34.52	35.98	14.82	16.52	17.17	28.23	31.04	32.25	0.112	0.069	0.191	0.126
Transformer + Copy	38.91	43.80	45.72	21.85	25.31	26.69	35.32	39.65	41.38	0.110	0.059	0.183	0.111
BASE	42.09	45.40	47.21	24.10	26.70	28.13	38.36	41.33	42.98	0.131	0.074	0.223	0.139
BASE + Diverse	-	45.83	47.89	-	26.62	28.28	-	41.77	43.71	0.182	0.111	0.313	0.213
BASE + Filter	39.44	41.52	43.89	20.82	21.81	23.60	35.30	37.12	39.38	**0.378**	**0.294**	**0.637**	**0.575**
BASE + KEY	43.53	47.07	49.08	25.27	27.81	29.44	39.50	42.76	44.67	0.193	0.121	0.309	0.218
BASE + AddFuse	**44.30**	47.36	49.46	**25.98**	27.39	29.11	**40.24**	42.48	44.47	0.235	0.156	0.385	0.290
BASE + ParallelFuse	43.74	47.28	49.69	25.20	27.56	29.49	39.41	42.50	44.77	0.261	0.177	0.430	0.333
BASE + StackFuse	43.97	47.63	49.74	25.32	27.90	29.69	39.60	42.96	44.97	0.201	0.127	0.332	0.237
BASE + AddFuse + KEY	43.12	46.82	49.16	24.66	27.09	28.91	38.91	42.18	44.37	0.276	0.190	0.447	0.350
BASE + ParallelFuse + KEY	43.09	**47.70**	49.84	24.92	**28.08**	29.82	39.00	43.08	45.12	0.206	0.130	0.337	0.242
BASE + StackFuse + KEY	43.87	47.71	**49.96**	25.50	28.05	**29.94**	39.94	**43.27**	**45.43**	0.242	0.160	0.392	0.293

Table 2: Multi-Headline Generation Results

sity for multiple headlines generation. To sum up, there are a total of 13 models to compare.

Implementation and Hyperparameters. The encoder and the decoder of BASE have the same architecture hyperparameters as BERT-base. All the variants of keyphrase-aware headline generation models also have the same architecture hyperparameters as BASE. The only difference among them is their computation steps of each block in the decoder, as shown in Figure 2. We follow the same training strategy in Vaswani et al. (2017) for training. And the implementation of them are based on Tensor2Tensor[4]. The implementations of PT-NET[5] and SEASS[6] are based on their open-source code.

4.2.1 Multi-Headline Generation

In this experiment, we only give models the news articles without the golden keyphrases. We use SLOT to generate top-K keyphrases for each article. Then each keyphrase-aware generation model using them to generates K different keyphrase-relevant headlines. For keyphrase-agnostic baselines, we apply the beam search to generate top k headlines for each article. We also apply the diverse decoding to BASE as a strong baseline (BASE + Diverse) for further comparison. The diversity penalty is set to be 1.0. It should be noted that we can also apply the diverse decoding to our keyphrase-aware models to further improve diversity.

Metrics. Following Li et al. (2016a), we use *Distinct-1* and *Distinct-2* (the higher the better) to evaluate diversity, which report the degree of the diversity by calculating the number of distinct unigrams and bigrams in generated headlines for each article. Since randomly generated headlines are also highly diverse, we measure the quality as well. As we discussed in Section 1, one news article can have multiple reasonable headlines. However, each article in our test set has only one human written headline, which may only focus on one keyphrase of the news article. We should emphasize that there may be only one generated headline that focuses on the same keyphrase of the human-written headline, while others focus on distinct keyphrases. It is thus not reasonable if we use the same human-written headline as the ground-truth to evaluate all generated headlines. We assume that if the headlines generated by the model are high-quality and diverse, there would be a higher probability that one of the headlines is closer to the single ground-truth. Therefore, we report the highest ROUGE score among the multiple generated headlines for each article. This criterion is similar to top-K errors (He et al., 2016) in image classification tasks. We report the results with K=1, 3, 5.

