

# Generating Representative Headlines for News Stories

Xiaotao Gu<sup>1\*</sup>, Yuning Mao<sup>1</sup>, Jiawei Han<sup>1</sup>, Jialu Liu<sup>2</sup>, Hongkun Yu<sup>2</sup>, You Wu<sup>2</sup>, Cong Yu<sup>2</sup>,  
Daniel Finnie<sup>3</sup>, Jiaqi Zhai<sup>3</sup>, Nicholas Zukoski<sup>3</sup>

<sup>1</sup>University of Illinois at Urbana-Champaign <sup>2</sup>Google Research <sup>3</sup>Google Inc.  
<sup>1</sup>{xiaotao2,yuningm2,hanj}@illinois.edu <sup>2</sup>{jialu,hongkunyu,wuyou,congyu}@google.com  
<sup>3</sup>{danfinnie,jiaqizhai,nzukoski}@google.com

## ABSTRACT

Millions of news articles are published online every day, which can be overwhelming for readers to follow. Grouping articles that are reporting the same event into *news stories* is a common way of assisting readers in their news consumption. However, it remains a challenging research problem to efficiently and effectively generate a representative *headline* for each story. Automatic summarization of a document set has been studied for decades, while few studies have focused on generating representative headlines for a set of articles. Unlike summaries, which aim to capture most information with least redundancy, headlines aim to capture information jointly shared by the story articles in short length and exclude information specific to each individual article.

In this work, we study the problem of generating representative headlines for news stories. We develop a distant supervision approach to train large-scale generation models without any human annotation. The proposed approach centers on two technical components. First, we propose a multi-level pre-training framework that incorporates massive unlabeled corpus with different quality-vs.-quantity balance at different levels. We show that models trained within the multi-level pre-training framework outperform those only trained with human-curated corpus. Second, we propose a novel self-voting-based article attention layer to extract salient information shared by multiple articles. We show that models that incorporate this attention layer are robust to potential noises in news stories and outperform existing baselines on both clean and noisy datasets. We further enhance our model by incorporating human labels, and show that our distant supervision approach significantly reduces the demand on labeled data. Finally, to serve the research community, we publish the first manually curated benchmark dataset on headline generation for news stories, NewSHead, which contains 367K stories (each with 3-5 articles), 6.5 times larger than the current largest multi-document summarization dataset.

## 1 INTRODUCTION

Today’s news consumers are inundated with news content—over two million news articles and blog posts are published everyday<sup>1</sup>. As a result, services that organize news articles have become popular among online users. One approach is to categorize articles into pre-defined *news topics*, each with a short category label, such as “Technology,” “Entertainment,” and “Sports”. While more organized, redundant content can still appear within each topic. Another more effective approach is to further group articles within each topic based on *news stories*. Here, each story consists of a cluster of articles

<sup>1</sup><http://www.marketingprofs.com/articles/2015/27698/2-million-blog-posts-are-written-every-day-heres-how-you-can-stand-out>

\* Work done while interning at Google. Corresponding Author: Jialu Liu.

### Raptors vs. Bucks

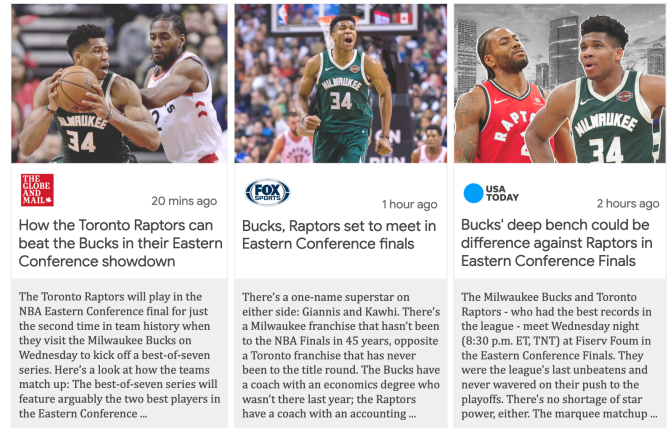


Figure 1: An example of automatically generated story headline for articles about “Raptors vs. Bucks”.

reporting the same event. News stories make it more efficient to complete a reader’s news consumption journey — the reader can move from story to story and dive into each one as desired. However, when we only present a list of article titles, readers can hardly get the gist of a story until they have read through several articles, as article titles are tailored to specific articles and do not provide an overview of the entire story. Also, titles can be too long to scan through, especially on mobile devices.

To tackle this problem, we propose to summarize news stories with succinct and informative headlines. For example, “Raptors vs. Bucks” in Figure 1 headlines the cluster of news articles regarding the game between the two teams. Intuitively, headline is a useful complement to news clusters—users can quickly identify what stories they are planning to read in depth, without skimming a flotilla of unorganized news feeds. In practice, the value of headlines is also affirmed: Google News attaches a headline at the top of its story full coverage page<sup>2</sup>. However, it remains a challenging research problem to automatically generate story headlines. First of all, selecting an existing article title may not be suitable, as article titles can be *too long* (especially for readers on mobile devices) or *unilateral* (incomplete perspective) to cover the general information of the story. Next, relying on human editors to curate high-quality story headlines is inefficient, as it is not only expensive but also hard to scale due to the vast amount of emerging news stories and strict latency requirements.

<sup>2</sup><https://www.blog.google/products/news/new-google-news-ai-meets-human-intelligence/>

To this end, we propose to study the problem of automatically generating representative headlines for news stories. The main idea behind it, known as “document summarization”<sup>3</sup>, has been studied for over six decades [29]. The summarization task is to compress a single article into a concise summary while retaining its major information and reducing information redundancy in the generated summary [12, 33, 61]. As a special task of summarization, *single-document* headline generation has also been thoroughly studied, whose generated summaries are not longer than one sentence [1, 9, 65]. Recently, end-to-end neural generation models have brought encouraging performance for both abstractive summarization [6, 38, 40, 54, 69] and single-document headline generation [16, 19, 38, 53]. Summarization of multiple documents has also gained much attention [2, 4, 11, 18, 26], while headline generation for *a set of documents* remains a challenging research problem.

The main challenge comes from the lack of high-quality training data. Current state-of-the-art summarization models are dominated by neural encoder-decoder models [16, 19, 64, 69]. The high performance of these data-hungry models is supported by massive annotated training samples. For single-document headline generation, such training data can be easily fetched from unlimited news articles with little cost: existing article-title pairs form perfect training samples. Unfortunately, such a mapping is unavailable for multi-document settings. Manually writing the summary or headline of a set of documents is also much more time consuming than that in the context of single-document summarization. Hence, recent multi-document summarization (MDS) models either seek to adapt from single-document models [22, 67] or leverage external resources such as Wikipedia pages [26, 27]. [11] recently provides a crowd-sourced dataset for multi-document summarization, but such resources remain absent for multi-document headline generation.

To facilitate standard study and evaluation, we publish the first dataset for multi-document headline generation. The published dataset consists of 367K news stories with human-curated headlines, 6.5 times larger than the biggest public dataset for multi-document summarization [11]. Large as it may seem, 367K news stories is still a drop in the ocean compared with the entire news corpus on the Web. More importantly, manual curation is slow and expensive, and can hardly scale to web-scale applications with millions of emerging articles every day. To this end, we propose to further leverage the unlabeled news corpus in two ways. Existing articles are first treated as a knowledge base and we automatically annotate unseen news stories by distant supervision (*i.e.*, with one of the article titles in the news story). We then propose a multi-level pre-training framework, which initializes the model with a language model learned from the raw news corpus, and transfers knowledge from single-document article-title pairs. The distant supervision framework enables us to generate another dataset for training without human effort, which is 6 times larger than the aforementioned human-curated dataset. We show that our model solely based on distant supervision can already outperform the same model trained on the human-curated dataset. In addition, fine-tuning the distantly-trained model with a small number of human-labeled examples further boosts its performance (Section 7.3). In real-world applications, the process of grouping

<sup>3</sup>Article and document are used interchangeably throughout this paper. We use the former in the context of news and the latter for document summarization.

**Table 1: Statistics comparison with MDS datasets.**

Dataset	# Instances	# Characters in target
NewSHead	367,099	27.76
NewSHead <sub>dist</sub>	2,204,224	38.37
TAC2011 [42]	176	642.36
DUC03+04 [44]	320	675.43
Multi-News [11]	56,216	1298.62

news stories, which is viewed as a prerequisite, is not always perfect. To tackle this problem, we design a self-voting-based document-level attention model, which proves to be robust to the noisy articles in the news stories (Section 7.4). Improving the quality of clustering is out of the scope of this work, but remains an interesting future direction.

Our contributions are summarized as follows:

- (1) We propose the task of headline generation for news stories, and publish the first large-scale human-curated dataset to serve the research community<sup>4</sup>;
- (2) We propose a distant supervision approach with a multi-level pre-training framework to train large-scale generation models without any human annotation. The framework can be further enhanced by incorporating human labels, but significantly reduces the demand of labels;
- (3) We develop a novel self-voting-based article attention module, which can effectively extract salient information jointly shared by different articles, and is robust to noises in the input news stories.