Results. Table 2 presents the results. For diversity, we can see that all of our keyphrase-aware generation models performs significantly better than other keyphrase-agnostic baselines in both *Distinct-1* and *Distinct-2* metrics for all K. After using the diverse decoding, BASE + Diverse achieves higher diversity. Nevertheless, the diversity is still lower than most keyphrase-aware generation models. As expected, BASE + Filter achieves the highest diversity, and BASE + KEY achieves the lowest diversity among the variants of our keyphrase-aware generation models. For quality, except BASE, there is still a big gap between other keyphrase-agnostic baselines and our keyphrase-aware generation models. Except for BASE + Filter, all of our keyphrase-aware generation models achieve higher ROUGE scores than BASE and BASE + Diverse (see the last 6 lines in Table 2). These results show that our keyphrase-aware generation models can effectively generate high-quality and diverse headlines.

[4] https://github.com/tensorflow/tensor2tensor.
[5] https://github.com/abisee/pointer-generator.
[6] https://github.com/magic282/SEASS.

Method	mAP@1			mAP@3			mAP@5			mAP@10		
	k=1	k=3	k=5	k=1	k=3	k=5	k=1	k=3	k=5	k=1	k=3	k=5
HUMAN	49.25	-	-	63.94	-	-	69.40	-	-	75.70	-	-
PT-GEN	32.08	35.50	37.01	45.23	49.36	50.91	50.51	54.61	56.30	56.77	60.93	62.51
SEASS	28.82	31.66	32.85	42.09	45.27	46.60	47.66	50.59	52.11	54.51	57.74	59.13
Transformer + Copy	37.55	41.88	43.46	51.09	55.99	57.72	56.54	61.45	63.06	62.58	67.21	69.04
BASE	39.43	43.80	45.54	53.77	58.35	60.16	59.29	63.88	65.70	65.51	69.92	71.68
BASE + Diverse	39.30	45.02	47.07	53.69	60.05	62.18	59.10	65.49	67.61	65.37	71.65	73.73
BASE + Filter	34.30	43.02	45.91	48.28	58.07	61.61	53.91	63.78	67.42	60.41	70.51	74.06
BASE + KEY	39.38	45.04	46.81	53.81	60.05	61.91	59.39	65.75	67.59	65.59	71.99	73.97
BASE + AddFuse	39.95	**46.64**	48.72	54.59	**61.90**	63.92	60.20	**67.63**	**69.64**	66.56	**73.95**	75.96
BASE + ParallelFuse	39.82	46.57	**49.03**	54.44	61.80	**64.27**	59.91	67.50	69.85	66.47	73.82	**76.08**
BASE + StackFuse	40.11	45.96	47.95	54.52	61.05	63.05	60.11	66.69	68.70	66.57	72.91	74.86
BASE + AddFuse + KEY	39.19	46.01	48.55	53.73	61.32	63.70	59.34	66.99	69.38	65.70	73.54	75.77
BASE + ParallelFuse + KEY	**40.24**	46.46	48.66	**54.82**	61.64	63.68	**60.46**	67.24	69.30	**67.01**	73.70	75.53
BASE + StackFuse + KEY	39.78	46.33	48.55	54.28	61.43	63.62	59.95	67.07	69.21	66.20	73.35	75.37

Table 3: News Article Retrieval Results

4.2.2 News Article Retrieval

To further evaluate the quality and diversity of the generation, we design an experiment that uses a search engine to help us verify the diversity and quality of the generated headlines. It should be noted that the main purpose of this experiment is not to improve the performance of the search engine, but to measure the quality and diversity of the generated multiple headlines through a real-world search engine. We first collect the data pairs of the news article and its related user search query $\langle X, Q\rangle$ in the following way. If the article X is returned by the search engine based on a user query Q and the user clicks on the article X, then we take the query and the article as a data pair $\langle X, Q\rangle$. After collection, each article X in the test set has 10 different related user queries on average. The article X is used as the ground-truth for Q in the following evaluation. We replace the search key in the original search engine for each article with the K generated multi-headlines. Also, we re-build the indexes of the search engine that contains 10,000 news articles in the test set. Then we re-use the user search queries to retrieve the article. We believe that if the generated multi-headlines have high diversity and quality, then given different user queries, there should be a high probability that the golden article can be retrieved.

Metrics. We use the mean average precision (mAP), which is widely used for information retrieval as a metric. We report the results of mAP@N (N=1, 3, 5, and 10) which test the average probability that the golden article is ranked by the search engine to the top N. Using the human-written headline (HUMAN) as the search key is evaluated as a strong baseline. We also compare the performance of using a different number of headlines (K=1, 3, and 5). It should be noted that increasing the number of headlines as the search key does not ensure the improvement of the mAP, because the number of search keys of all other articles will also be increased. If the generated multi-headlines are not good enough, it will introduce noise and even cause mAP decreasing.

Results. Table 3 presents the results. Similarly, our models perform much better than other keyphrase-agnostic baselines. In most cases, BASE + Diverse outperforms BASE, but still performs worse than 7 keyphrase-aware generation models (see the last 7 lines in Table 3). Generally, with the number of headline K increases, we can see that the performance of our keyphrase-aware generation models improves much higher than other baselines. We find that the mAP@10 of BASE + ParallelFuse (K=5) achieves 76.08, which is even better than HUMAN. These results demonstrate that our keyphrase-aware generation models can generate high-quality and diversity headlines.

4.2.3 Human Evaluation

At a similar level of diversity, we want to investigate the quality of the headlines generated by our keyphrase-aware headline generation model compared to BASE + Diverse. Since the Distinct-1 and Distinct-2 of BASE + Diverse and BASE + StackFuse are close, we compare the quality of them through human evaluation. We randomly sample 100 articles from the test set, let each model generate 3 different headlines. We also mix a random headline and the golden headline for each article, and thus each article has 8 headlines. Three experts are asked to judge whether each headline could be used as the headline of the news. If more than two experts believe that it can be used as the headline of the news, then this headline is considered qualified. The results of the qualified rate of golden, BASE + Stack, BASE + Diverse, and random are 91.8%,

Article#1	the mountain east conference — which says farewell to two members after this season — announced two new members thursday, one full member and one associate member. davis & elkins and the university of north carolina at pembroke will join frostburg state as new conference members. *... [7 sentences with 151 words are abbreviated from here.]* two original conference members will depart after the 2018-19 season. shepherd will move to the pennsylvania state athletic conference and uva-wise will move to the south atlantic conference. member school wheeling jesuit is adding football and will play a full schedule next season. when unc pembroke joins, the conference will have 12 football programs.
Golden	mountain east conference to welcome davis & elkins as full member, unc pembroke as associate member
BASE	mountain east conference announces new conference members mountain east conference announces two new conference members unc pembroke announces new conference members
BASE + Diverse	unc pembroke announces new conference members mountain east conference announces two new conference members unc pembroke announces 2019 - 20 conference members
BASE +AddFuse	**mountain east conference** announces two new conference members [**mountain east conference**] unc pembroke to join **frostburg state** as new members [**frostburg state**] unc pembroke will join the conference in **football** as new **football** conference members in 2020 [**football**]
Article#2	one of the fbi's 10 most wanted was shot and killed during an incident involving apex police and the fbi on wednesday. the fbi and apex police were at woodspring suites, located at 901 lufkin road in apex, after following a tip concerning a fugitive. *... [6 sentences with 129 words are abbreviated from here.]* according to officials carlson posted a bond and fled to mount pleasant in south carolina. he was placed on the fbi's list of top ten fugitives in september 2018. the medical examiner will need to positively identify carlson. "the fbi is grateful to our partners with the apex police department for the assistance," the fbi said.
Golden	suspect on fbi's 10 most wanted list killed in north carolina
BASE	fbi ' s 10 most wanted shot and killed in incident involving apex police fbi ' s 10 most wanted shot , killed in incident involving apex police and fbi fbi ' s 10 most wanted shot and killed in apex incident
BASE + Diverse	fbi ' s 10 most wanted shot , killed in apex incident wpd : fbi ' s most wanted shot , killed in apex incident fbi ' s most wanted shot and killed in apex , fbi says
BASE +AddFuse	one of fbi ' s **10 most wanted** , killed during an incident [**10 wanted**] suspect arrested , killed in incident involving **apex police** [**apex police**] fbi ' s 10 most wanted shot , killed in **south carolina** [**north carolina**]

Figure 3: Examples of original articles, golden headlines and multiple generated outputs by BASE, BASE + Diverse and BASE + AddFuse. Each generated keyphrase is shown at the end of each generated headline.