## 2 PROBLEM DESCRIPTION

Given a news story  $\mathcal{A}$  as a cluster of news articles regarding the same event, where each article  $\mathbf{a} \in \mathcal{A}$  is composed of a token sequence  $[a^1, a^2, \dots]$ , we aim to generate a succinct and informative headline of the story, represented as another token sequence  $\mathbf{y} = [y^1, y^2, \dots]$ , such as “Raptors”:  $y^1$ , “vs.”:  $y^2$  and “Bucks”:  $y^3$  in Figure 1 for a list of articles discussing about the series between the two teams.

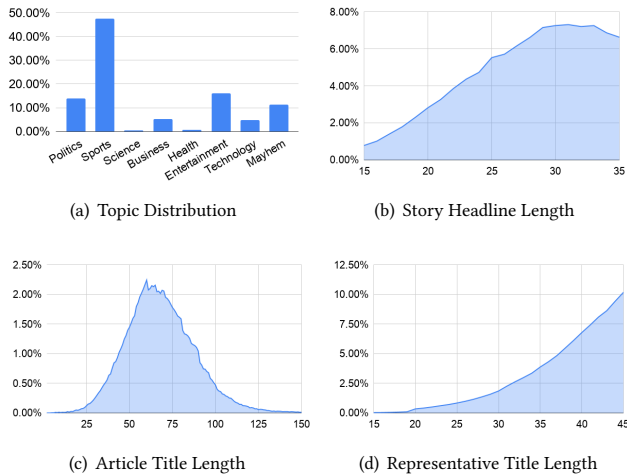
Although the original title of each article can be a strong signal for headline generation, it is not included in our model input as (1) it increases the risk of generating clickbaity and biased headlines, and (2) high-quality titles can be missing in some scenarios (*e.g.*, user-generated content). For these reasons, we only consider the main passage of each article as model input at this stage.

## 3 THE NEWSHEAD DATASET

To help future research and evaluation, we publish, to the best of our knowledge, the first expert-annotated dataset, NewSHead, for the task of **News Story Headline** Generation. The NewSHead dataset is collected from news stories published between May 2018 and May 2019. NewSHead includes the following topics: Politics, Sports, Science, Business, Health, Entertainment, and Technology as depicted in Figure 2(a). A proprietary clustering algorithm iteratively loads articles published in a recent time window and groups them based on content similarity. For each news story, curators

<sup>4</sup>Data: <https://github.com/google-research-datasets/NewSHead>

<sup>5</sup>Code: <https://github.com/tensorflow/models/tree/master/official/nlp/nhnet>

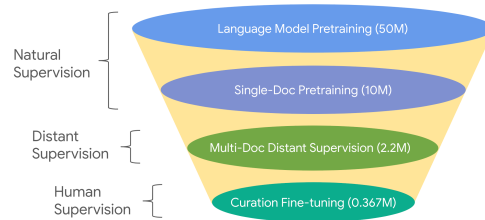


**Figure 2: Visualization of data statistics: (a) topic distribution of articles in NewSHead; (b) length distribution of manually curated story headlines (in character); (c) length distribution of the original article titles; (d) length distribution of selected representative titles in NewSHead<sub>dist</sub>.**

from a crowd-sourcing platform are requested to provide a headline of up to 35 characters to describe the major information covered by the articles in each story. The curated headlines are then validated by other curators before they are included in the final dataset. Note that a story may contain hundreds of articles, and it is not realistic to ask curators to read through all the articles before curating a headline. Thus, only three to five representative articles that are close to the cluster centroid are picked to save human efforts.

Table 1 shows the statistics of our dataset and the existing datasets for multi-document summarization (MDS). In NewSHead, each news story consists of 3-5 news articles. This gives us 367K data instances, which is 6.5 times larger than the biggest dataset for multi-document summarization [11]. We split the dataset by timestamps: the timestamps of all articles in the validation set are strictly greater than those in the training set. The same goes for the test set vs. validation set. By avoiding overlapped time window, we can penalize overfitted models that memorize observed labels. Overall, we generate 357K stories for training, 5K stories for validation, and 5K stories for testing, respectively. As for the human-curated reference labels, as Table 1 shows, the lengths of curated story headlines are much shorter than traditional summaries, and even shorter than article titles in our dataset depicted in Figure 2(c). Figure 1 shows an example of a curated news story. The story headline is much more concise than article titles in the cluster, and covers only general information shared by the articles.

Large as it may seem, the training data in NewSHead remains a drop in the ocean compared with the entire news corpus, still leaving substantial room for performance improvement, under the assumption that modern models can achieve better performance with more data [28, 49]. Nevertheless, manual annotation is slow and expensive. The amount of work and resource needed to create the NewSHead dataset is already so much that scaling it up



**Figure 3: Multi-level supervision used in our framework.**

to just 500K instances seems to be cost-prohibitive. Facing this practical challenge, in the next section, we will present a novel framework based on distant supervision to fetch additional training data without human annotation.

## 4 LEARNING FRAMEWORK

Learning an end-to-end generation model requires a large amount of annotated training data. Human annotation, however, is slow and expensive. It is thus hard for human annotation alone to provide sufficient data or scale to future scenarios. In the following sections, we present a novel framework that leverages multiple levels of supervision signals to alleviate or even remove the dependency on human annotation. As shown in Figure 3, we seek natural supervision signals in the existing news corpus to pre-train the representation learning and language generation module of our framework. This process includes *Language Model Pre-training* from massive text corpus, and transferring knowledge from article-title pairs (*Single-Doc Pre-training*). We then propose to generate heuristic training labels via *Multi-Doc Distant Supervision*. These supervision signals can be automatically fetched from the existing data with almost no cost. Later in Section 7, we show that models trained from these free signals can outperform those purely trained on the manually curated training set.

### 4.1 Distant Supervision: NewSHead<sub>dist</sub>

In this section, we show how to generate abundant training data from the existing corpus without human story curators. A related technique, namely distant supervision, has proven to be effective in various information extraction tasks (e.g., Relation Extraction [35] and Entity Typing [51]). The basic idea is to use existing *task-agnostic labels* in a knowledge base (KB) to heuristically generate imperfect labels for specific tasks. For example, in Entity Typing, one can match each entity mention (e.g., *Donald Trump*) in the sentence to an entry in the KB, and then label it with existing types of the entry (e.g., *Politician*, *Human*, etc). In this way, by leveraging existing labels and some heuristics, there is no need to spend extra time to curate labels for the specific task of interest.

Here, we view the news corpus as a KB, and treat existing article titles as candidate labels for news stories. Note that not all article titles are suitable as story headlines, since, as we mentioned, some titles are too specific to cover the major information of the story. Hence we need to automatically select high-quality story labels from many candidates without annotations from human experts. Specifically, given a news corpus, we first group news articles into news stories. This is an unsupervised process—the same as what we

**Table 2: A comparison between the headline label generated by human curators in NewSHead and the one generated by Distant Supervision in NewSHead<sub>dist</sub>.**

<b>NewSHead:</b> alabama senate abortion bill
<b>NewSHead<sub>dist</sub>:</b> alabama senate approves near-total abortion ban
<b>Title<sub>1</sub>:</b> Alabama state Senate passes near total abortion ban in direct challenge to Roe v. Wade The Senate rejected an attempt to add an exception for rape and incest, voting down the amendment 21-11. Alabama’s Republican-dominated state Senate has passed a bill to ban nearly ...
<b>Title<sub>2</sub>:</b> Alabama Senate approves near-total abortion ban The Alabama state Senate on Tuesday approved a bill essentially banning abortion in the state, a move specifically aimed at challenging more than 40 years...
<b>Title<sub>3</sub>:</b> Alabama Senate bans nearly all abortions, including rape cases (Reuters) - Alabama’s state Senate passed a bill on Tuesday to outlaw nearly all abortions, creating exceptions only to protect the mother’s health, as part of a multistate effort to have the ...
<b>Title<sub>4</sub>:</b> Alabama state Democrat says near-total abortion bill passage ‘raped women last night’ (CNN) A Democratic state senator from Alabama equated a state bill that would ban abortion with no exceptions for rape and incest to the rape of Alabama women...
<b>Title<sub>5</sub>:</b> 25 men voted to advance most restrictive abortion ban in the country. The female governor signed it Washington (CNN) Only male Alabama senators voted Tuesday to pass the most restrictive abortion bill in the country, which criminalizes abortion in the state and bans the procedure in nearly all cases including rape and incest...

did for creating the NewSHead dataset. For each story  $\mathcal{A}$ , we aim to get its heuristic headline  $\hat{y}$  by selecting the most representative article title:

$$\hat{y} = \operatorname{argmax}_{\mathcal{T}_a: a \in \mathcal{A}} \frac{1}{|\mathcal{A}|} \sum_{a' \in \mathcal{A} \setminus \{a\}} f(\mathcal{T}_a, a'), \quad (1)$$

where  $\mathcal{T}_a$  stands for the title of article  $a$  and  $f(\mathcal{T}, a)$  stands for the semantic matching score between any title  $\mathcal{T}$  and article  $a$ . Note that  $a$  only contains tokens in the main passage. In other words, the score of an article title is the average matching score between the title and other articles in the story.