62.6%, 36.2%, and 0.0%, respectively. These results show that the quality of BASE + StackFuse is also higher than BASE + Diverse. We present some examples for comparison as shown in Figure 3.

5 Related Works

News headline generation is a subtask of summarization which has been extensively studied recently (Rush et al., 2015; Takase et al., 2016; Ayana et al., 2016; Tan et al., 2017; Zhou et al., 2017b; Higurashi et al., 2018; Zhang et al., 2018; Murao et al., 2019). With the rapid development of neural networks, various neural models have been successfully used in the headline generation task. Rush et al. (2015) propose an attention-based neural network for headline generation. Takase et al. (2016) propose a method based on encoder-decoder architecture and design an AMR encoder for headline generation. To take evaluation metrics into consideration, Ayana et al. (2016) apply minimum risk training method to the generation model. Tan et al. (2017) propose a coarse-to-fine method, which first extracts the salient sentences and then generates the headline based on these sentences. Zhou et al. (2017b) propose a method which divides the process of headline generation into three phases: a sentence encoder, a selective gate network for sentence selection, and a headline decoder. Higurashi et al. (2018) propose an extractive headline generation method, different from previous works that target the headline generation for the articles, this work focus on the task of headline generation for the community question answering forums. Due to lack of supervised training data, they propose a learning-to-rank based method to extract the essential substring from a question and use this substring as the headline of the forums. Zhang et al. (2018) propose a method for question headline generation, which designs a dual-attention seq2seq model.

However, most previous headline generation methods focus on one-to-one mapping, and the headline generation process is not controllable. In this work, we focus on the news multi-headline generation problem and design a keyphrase-aware headline generation method. Different information aware methods have been successfully used in natural language generation tasks (Zhou et al., 2017a, 2018; Wang et al., 2017), such as responses generation in the dialogue system. Similar to our task, responses generation in a dialogue system is also a one-to-many problem, Zhou et al. (2017a) propose a mechanism-aware seq2seq model for controllable response generation. They model different mechanisms as latent embeddings and learn the latent

embeddings in their seq2seq model. Incorporating these mechanisms, their model can generate controllable responses. Zhou et al. (2018) propose a commonsense knowledge aware conversation generation method. More concretely, in the first stage, their model retrieves subgraphs from a knowledge base, and the model encodes the subgraphs using a dynamic graph neural network to facilitate better conversation generation in the second stage. Wang et al. (2017) propose an encoder-decoder based neural network for response generation. To our best knowledge, we are the first to consider keyphrase-aware mechanism on news headline generation and build the first keyphrase-aware news headline corpus.

6 Conclusion

In this paper, we demonstrate how to enable news headline generation systems to be aware of keyphrases such that the model can generate diverse news headlines in a controlled manner. We also build a first large-scale keyphrase-aware news headline corpus, which is based on mining the keyphrases of users' interests in news articles with user queries. Moreover, we propose a keyphrase-aware news multi-headline generation model that contains a multi-source Transformer decoder with three variants of attention-based fusing mechanisms. Extensive experiments on the real-world dataset show that our approach can generate high-quality, keyphrase-relevant, and diverse news headlines, which outperforms many strong baselines.

Acknowledgment

This work is supported in part by the National Natural Science Fund for Distinguished Young Scholar under Grant 61625204, and in part by the State Key Program of the National Science Foundation of China under Grant 61836006.

References

Shiqi Shen Ayana, Zhiyuan Liu, and Maosong Sun. 2016. Neural headline generation with minimum risk training. *arXiv preprint arXiv:1604.01904*.

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. 2016. Layer normalization. *arXiv preprint arXiv:1607.06450*.

Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using rnn encoder-decoder for statistical machine translation. In *EMNLP*.