The only problem now is to compute the matching score  $f(\mathcal{T}, a)$ . Instead of defining a heuristic score (e.g., lexical overlaps), we train a scorer with a binary classification task, where  $f(\mathcal{T}, a)$  is the probability that  $\mathcal{T}$  can be used to describe  $a$ . Training data can be fetched by using existing article-title pairs as positive instances and sampling random pairs as negative instances. We use a BERT-pre-trained Transformer model for this classification task with a cross-entropy loss.

We then follow Equation 1 to generate heuristic labels for unlabeled news stories. It is likely that none of the article titles in the story is representative enough to be the story headline. Hence, we only include positive labels<sup>6</sup> in the generated training data (leaving around 20% of the stories). The length distribution of the generated labels is shown in Figure 2(d). The average length is longer than human labels, but is in a reasonable area to generate enough training instances. This way, we generate around 2.2M labeled news stories without relying on human curators. The new dataset, namely NewSHead<sub>dist</sub>, is 6x larger than the annotated NewSHead dataset. Table 2 shows an example of heuristically generated labels in NewSHead<sub>dist</sub>. Among all five candidate titles, the second title is ranked as the top choice since it well describes the general information of the story. In comparison, the last title is not suitable as it is too specific and does not match some of the articles.

<sup>6</sup>A label is positive if its average prediction score among all articles is above 0.5.

This way we can easily generate abundant high-quality training labels. This generation process does not depend on human curators. NewSHead<sub>dist</sub> is hence easy to scale as time goes by and the size of the news corpus gets larger.

## 4.2 Pre-training With Natural Supervision

With the distantly supervised dataset comes a natural question: how far can we progress on the story headline generation task without human annotation? To make the most of the massive unlabeled corpus, we apply pre-training techniques to enhance our model. As an overview, our model includes an encoder-decoder module as its building block, and an article-level attention layer to integrate information from different articles in the story. The detailed model architecture will be presented in Section 5.1. At the pre-training stage, aside from *Multi-Doc Distant Supervision*, we aim to initialize different modules with two kinds of *Natural Supervision* signals from the existing news corpus as follows.

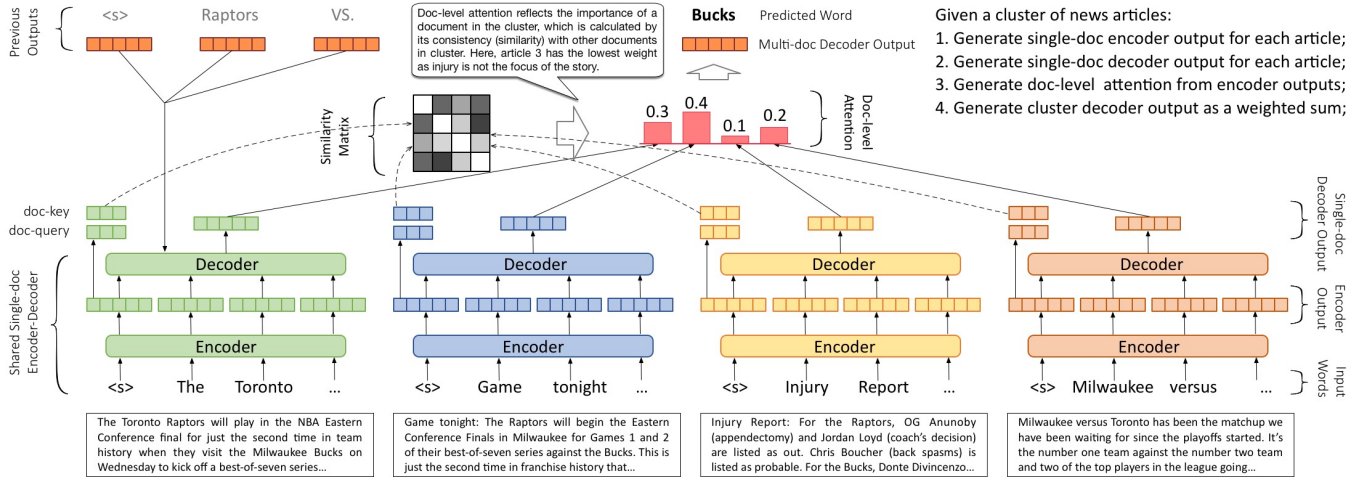
**Language Model Pre-training** transfers knowledge from the massive raw corpus in the news domain to enhance the representation learning module of our model. We followed the BERT pre-training paradigm [8] to construct a dataset based on the main passages of over 50M news articles collected from the Web<sup>7</sup>. This dataset consists of 1.3 billion sentences in total. Both tasks of masked language model and next sentence prediction are included. The learned parameters are used to initialize the encoder module and the word embedding layer. The decoder module can also be initialized with encoder parameters as suggested in literature [52]. However, cross attention between the encoder and decoder modules remains uninitialized in this phase.

**Single-Doc Pre-training** leverages abundant existing article-title pairs to train the encoder-decoder module to enhance both representation learning and language generation. In this step, we further adjust parameters for the encoder and decoder modules, together with cross attentions between them. We clean the 50M raw news articles and leave only 10M high-quality article-title pairs for training. For data cleaning, we first filter out article titles that are either too short (<15 characters) or too long (>65 characters). We then use additional classifiers to remove article titles that are clickbait, offensive or opinion-like. These additional classifiers are trained by binary labels collected from crowd-sourcing platforms. Note that this filtering step is not mandatory in the framework.

The model is further trained on distant supervision data with weights initialized from previous stages. The previous two pre-training stages are used to initialize the encoder and decoder modules in the single document setting. When it comes to multiple documents, where the model involves additional document attentions, we train such parameters together with previously mentioned model components.

Experiments show that models trained with above cost-free signals can even outperform models trained on manually curated training data. By fine-tuning the model on human-curated labels, we can combine the two sources of supervision and further improve performance.

<sup>7</sup>One can use CommonCrawl to fetch web articles.



**Figure 4: The overall architecture of our model NNet. We use a shared encoder-decoder module to generate representation for each individual article, then integrate the results via an article-level self-voting attention layer (Section 5.1).**

## 5 THE NHNET MODEL

In this section, we elaborate on the mathematical mechanism of our multi-doc news headline generation model, namely NNet. We extend a standard Transformer-based encoder-decoder model to multi-doc setting and propose to use an article-level attention layer in order to capture information common to most (if not all) input articles, and provide robustness against potential outliers in the input due to clustering quality. We analyze the model complexity compared to the standard Transformer model.

### 5.1 Model Architecture

Figure 4 illustrates the basic architecture of our generation model. To take full advantage of massive unlabeled data, we start from a Transformer-based single-document encoder-decoder model as the building block. The single-doc model generates decoding output for each article in the cluster individually. To effectively extract common information from different articles, the model fuses decoding outputs from all articles via a self-voting-based article-level attention layer. The framework proves to be not only easy to pre-train with distant supervision, but also robust to potential noises in the clustering process.

We start from a standard Transformer-based encoder-decoder model [59] as the building block. In the single-doc setting, the sequence of tokens from an input article  $a$  passes through a standard  $L$ -layer Transformer unit, with  $h$  attention heads. At decoding step  $i$ , the model takes in the full input sequence along with the output sequence produced up to step  $i - 1$  (i.e.,  $[y^1, y^2, \dots, y^{i-1}]$ ), and yields a  $d_H$ -dimensional hidden vector (let it be denoted by  $H_a^i$ ). The end-to-end single-doc architecture would end up predicting the next token  $y^i$  in the output sequence from  $H_a^i$  as follows.

$$P(y^i = y | [y^1, y^2, \dots, y^{i-1}]) = \frac{\exp(H_a^i \cdot M_y)}{\sum_{y' \in \mathcal{V}} \exp(H_a^i \cdot M_{y'})} \quad (2)$$

Equation 2 above defines the probability of  $y^i$  being  $y$  given  $[y^1, y^2, \dots, y^{i-1}]$ .  $\mathcal{V}$  is the entire vocabulary,  $M_y$  is the column of

a learnable embedding matrix  $M \in \mathbb{R}^{d_H \times |\mathcal{V}|}$ , in correspondence to token  $y$ . Beam search is applied to find top- $k$  output sequences that maximize

$$\prod_{i=1}^L P(y^i | [y^1, y^2, \dots, y^{i-1}]).$$

Extending to the multi-doc setting, we let each input article pass through the same Transformer unit and independently yield  $H_a^i$ , i.e., apply the single-doc setting on all input articles up to the point of hidden vector representation. Then we compute the vector representation of the input article group  $\mathcal{A}$  as the weighted sum of  $\{H_a^i\}$ , i.e.,  $H_{\mathcal{A}}^i = \sum_{a \in \mathcal{A}} w_a H_a^i$ . The weights  $\{w_a\}$  are determined by a similarity matrix learned via article-level attention (detailed next in Section 5.2). Finally, for predicting the next token  $y^i$ , we use  $H_{\mathcal{A}}^i$  in place of  $H_a^i$  in Equation 2.