Carlos A Colmenares, Marina Litvak, Amin Mantrach, Fabrizio Silvestri, and Horacio Rodrıguez. 2019. Headline generation as a sequence prediction with conditional random fields. *Multilingual Text Analysis: Challenges, Models, And Approaches*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL*.

Sebastian Gehrmann, Yuntian Deng, and Alexander M Rush. 2018. Bottom-up abstractive summarization. In *EMNLP*.

Jiatao Gu, Zhengdong Lu, Hang Li, and Victor OK Li. 2016. Incorporating copying mechanism in sequence-to-sequence learning. In *ACL*.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *CVPR*.

Tatsuru Higurashi, Hayato Kobayashi, Takeshi Masuyama, and Kazuma Murao. 2018. Extractive headline generation based on learning to rank for community question answering. In *COLING*.

Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. 2016a. A diversity-promoting objective function for neural conversation models. In *NAACL*.

Jiwei Li, Will Monroe, and Dan Jurafsky. 2016b. A simple, fast diverse decoding algorithm for neural generation. *arXiv preprint arXiv:1611.08562*.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*.

Kazuma Murao, Ken Kobayashi, Hayato Kobayashi, Taichi Yatsuka, Takeshi Masuyama, Tatsuru Higurashi, and Yoshimune Tabuchi. 2019. A case study on neural headline generation for editing support. In *NAACL-HLT*.

Alexander M Rush, Sumit Chopra, and Jason Weston. 2015. A neural attention model for abstractive sentence summarization. *arXiv preprint arXiv:1509.00685*.

Abigail See, Peter J Liu, and Christopher D Manning. 2017. Get to the point: Summarization with pointer-generator networks. *arXiv preprint arXiv:1704.04368*.

Minjoon Seo, Aniruddha Kembhavi, Ali Farhadi, and Hannaneh Hajishirzi. 2017. Bidirectional attention flow for machine comprehension. In *ICLR*.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *NIPS*.

Sho Takase, Jun Suzuki, Naoaki Okazaki, Tsutomu Hirao, and Masaaki Nagata. 2016. Neural headline generation on abstract meaning representation. In *EMNLP*.

Jiwei Tan, Xiaojun Wan, and Jianguo Xiao. 2017. From neural sentence summarization to headline generation: A coarse-to-fine approach. In *IJCAI*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *NIPS*.

Jianan Wang, Xin Wang, Fang Li, Zhen Xu, Zhuoran Wang, and Baoxun Wang. 2017. Group linguistic bias aware neural response generation. In *SIGHAN Workshop on Chinese Language Processing*.

Kyle Williams. 2019. Neural lexicons for slot tagging in spoken language understanding. In *NAACL-HLT*.

Qi Zhang, Yang Wang, Yeyun Gong, and Xuanjing Huang. 2016. Keyphrase extraction using deep recurrent neural networks on twitter. In *EMNLP*.

Ruqing Zhang, Jiafeng Guo, Yixing Fan, Yanyan Lan, Jun Xu, Huanhuan Cao, and Xueqi Cheng. 2018. Question headline generation for news articles. In *CIKM*.

Yongzheng Zhang, Evangelos Milios, and Nur Zincir-Heywood. 2007. A comparative study on key phrase extraction methods in automatic web site summarization. In *JDIM*.

Ming Zhong, Pengfei Liu, Danqing Wang, Xipeng Qiu, and Xuanjing Huang. 2019. Searching for effective neural extractive summarization: What works and what's next. In *ACL*.

Ganbin Zhou, Ping Luo, Rongyu Cao, Fen Lin, Bo Chen, and Qing He. 2017a. Mechanism-aware neural machine for dialogue response generation. In *AAAI*.

Hao Zhou, Tom Young, Minlie Huang, Haizhou Zhao, Jingfang Xu, and Xiaoyan Zhu. 2018. Commonsense knowledge aware conversation generation with graph attention. In *IJCAI*.

Qingyu Zhou, Nan Yang, Furu Wei, and Ming Zhou. 2017b. Selective encoding for abstractive sentence summarization. In *ACL*.