As we will show in Section 5.2, the article-level attention introduces significantly fewer parameters to learn in addition to those in a standard Transformer model.

### 5.2 Voting-based Article-level Attention

The article-level attention layer is used to integrate information from all articles. It assigns different attention weights to articles, which indicate the importance of each article. To achieve this, previous works [67] use a learnable external query vector, denoted as the *referee query vector*  $Q_r \in \mathbb{R}^{d_H}$ , to individually decide the weight of each article. Specifically, the attention weight of article  $a$  is computed as

$$w_a = \frac{\exp(Q_r^T \cdot K_a)}{\sum_{a' \in \mathcal{A}} \exp(Q_r^T \cdot K_{a'})}, \quad (3)$$

where  $K_a$  is the key vector of article  $a \in \mathcal{A}$ , which is usually linearly transformed from its encoded representations. Such a design is intuitive, but ignores the interaction between articles. For headline generation, our goal is to capture the common information shared by most articles in the story, where inter-article connections are important to consider. More importantly, the clustered news story itself may not be perfectly clean: some articles may be loosely related, or even unrelated to other articles in the story. In these

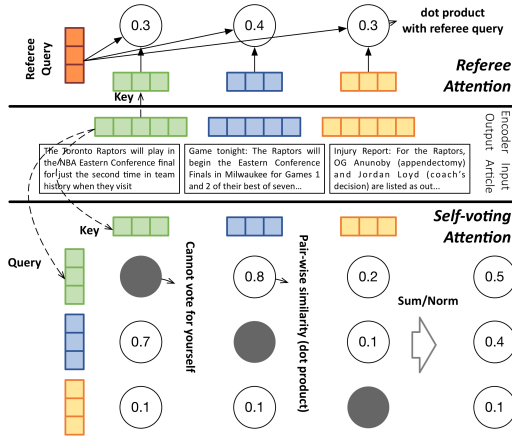


Figure 5: Comparison between the referee attention and the proposed self-voting attention (Section 5.2).

scenarios, attention scores over articles should be determined by all articles together, instead of relying on an external referee.

To this end, we design a simple yet effective *self-voting-based article attention layer*. The basic idea is to let each article vote for other articles in the story. Articles with higher total votes from other articles shall have higher attention scores. The advantages are two folds: common information shared by most articles will be echoed and amplified as it gets more votes, while irrelevant articles will be downplayed in this interactive process to reduce harmful interference on the final output. Specifically, given the representation vector  $H_a$  of article  $a$ , we calculate its query vector and key vector as  $Q_a = W_Q \cdot H_a$  and  $K_a = W_K \cdot H_a$ , respectively, where  $W_Q$  and  $W_K$  are learnable matrices shared by all articles. The attention score of  $a$  is then computed as

$$w_a = \frac{\exp(\sum_{a' \in \mathcal{A} \setminus \{a\}} Q_{a'}^T \cdot K_a)}{\sum_{a'' \in \mathcal{A}} \exp(\sum_{a' \in \mathcal{A} \setminus \{a''\}} Q_{a'}^T \cdot K_{a''})}, \quad (4)$$

where  $\exp(Q_{a'}^T \cdot K_a)$  represents the vote of  $a'$  on  $a$ . Figure 5 illustrates the similarities and differences between the referee attention and self-voting attention. Among the three example articles, the first two are introducing the game between the two teams, while the third one is focusing on injury information, which is too specific to be included in the headline. Through the self-voting process, the article group finds that the third article is more distant from the central topic, and hence downplays its weight. As a result, the generated headline focuses more on the game between the Raptors and Bucks instead of injury information. The referee attention can hardly achieve the same goal as it ignores other articles when assigning the weight to each individual article. As one can expect, the self-voting attention module is also more robust to potential noises in the cluster. In the sanity checking experiments, the attention module usually gives weights close to zero for intentionally added noisy articles, demonstrating its capability in detecting off-topic articles, whereas the referee attention can hardly sense the differences.

*Model complexity.* The standard Transformer model consists of  $O(L \cdot d_H \cdot h \cdot d_v)$  parameters, where  $d_v$  is the dimension of projected value vector space, and  $L, d_H, h$  are as defined in Section 5.1. Adding article-level attention introduces an additional  $O(d_H \cdot h \cdot d_v)$  parameters— $1/L$  of what is already required by Transformer.

## 6 EXPERIMENTAL SETTINGS

For standard evaluation, we compare all methods on the test set of NewSHead, and tune parameters on the validation set.

As headline generation for news stories is a new task, instead of a “state-of-the-art” race between various models, we are more curious about the following questions regarding realistic applications:

- (1) *How far* can we go without any human annotation but using existing natural supervision signals only?
- (2) *Can we* further boost the performance of distantly supervised models by incorporating human labels?
- (3) *How does* the performance change with more human labels?
- (4) *How robust* are these methods to potential noises in news stories, as the story clustering process can be imperfect?

### 6.1 Baseline Methods

Although no previous methods follow exactly the same setting of our task, variants of methods for multi-document summarization (MDS) can serve as our baseline methods with minor adjustments. Specifically, we consider two families of models.

**Extractive Methods.** Extractive MDS models cannot be directly applied as they generate the summary by selecting sentences from the document set, while our expected output is a concise headline. Extracting suitable words from article bodies is also challenging. Here we consider two competitive baseline methods that (cheat-ingly) extract information from article titles:

- **LCS** extracts the longest common (word) sequence of article titles in the story. In case that no common word sequence exists, we relax the constraint of finding a common sequence shared by at least two articles.
- **RepTitles** uses the title scorer that we introduced in Section 4.1 for distant supervision, to select the most representative article title in the story as the predicted headline.

Note that the article titles are unavailable for abstractive models including our model.

**Abstractive Methods.** Abstractive models take article bodies as input and generate the story headline in an end-to-end manner. Since our full model makes use of different kinds of additional natural supervision (e.g., unlabeled corpus and article-title pairs), it would be unfair to compare with models that are not designed to use such signals. For illustration, we compare with both such traditional models (e.g., WikiSum [26]) and models designed to leverage additional supervision for pre-training (SinABS [67]). Additional enhancements are applied to baseline models to make them even stronger and more comparable to our model. All methods are pre-trained with the same resources.

- **WikiSum** [26], as a representation of supervised abstractive models, generates summaries from the concatenation of the ordered paragraph list in the original articles. It proposes a

**Table 3: Performance comparison of different methods. H stands for training with human-curated labels. L/S/D denotes Language model pre-training, Single-document pre-training and Distant supervision pre-training respectively.**

Method	R1-P	R1-R	R1-F	R2-P	R2-R	R2-F	RL-P	RL-R	RL-F	Len-C	Len-W
<b>LCS</b>	0.437	0.646	0.486	0.241	0.385	0.272	0.413	0.620	0.462	0.796	0.788
<b>RepTitles</b>	0.664	0.364	0.458	0.340	0.172	0.220	0.598	0.328	0.413	1.984	1.933
<b>WikiSum (H)</b>	0.561	0.591	0.567	0.312	0.326	0.313	0.509	0.535	0.514	0.958	0.966
<b>SinABS (H+LS)</b>	0.744	0.748	0.738	0.499	0.501	0.493	0.680	0.682	0.674	1.021	1.018
<b>Concat (H+LSD)</b>	0.752	0.756	0.746	0.510	0.510	0.503	0.689	0.694	0.685	1.017	1.013
<b>SinABS (H+LSD)</b>	0.758	0.769	0.755	0.522	0.530	0.518	0.695	0.704	0.692	1.004	1.006
<b>NHNet (H)</b>	0.596	0.621	0.600	0.327	0.338	0.327	0.539	0.560	0.542	0.983	0.980
<b>NHNet (LSD)</b>	0.726	0.590	0.639	0.440	0.354	0.382	0.667	0.542	0.588	1.327	1.286
<b>NHNet (H+L)</b>	0.716	0.728	0.714	0.466	0.471	0.462	0.657	0.668	0.656	1.003	1.000
<b>NHNet (H+LS)</b>	0.751	0.776	0.755	0.514	0.530	0.514	0.688	0.710	0.691	0.992	0.988
<b>NHNet (H+LSD)</b>	<b>0.762</b>	<b>0.779</b>	<b>0.762</b>	<b>0.531</b>	<b>0.542</b>	<b>0.529</b>	<b>0.703</b>	<b>0.718</b>	<b>0.703</b>	1.003	0.997

decoder-only module with memory-compressed attention for the abstractive stage.

- **Concat** first concatenates body texts in all articles of a story, and then uses the single-document encoder-decoder building block of our model to generate the headline. This way, every attention layer in Transformer is able to access tokens in the entire story. To avoid losing word position information after concatenation, the positional encoding is reset at the first token for each article, so the model can still identify important leading sentences of each article.
- **SinABS** [67] is a recently proposed model which adapts and outperforms the state-of-the-art MDS models by pre-training on single-document summarization tasks. It uses a referee attention module to integrate encoding outputs from different articles as the representation for the article set. For fair comparison, we replace the original LSTM encoder with the transformer architecture, with the same parameter size as our model.
- **SinABS (enhanced)**: The original model only makes use of knowledge from single articles. We further enhance it with distant supervision (H+LSD in Table 3).
- **NHNet** is the model we proposed in Section 5.

We test these baseline methods with different training settings, and report detailed performance comparison and analysis in Section 7.

## 6.2 Datasets

As introduced in the multi-level training framework in Sections 3 and 4, the datasets used in this work include

- (1) Language model Pre-training (**L**) with 50M articles;
- (2) Single-Doc Pre-training (**S**) with 10M articles;
- (3) Distant Supervision NewSHead<sub>dist</sub> (**D**) with 2.2M stories;
- (4) Human annotations from NewSHead (**H**) with 357k stories.

The entire NewSHead contains 367k instances, among which we use 5k for validation and 5k for final testing.

The labels for training and testing in this work are uncased, since both human-labeled and distantly supervised headline labels may contain various case formats, which can influence the learning and evaluation process. In application, the task of recovering case information from generated uncased headlines, which is also called true casing in natural language processing, is treated as a separate

task. Meanwhile, we discover that simple majority voting from cased frequent n-grams in the story content is an accurate solution.

## 6.3 Reproduction Details

We use the WordPiece tool [62] for tokenization. The vocabulary size is set to 50k and is derived from uncased news corpus. Each sentence in the news article is tokenized into subwords, which significantly alleviates out-of-vocabulary (OOV) issues. We also tried to incorporate copy mechanism [17], another popular choice to reduce OOV, but did not see significant improvement. For better efficiency and less memory consumption, we use up to 200 WordPiece tokens per article as input.

As for the Transformer model, we adopt a standard ( $L=$ )12-layer architecture, with ( $h=$ )16 heads, hidden states of ( $d_H=$ )768 dimensions, and projected value space of ( $d_v=$ )48 dimensions. For training, we use the Adam [21] optimizer with a 0.05 learning rate and a batch size of 1024. For every model we use the same early stopping strategy to alleviate overfitting: we first let the model train for 10k steps, after that we stop the training process once the model cannot steadily achieve higher performance on the validation dataset.

We implement the model in Tensorflow and train the model on cloud TPUs. Our code will be released together with the NewSHead dataset.

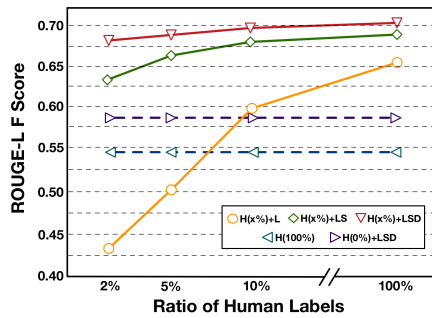
## 6.4 Evaluation Metrics

For evaluation we use the open source scoring tool<sup>8</sup> that aggregates scores from a bootstrapping process. We report *average* results on the 5k NewSHead test set. We use the following metrics to evaluate the generated headlines:

**ROUGE** measures n-gram overlap between predicted headlines and gold labels. Here we report the R-1, R-2, R-L scores in terms of (p)recision ( $\frac{\#overlap}{\#predicted}$ ), (r)ecall ( $\frac{\#overlap}{\#gold}$ ) and F1 ( $\frac{2 \cdot p \cdot r}{p+r}$ ).

**Relative Length** measures the ratio between the length of predicted headlines and the gold labels ( $\frac{Len_{Predict}}{Len_{Gold}}$ ). Here we report the ratio in terms of both words ( $Len-W$ ) and characters ( $Len-C$ ). The closer the relative length is to 1.0, the more likely it is that the generated headline has a similar length to the gold headline, whose distribution is shown in Figure 2(b).

<sup>8</sup><https://github.com/google-research/google-research/blob/master/rouge/scoring.py>



**Figure 6: An illustration of how performance changes with more manual labels involved for fine-tuning. X-axis stands for the ratio of training data used. Y-axis stands for the RL-F score on the test set.**

## 7 EXPERIMENT RESULTS

In the following, we answer the questions raised in the beginning of the former section.

### 7.1 Performance Comparison

Table 3 shows the performance of compared methods trained with different combinations of datasets. Generally, abstractive methods outperform extractive methods, even though the latter have access to already summarized information from existing article titles. Among abstractive methods, the carefully designed concatenation model can achieve comparable performance with the existing state-of-the-art. When enhanced with the distantly supervised training data (H+LSD), the SinABS model can have stronger performance. Our model, when trained with the same resources, consistently outperforms baseline methods.

### 7.2 How effective is distant supervision?

To investigate the necessity of manual curation for this task, we compare the fully supervised model with human annotations only and the distantly supervised model with all natural supervision signals (*i.e.*, Language Model Pre-training and Single-Document Pre-training). To our surprise, the distantly supervised model outperforms the fully supervised model by a significant margin, even though the distantly supervised training labels have different styles and lengths compared to those in the final test set. The result is encouraging by revealing an effort-light approach for this task. Instead of relying on human experts to curate story headlines for over a year, we can automatically mine high-quality headlines and natural supervision signals from existing news data for training in one day. This observation can serve as a solid foundation for the future development of large-scale production models. Fine-tuning the learned model with human labels can further boost performance. As an ablation study, we investigate the model fine-tuned with full human annotation under different pre-training settings. Starting from the fully supervised model with only human annotations (H), every pre-training process (+L+S+D) brings considerable improvements as we expect.

### 7.3 How many manual labels do we need?

Our model achieves the best performance when combining human annotation with all kinds of distant and natural supervision (H(100%)+LSD). Since the distantly supervised model may still need some manual labels to adjust the style and length of its generated headlines, it is hence meaningful to investigate the trade-off between the number of manual labels and the final performance to reasonably save human efforts.

Figure 6 shows how test performance changes as we use different ratios of manual labels to fine-tune the distantly supervised model. Generally, more manual labels lead to better test performance, as one can expect. However, different models have various demand for the human labels to achieve the same performance. As shown, when only 2% of human labels are available, the supervised model, even with Language Model Pre-training (H(2%)+L), achieves significantly worse performance than the model trained with 100% human labels. Our model with distant supervision (H(2%)+LSD), on the contrary, outperforms the fully supervised model by a large margin. More labels beyond this amount only bring slight improvements. This further validates our idea: through distant supervision, we are able to learn high-quality models with very few manual labels.

### 7.4 Are attention modules robust to noises?

So far we have mostly considered models in ideal settings, where the news stories for training and testing are relatively clean and validated. In real-world scenarios, however, automatically clustered news stories can be noisy, *i.e.*, they may include some irrelevant or even off-topic articles. This is when the article-level attention module plays its role by assigning different weights to articles.

In this experiment, we intentionally add noises to the training and testing stories (both for NewSHead and NewSHead<sub>dist</sub>) by randomly replacing an article in each story with a randomly sampled article from the entire corpus. Under the same architecture, we compare three different article-level attention designs: (1) Uniform Attention, which assigns equal weight to each article; (2) Referee Attention, which determines article weights by an external “referee” vector, as introduced in Section 5.2 and Figure 5; (3) Our self-voting-based Attention. We compare their performances under all pre-training settings.

Table 4 shows the performance of different attention designs. Among all attention modules, Referee Attention achieves the worst performance, which is as expected since Referee Attention can assign inappropriately large weights to noisy articles, severely corrupting the final output. This would be further verified with real examples in Section 8. For comparison, the simple Uniform Attention module will at least avoid focusing on the wrong articles, and hence achieves better performance than Referee. From the model perspective, the Referee model is more complex and is harder to train with limited resources. This experiment also shows that, when on-topic articles dominate the story, even simple uniform attention can achieve satisfactory performance by integrating decoding outputs from different articles, but the traditional Referee Attention can produce risky results. Our Self-voting-based Attention achieves best performances under all settings, thanks to its capability of leveraging the dynamic voting process between articles to emphasize shared common information and identify noises. When tested



**Table 4: Robustness Comparison of Different Attention Designs on the Noisy Dataset.**

Method	R1-P	R1-R	R1-F	R2-P	R2-R	R2-F	RL-P	RL-R	RL-F	Len-C	Len-W
<b>Uniform</b> (H)	0.544	0.560	0.543	0.284	0.290	0.281	0.492	0.505	0.491	0.994	1.003
<b>Uniform</b> (H+L)	0.680	0.701	0.683	0.428	0.443	0.429	0.626	0.646	0.629	0.986	0.990
<b>Uniform</b> (H+LS)	0.736	0.747	0.733	0.494	0.500	0.490	0.672	0.681	0.669	1.011	1.011
<b>Uniform</b> (H+LSD)	0.742	0.759	0.743	0.507	0.519	0.506	0.682	<b>0.697</b>	0.682	0.999	0.998
<b>Referee</b> (H)	0.299	0.308	0.297	0.118	0.120	0.115	0.272	0.281	0.270	0.995	1.085
<b>Referee</b> (H+L)	0.695	0.673	0.675	0.434	0.418	0.418	0.638	0.617	0.620	0.993	0.989
<b>Referee</b> (H+LS)	0.737	0.719	0.719	0.484	0.472	0.470	0.672	0.655	0.655	0.999	0.996
<b>Referee</b> (H+LSD)	0.739	0.750	0.737	0.496	0.503	0.492	0.675	0.684	0.673	1.009	1.010
<b>NHNet</b> (H)	0.565	0.594	0.571	0.307	0.320	0.307	0.512	0.538	0.517	0.972	0.972
<b>NHNet</b> (H+L)	0.692	0.707	0.692	0.434	0.444	0.432	0.631	0.645	0.631	0.996	1.000
<b>NHNet</b> (H+LS)	0.745	0.756	0.743	0.506	0.512	0.502	0.684	0.693	0.681	1.010	1.007
<b>NHNet</b> (H+LSD)	<b>0.758</b>	<b>0.761</b>	<b>0.752</b>	<b>0.521</b>	<b>0.521</b>	<b>0.514</b>	<b>0.695</b>	<b>0.697</b>	<b>0.689</b>	1.018	1.017

**Table 5: Case studies on the effectiveness of distant supervision and human supervision in the clean dataset. At most three articles of each story are shown for the interest of space.**

<p><b>Gold Label:</b> brazil's copa america squad  <b>Distant Supervision:</b> brazil announce squad for copa america  <b>Distant Supervision+Human Supervision:</b> brazil copa america squad</p> <p><b>Article<sub>1</sub>:</b> Neymar Leads Brazil's Copa America Squad; Ajax's David Neres Included  RIO DE JANEIRO (AP) - Neymar and David Neres are in for the <b>Copa America</b>, Lucas Moura and Vinicius Jr. are out. Brazil coach Tite named his squad Friday...</p> <p><b>Article<sub>2</sub>:</b> Vinicius Jr. not in Brazil's Copa America squad  Brazil coach Tite has included Neymar in his 23-man <b>squad for the Copa America</b> his country will host but has yet to decide if the Paris-Saint Germain forward will retain the captaincy - while Real Madrid's teenage forward Vinicius Junior did not make the cut...</p> <p><b>Article<sub>3</sub>:</b> Brazil Announce Their Copa America Squad, Some Big Names Missing From The Team  Brazil have <b>announced</b> their <b>squad</b> ahead of the upcoming <b>Copa America</b>. Head coach Tite will be hoping that his side are able to capture the trophy on home soil and he has a stacked <b>squad</b> of players to use for the competition...</p>	<p><b>Gold Label:</b> porter jr. to play in summer league  <b>Distant Supervision:</b> michael porter jr. will play in summer league  <b>Distant Supervision+Human Supervision:</b> porter jr to play in summer league</p> <p><b>Article<sub>1</sub>:</b> Michael Porter Jr. is pain-free and expects to play in Summer League  Early Monday morning, less than 24 hours after the Denver Nuggets season ended in heartbreaking fashion, the 14th overall pick of the 2018 draft, <b>Michael Porter Jr.</b>, was made available to the media during exit interviews...</p> <p><b>Article<sub>2</sub>:</b> Michael Porter Jr. Fully Cleared, Will Play In Summer League  <b>Michael Porter Jr.</b> has been fully cleared for all basketball related activities and expects <b>to play in Summer League</b>. "I can't wait for that," said <b>Porter</b>. <b>Porter</b> said he also could have played towards the end of the 18-19 season. <b>Porter</b> was the 14th overall pick...</p> <p><b>Article<sub>3</sub>:</b> Denver Nuggets set to add Michael Porter Jr. to the mix next season  Even with an earlier playoff exit than they would have liked, after earning the No. 2 seed in the West, the Denver Nuggets look like a team on the rise. But they're set to add a notable talent, as <b>Michael Porter Jr.</b> has been medically cleared and according to Nick Kosmider of The Athletic he will <b>play in the Summer League</b>...</p>
<p><b>Gold Label:</b> walmart announces next day delivery  <b>Human Supervision:</b> amazon orders \$day to buy amazon  <b>Distant Supervision+Human Supervision:</b> walmart to offer next day delivery</p> <p><b>Article<sub>1</sub>:</b> Walmart announces next-day delivery, firing back at Amazon  <b>Walmart</b> is firing back. The biggest retailer in the world will now offer shoppers the option to have their online <b>orders delivered the next day</b>, following <b>Amazon</b>...</p> <p><b>Article<sub>2</sub>:</b> Walmart Fires Back at Amazon, Announces Next-Day Delivery on 220,000 Items  <b>Walmart</b> is <b>rolling out free next-day delivery</b> on its most popular items, increasing the stakes in the retail shipping wars with <b>Amazon</b>. The nation's largest retailer said...</p> <p><b>Article<sub>3</sub>:</b> Walmart announces next-day delivery on 200K+ items in select markets  This month, <b>Amazon</b> announced it's investing \$800 million in its warehouses and delivery infrastructure in order to double the speed of Prime shipping by reducing it to only one day. Now <b>Walmart</b> is following suit with a <b>one-day shipping announcement</b>...</p>	<p><b>Gold Label:</b> new pokemon mobile game  <b>Human Supervision:</b> dena sword launch  <b>Distant Supervision+Human Supervision:</b> new pokemon mobile game</p> <p><b>Article<sub>1</sub>:</b> A New Pokemon Mobile Game Is In Development  <b>Pokemon Sword and Shield</b> might have already been announced, but we now know there's another <b>new Pokemon game</b> on the way from <b>DeNA</b>. The Japanese developer...</p> <p><b>Article<sub>2</sub>:</b> New Pokemon Mobile Game Announced!  Exciting news, Trainers! <b>DeNA</b>, a big player in the gaming world, has announced a <b>new Pokemon mobile game</b> in their financial report! In partnership with The Pokemon Company, <b>DeNA</b> is working on a whole <b>new Pokemon mobile game</b>...</p> <p><b>Article<sub>3</sub>:</b> There's a new Pokemon mobile game in the works  A <b>new Pokemon mobile game</b> is in the works. Japanese firm <b>DeNA</b> - behind Mario Kart Tour and Super Mario Run - is working on a brand new title based on the popular franchise with The Pokemon Company. According to TechRadar, <b>DeNa</b> said in an official document: [We] plan to launch a new and exciting smartphone game...</p>

without any pre-training (H), where the initial article representations are far from perfect, the performance gap between different attention designs becomes more vivid.

## 8 CASE STUDY

We conduct case studies to better understand the advantages of the proposed model. Table 5 compares the full model with the variant without fine-tuning on human supervision and without distant supervision on the clean dataset. As one may find, the model using pure distant supervision can already generate headlines of high quality in terms of representativeness and informativeness. Fine-tuning on human labels further adjusts the length and style of the outputs by dropping undesirable verbosity such as full person names and prepositions.

The model without distant supervision fails to pay attention to important words, which results in less meaningful headlines that are often inarticulate. For instance, in the news stories discussing Walmart's move against Amazon Prime delivery, the model accidentally generates a somewhat meaningless headline regarding Amazon, as the word "Amazon" appears frequently in the news stories. In contrast, the full model generates a high-quality headline with very close semantics to the human label, replacing "announces" with "to offer". A similar case can also be observed in the story concerning "new Pokemon mobile game".

Table 6 shows a representative comparison between Self-voting Attention and Referee Attention on the noisy dataset as described in Section 7.4. When an outlier article is added to the story (article 2), the Referee Attention still assigned a relatively high attention weight to it, and hence introduced false information to the headline

**Table 6: A case study on the noisy dataset.**

<b>Gold Label:</b> austin riley mlb debut
<b>Referee Attention:</b> austin riley homers in game 7 <b>Self-Voting Attention:</b> austin riley mlb debut
<b>Article<sub>1</sub>:</b> Braves prospect Riley homers in 2nd MLB AB ATLANTA – As <b>Austin Riley</b> soaked in the excitement of highlighting his Major League <b>debut</b> with a monstrous home run that helped the Braves claim a 4-0 win over the Cardinals on Wednesday night...
<b>Article<sub>2</sub> (noise):</b> SMB completes PH Cup five-peat after gripping Game 7 win over Magnolia FIVE rings to adorn this San Miguel dynasty. The Beermen extended their reign in the PBA Philippine Cup in dramatic fashion, overcoming a 17-point deficit to beat Magnolia, 72-71, in a <b>Game Seven</b> to remember...
<b>Article<sub>3</sub>:</b> Called Up: Austin Riley Yesterday, the Braves called up <b>Austin Riley</b> , who we ranked second in their system and 33rd in our Top 100. He continued his blazing hot 2019, going 1-for-3 in his big league <b>debut</b> last night, including a home run...

(“game seven”). On the contrary, our model successfully identified the outlier through the dynamic voting process, and avoided adding noise to the generated headline.

During our study, we also find cases that can be improved. Specifically, in some cases, headlines generated by existing models may focus on different information than human labels. For example, a story labeled as “biden on china tariffs” by human experts is labeled by our algorithm as “biden criticizes trump on tariffs”. Both are good summaries for the story, but focus on different aspects. In the future, we may consider two directions to further satisfy personalized information needs.

## 9 RELATED WORK

Three lines of research are closely related: document summarization, news headline generation, and language model pre-training.

**Single-Document Summarization (SDS)** has been studied for over six decades [29]. Early extractive methods incorporate hand-crafted features and graph-based structural information to select informative sentences to form the summary [12, 33, 61]. Neural extractive models achieve significant improvement by taking advantage of effective representation learning [6, 37, 40, 69]. Recent success of seq2seq models has inspired various abstractive summarization methods with an end-to-end encoder-decoder architecture to achieve new state-of-the-art performance [53]. The encoder module represents input sentences with word embeddings, linguistic features [38], and abstract meaning representation [25, 58]. The input sequence is encoded to an intermediate representation and decoded to target sequence with an RNN or Transformer and their variants [5, 23, 26, 38]. To alleviate the Out-Of-Vocabulary (OOV) issue, prior works also incorporate various copy mechanisms [17, 54, 60] in the framework. Recent works improve the quality of generated summaries in length [20, 30, 39] and informativeness [24, 43, 46, 64]. In addition, many alternative evaluation metrics [31, 41, 63] have been proposed recently due to the limits of ROUGE.

**Multi-Document Summarization (MDS)**, on the other hand, aims to generate summaries for a set of documents. Early works on MDS explore both extractive [4, 10, 18, 34, 47] and abstractive methods [2, 15, 32, 48]. End-to-end abstractive models for MDS are limited by a lack of large-scale annotated datasets. Recent works either try to leverage external resources [26, 27] or adapt single-document summarization models to MDS tasks [3, 22, 67]. The

recently developed multi-news dataset [11] provides the first large-scale training data for supervised MDS, while such a dataset is still absent for the multi-document headline generation task.

**Headline Generation** is a special task of Document Summarization to generate headline-style abstracts of articles [55], which are usually shorter than a sentence [1]. Over the past decade, both rule-based [9], compression-based [13, 14] and statistical-based methods [1, 65] have been explored with handcrafted features and linguistic rules. Recent state-of-the-art headline generation models are dominated by end-to-end encoder-decoder architectures [16, 19, 36, 53, 68]. Similar to summarization models, the encoder module considers different formats of input representation including word position embedding [7], abstractive meaning representations [58] and other linguistic features [38]. Pointer networks [38] and length-controlling mechanisms [20] are also developed for this task. However, to the best of our knowledge, the task of headline generation for multiple documents has barely been explored before. **Language Model Pre-training** has been proven to be effective in boosting the performance of various NLP tasks with little cost [8, 45, 49, 50, 57, 70]. Recent works applying pre-trained language models also achieve significant success in summarization and headline generation tasks [52, 56, 66]. In this work, we study how pre-training at different levels can benefit multi-document headline generation.

## 10 CONCLUSIONS

In this work, we propose to study the problem of generating headline-style abstracts of articles in the context of news stories. For standard research and evaluation, we publish the first benchmark dataset, NewSHead, curated by human experts for this task. The slow and expensive curation process, however, calls for effort-light solutions to achieve abundant training data from web-scale unlabeled corpus. To this end, we propose to automatically annotate unseen news stories by distant supervision where a representative article title is selected as the story headline. Together with a multi-level pre-training framework, this new data augmentation approach provides us with a 6 times larger dataset without human curator and enables us to fully leverage the power of Transformer-based model. A novel self-voting-based article attention is applied afterward to better extract salient information shared by multiple articles. Extensive experiments have been conducted to verify NHNet’s superior performance and its robustness to potential noises in news stories.

## REFERENCES

- [1] Michele Banko, Vibhu O Mittal, and Michael J Witbrock. 2000. Headline generation based on statistical translation. In *Proceedings of the 38th Annual Meeting on Association for Computational Linguistics*. Association for Computational Linguistics, 318–325.
- [2] Regina Barzilay, Kathleen R McKeown, and Michael Elhadad. 1999. Information fusion in the context of multi-document summarization. In *Proceedings of the 37th annual meeting of the Association for Computational Linguistics*. 550–557.
- [3] Tal Baumel, Matan Eyal, and Michael Elhadad. 2018. Query focused abstractive summarization: Incorporating query relevance, multi-document coverage, and summary length constraints into seq2seq models. *arXiv preprint arXiv:1801.07704* (2018).
- [4] Jaime G Carbonell and Jade Goldstein. 1998. The use of MMR, diversity-based reranking for reordering documents and producing summaries. In *SIGIR*, Vol. 98. 335–336.
- [5] Qian Chen, Xiaodan Zhu, Zhenhua Ling, Si Wei, and Hui Jiang. 2016. Distraction-based Neural Networks for Modeling Documents. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence (IJCAI’16)*. AAAI

- Press, 2754–2760.
- [6] Jianpeng Cheng and Mirella Lapata. 2016. Neural Summarization by Extracting Sentences and Words. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 484–494.
  - [7] 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*. 93–98.
  - [8] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. 4171–4186.
  - [9] Bonnie Dorr, David Zajic, and Richard Schwartz. 2003. Hedge trimmer: A parse-and-trim approach to headline generation. In *Proceedings of the HLT-NAACL 03 on Text summarization workshop-Volume 5*. Association for Computational Linguistics, 1–8.
  - [10] Günes Erkan and Dragomir R Radev. 2004. Lexrank: Graph-based lexical centrality as salience in text summarization. *Journal of artificial intelligence research* 22 (2004), 457–479.
  - [11] Alexander Fabbri, Irene Li, Tianwei She, Suyi Li, and Dragomir Radev. 2019. Multi-News: A Large-Scale Multi-Document Summarization Dataset and Abstractive Hierarchical Model. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Florence, Italy, 1074–1084. <https://doi.org/10.18653/v1/P19-1102>
  - [12] Elena Filatova and Vasileios Hatzivassiloglou. 2004. Event-based extractive summarization. In *Text Summarization Branches Out*. 104–111.
  - [13] Katja Filippova, Enrique Alfonseca, Carlos A Colmenares, Lukasz Kaiser, and Oriol Vinyals. 2015. Sentence compression by deletion with lstms. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. 360–368.
  - [14] Katja Filippova and Yasemin Altun. 2013. Overcoming the lack of parallel data in sentence compression. (2013).
  - [15] Kavita Ganesan, ChengXiang Zhai, and Jiawei Han. 2010. Opinosis: A graph based approach to abstractive summarization of highly redundant opinions. In *Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010)*. 340–348.
  - [16] Daniil Gavrillov, Pavel Kalaidin, and Valentin Malykh. 2019. Self-Attentive Model for Headline Generation. In *European Conference on Information Retrieval*. Springer, 87–93.
  - [17] Jiatao Gu, Zhengdong Lu, Hang Li, and Victor OK Li. 2016. Incorporating Copying Mechanism in Sequence-to-Sequence Learning. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 1631–1640.
  - [18] Aria Haghighi and Lucy Vanderwende. 2009. Exploring content models for multi-document summarization. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*. Association for Computational Linguistics, 362–370.
  - [19] Yuko Hayashi and Hidekazu Yanagimoto. 2018. Headline generation with recurrent neural network. In *New Trends in E-service and Smart Computing*. Springer, 81–96.
  - [20] Yuta Kikuchi, Graham Neubig, Ryohei Sasano, Hiroya Takamura, and Manabu Okumura. 2016. Controlling Output Length in Neural Encoder-Decoders. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. 1328–1338.
  - [21] Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).
  - [22] Logan Lebanoff, Kaiqiang Song, and Fei Liu. 2018. Adapting the Neural Encoder-Decoder Framework from Single to Multi-Document Summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. 4131–4141.
  - [23] Piji Li, Wai Lam, Lidong Bing, and Zihao Wang. 2017. Deep Recurrent Generative Decoder for Abstractive Text Summarization. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. 2091–2100.
  - [24] Wei Li, Xinyan Xiao, Yajuan Lyu, and Yuanzhuo Wang. 2018. Improving neural abstractive document summarization with explicit information selection modeling. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. 1787–1796.
  - [25] Kexin Liao, Logan Lebanoff, and Fei Liu. 2018. Abstract Meaning Representation for Multi-Document Summarization. In *Proceedings of the 27th International Conference on Computational Linguistics*. 1178–1190.
  - [26] Peter J Liu, Mohammad Saleh, Etienne Pot, Ben Goodrich, Ryan Sepassi, Lukasz Kaiser, and Noam Shazeer. 2018. Generating Wikipedia by Summarizing Long Sequences. (2018).
  - [27] Yang Liu and Mirella Lapata. 2019. Hierarchical Transformers for Multi-Document Summarization. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Florence, Italy, 5070–5081. <https://doi.org/10.18653/v1/P19-1500>
  - [28] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692* (2019).
  - [29] Hans Peter Luhn. 1958. The automatic creation of literature abstracts. *IBM Journal of research and development* 2, 2 (1958), 159–165.
  - [30] Takuya Makino, Tomoya Iwakura, Hiroya Takamura, and Manabu Okumura. 2019. Global Optimization under Length Constraint for Neural Text Summarization. In *Proceedings of the 57th Conference of the Association for Computational Linguistics*. 1039–1048.
  - [31] Yuning Mao, Liyuan Liu, Qi Zhu, Xiang Ren, and Jiawei Han. 2019. Facet-Aware Evaluation for Extractive Text Summarization. *arXiv preprint arXiv:1908.10383* (2019).
  - [32] Kathleen McKeown and Dragomir R Radev. 1995. Generating summaries of multiple news articles. In *Proceedings of the 18th annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, 74–82.
  - [33] Rada Mihalcea. 2005. Language independent extractive summarization. In *ACL*, Vol. 5. 49–52.
  - [34] Rada Mihalcea and Paul Tarau. 2004. Textrank: Bringing order into text. In *Proceedings of the 2004 conference on empirical methods in natural language processing*. 404–411.
  - [35] Mike Mintz, Steven Bills, Rion Snow, and Dan Jurafsky. 2009. Distant supervision for relation extraction without labeled data. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 2-Volume 2*. Association for Computational Linguistics, 1003–1011.
  - [36] Kazuma Muroa, 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 *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Industry Papers)*. 73–82.
  - [37] Ramesh Nallapati, Feifei Zhai, and Bowen Zhou. 2017. Summarunner: A recurrent neural network based sequence model for extractive summarization of documents. In *Thirty-First AAAI Conference on Artificial Intelligence*.
  - [38] Ramesh Nallapati, Bowen Zhou, Cicero dos Santos, Caglar Gulcehre, and Bing Xiang. 2016. Abstractive Text Summarization using Sequence-to-sequence RNNs and Beyond. In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*. 280–290.
  - [39] Shashi Narayan, Shay B Cohen, and Mirella Lapata. 2018. Don't Give Me the Details, Just the Summary! Topic-Aware Convolutional Neural Networks for Extreme Summarization. *arXiv preprint arXiv:1808.08745* (2018).
  - [40] Shashi Narayan, Shay B Cohen, and Mirella Lapata. 2018. Ranking Sentences for Extractive Summarization with Reinforcement Learning. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*. 1747–1759.
  - [41] Shashi Narayan, Andreas Vlachos, et al. 2019. HighRES: Highlight-based Reference-less Evaluation of Summarization. *arXiv preprint arXiv:1906.01361* (2019).
  - [42] Karolina Owczarzak and Hoa Trang Dang. 2011. Overview of the TAC 2011 summarization track: Guided task and AESOP task. In *Proceedings of the Text Analysis Conference (TAC 2011), Gaithersburg, Maryland, USA, November*.
  - [43] Ramakanth Pasunuru and Mohit Bansal. 2018. Multi-Reward Reinforced Summarization with Saliency and Entailment. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*. 646–653.
  - [44] Over Paul and Yen James. 2004. An introduction to duc-2004. In *Proceedings of the 4th Document Understanding Conference (DUC 2004)*.
  - [45] Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep Contextualized Word Representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*. 2227–2237.
  - [46] Maxime Peyrard. 2019. A Simple Theoretical Model of Importance for Summarization. In *Proceedings of the 57th Conference of the Association for Computational Linguistics*. 1059–1073.
  - [47] Dragomir R Radev, Hongyan Jing, Małgorzata Styś, and Daniel Tam. 2004. Centroid-based summarization of multiple documents. *Information Processing & Management* 40, 6 (2004), 919–938.
  - [48] Dragomir R Radev and Kathleen R McKeown. 1998. Generating natural language summaries from multiple on-line sources. *Computational Linguistics* 24, 3 (1998), 470–500.
  - [49] Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. [n.d.]. Improving Language Understanding by Generative Pre-Training. ([n. d.]).
  - [50] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI Blog* 1, 8 (2019).

- [51] Xiang Ren, Ahmed El-Kishky, Chi Wang, Fangbo Tao, Clare R Voss, and Jiawei Han. 2015. Clustype: Effective entity recognition and typing by relation phrase-based clustering. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 995–1004.
- [52] Sascha Rothe, Shashi Narayan, and Aliaksei Severyn. 2019. Leveraging Pre-trained Checkpoints for Sequence Generation Tasks. *arXiv preprint arXiv:1907.12461* (2019).
- [53] 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*. Association for Computational Linguistics, Lisbon, Portugal, 379–389. <https://doi.org/10.18653/v1/D15-1044>
- [54] Abigail See, Peter J Liu, and 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 (Volume 1: Long Papers)*. 1073–1083.
- [55] Shi-Qi Shen, Yan-Kai Lin, Cun-Chao Tu, Yu Zhao, Zhi-Yuan Liu, Mao-Song Sun, et al. 2017. Recent advances on neural headline generation. *Journal of computer science and technology* 32, 4 (2017), 768–784.
- [56] Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. 2019. MASS: Masked Sequence to Sequence Pre-training for Language Generation. In *International Conference on Machine Learning*. 5926–5936.
- [57] Yu Sun, Shuohuan Wang, Yukun Li, Shikun Feng, Xuyi Chen, Han Zhang, Xin Tian, Danxiang Zhu, Hao Tian, and Hua Wu. 2019. ERNIE: Enhanced Representation through Knowledge Integration. *arXiv preprint arXiv:1904.09223* (2019).
- [58] 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*. 1054–1059.
- [59] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*. 5998–6008.
- [60] Oriol Vinyals, Meire Fortunato, and Navdeep Jaitly. 2015. Pointer networks. In *Advances in Neural Information Processing Systems*. 2692–2700.
- [61] Kristian Woodsend and Mirella Lapata. 2010. Automatic generation of story highlights. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, 565–574.
- [62] Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. 2016. Google’s neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144* (2016).
- [63] Stratos Xenouelas, Prodromos Malakasiotis, Marianna Apidianaki, and Ion Androutsopoulos. 2019. SUM-QE: a BERT-based Summary Quality Estimation Model. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Association for Computational Linguistics, Hong Kong, China, 6004–6010. <https://doi.org/10.18653/v1/D19-1618>
- [64] Yongjian You, Weijia Jia, Tianyi Liu, and Wenmian Yang. 2019. Improving Abstractive Document Summarization with Salient Information Modeling. In *Proceedings of the 57th Conference of the Association for Computational Linguistics*. 2132–2141.
- [65] David Zajic, Bonnie Dorr, and Richard Schwartz. 2002. Automatic headline generation for newspaper stories. In *Workshop on Automatic Summarization*. 78–85.
- [66] Haoyu Zhang, Yeyun Gong, Yu Yan, Nan Duan, Jianjun Xu, Ji Wang, Ming Gong, and Ming Zhou. 2019. Pretraining-Based Natural Language Generation for Text Summarization. *arXiv preprint arXiv:1902.09243* (2019).
- [67] Jianmin Zhang, Jiwei Tan, and Xiaojun Wan. 2018. Adapting Neural Single-Document Summarization Model for Abstractive Multi-Document Summarization: A Pilot Study. In *Proceedings of the 11th International Conference on Natural Language Generation*. 381–390.
- [68] Ruqing Zhang, Jiafeng Guo, Yixing Fan, Yanyan Lan, Jun Xu, Huanhuan Cao, and Xueqi Cheng. 2018. Question Headline Generation for News Articles. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*. ACM, 617–626.
- [69] Xingxing Zhang, Mirella Lapata, Furu Wei, and Ming Zhou. 2018. Neural Latent Extractive Document Summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Brussels, Belgium, 779–784. <https://doi.org/10.18653/v1/D18-1088>
- [70] Zhengyan Zhang, Xu Han, Zhiyuan Liu, Xin Jiang, Maosong Sun, and Qun Liu. 2019. ERNIE: Enhanced Language Representation with Informative Entities. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Florence, Italy, 1441–1451. <https://doi.org/10.18653/v1/P19-1139>