Microblog Hashtag Generation via Encoding Conversation Contexts

Yue Wang [1*] **Jing Li**[2†] **Irwin King**[1] **Michael R. Lyu**[1] **Shuming Shi**[2]
[1]Department of Computer Science and Engineering
The Chinese University of Hong Kong, HKSAR, China
[2]Tencent AI Lab, Shenzhen, China
[1]{yuewang, king, lyu}@cse.cuhk.edu.hk
[2]{ameliajli,shumingshi}@tencent.com

Abstract

Automatic hashtag annotation plays an important role in content understanding for microblog posts. To date, progress made in this field has been restricted to phrase selection from limited candidates, or word-level hashtag discovery using topic models. Different from previous work considering hashtags to be inseparable, our work is the first effort to annotate hashtags with a novel *sequence generation* framework via viewing the hashtag as a short sequence of words. Moreover, to address the data sparsity issue in processing short microblog posts, we propose to jointly model the target posts and the *conversation contexts* initiated by them with bidirectional attention. Extensive experimental results on two large-scale datasets, newly collected from English Twitter and Chinese Weibo, show that our model significantly outperforms state-of-the-art models based on classification.[1] Further studies demonstrate our ability to effectively generate rare and even unseen hashtags, which is however not possible for most existing methods.

1 Introduction

Microblogs have become an essential outlet for individuals to voice opinions and exchange information. Millions of user-generated messages are produced every day, far outpacing the human being's reading and understanding capacity. As a result, the current decade has witnessed the increasing demand for effectively discovering gist information from large microblog texts. To identify the key content of a microblog post, hashtags, user-generated labels prefixed with a "#" (such as "*#NAACL*" and "*#DeepLearning*"), have been widely used to reflect keyphrases (Zhang et al., 2016, 2018) or topics (Yan et al., 2013; Hong et al., 2012; Li et al., 2016). Hashtags can further benefit downstream applications, such as microblog search (Efron, 2010; Bansal et al., 2015), summarization (Zhang et al., 2013; Chang et al., 2013), sentiment analysis (Davidov et al., 2010; Wang et al., 2011), and so forth. Despite the widespread use of hashtags, there are a large number of microblog messages without any user-provided hashtags. For example, less than 15% tweets contain at least one hashtag (Wang et al., 2011; Khabiri et al., 2012). Consequently, for a multitude of posts without human-annotated hashtags, there exists a pressing need for automating the hashtag annotation process for them.

Target post for hashtag generation
This *Azarenka* woman needs a talking to from the umpire her weird noises are totes inappropes professionally. ***#AusOpen***
Replying messages forming a conversation
[T1] How annoying is she. I just worked out what she sounds like one of those turbo charged cars when they change gear or speed.
[T2] On the topic of noises, I was at the *Nadal-Tomic* game last night and I loved how quiet *Tomic* was compared to *Nadal*.
[T3] He seems to have a shitload of talent and the postmatch press conf. He showed a lot of maturity and he seems nice.
[T4] *Tomic* has a fantastic *tennis* brain...

Table 1: A post and its conversation snippet about "Australian Open" on Twitter. "*#AusOpen*" is the human-annotated hashtag for the target post. *Words indicative of the hashtag* are in blue and italic type.

Most previous work in this field focuses on extracting phrases from target posts (Zhang et al., 2016, 2018) or selecting candidates from a pre-

[*]This work was mainly done when Yue Wang was an intern at Tencent AI Lab.
[†]Jing Li is the corresponding author.
[1]To obtain our datasets, please contact Yue Wang and Jing Li.

Proceedings of NAACL-HLT 2019, pages 1624–1633
Minneapolis, Minnesota, June 2 - June 7, 2019.

defined list (Gong and Zhang, 2016; Huang et al., 2016; Zhang et al., 2017). However, hashtags usually appear in neither the target posts nor the given candidate list. The reasons are two folds. For one thing, microblogs allow large freedom for users to write whatever hashtags they like. For another, due to the wide range and rapid change of social media topics, a vast variety of hashtags can be daily created, making it impossible to be covered by a fixed candidate list. Prior research from another line employs topic models to generate topic words as hashtags (Gong et al., 2015; Zhang et al., 2016). These methods, ascribed to the limitation of most topic models, are nevertheless incapable of producing phrase-level hashtags.

In this paper, we approach hashtag annotation from a novel *sequence generation* framework. In doing so, we enable phrase-level hashtags beyond the target posts or the given candidates to be created. Here, hashtags are first considered as a sequence of tokens (e.g., "*#DeepLearning*" as "*deep learning*"). Then, built upon the success of sequence to sequence (seq2seq) model on language generation (Sutskever et al., 2014), we present a neural seq2seq model to generate hashtags in a *word-by-word* manner. To the best of our knowledge, *we are the first to deal with hashtag annotation in sequence generation architecture*.

In processing microblog posts, one major challenge we might face is the limited features to be encoded. It is mostly caused by the data sparsity exhibited in short and informal microblog posts.[2] To illustrate such challenge, Table 1 displays a sample Twitter post tagged with "*#AusOpen*", referring to Australian Open tennis tournament. Only given the short post, it is difficult to understand why it is tagged with "*#AusOpen*", not to mention that neither "*aus*" nor "*open*" appear in the target post. In such a situation, how shall we generate hashtags for a post with limited words?

To address the data sparsity challenge, we exploit conversations initiated by the target posts to enrich their contexts. Our approach is benefited from the nature that most messages in a conversation tend to focus on relevant topics. Content in conversations might hence provide contexts facilitating the understanding of the original post (Chang et al., 2013; Li et al., 2015). The effects of conversation contexts, useful on topic modeling (Li et al., 2016, 2018) and keyphrase extraction (Zhang et al., 2018), have never been explored on microblog hashtag generation. To show why conversation contexts are useful, we display in Table 1 a conversation snippet formed by some replies of the sample target post. As can be seen, key content words in the conversation (e.g., "*Nadal*", "*Tomic*", and "*tennis*") are useful to reflect the relevance of the target post to the hashtag "*#AusOpen*", because Nadal and Tomic are both professional tennis players. Concretely, our model employs a dual encoder (i.e., two encoders), one for the target post and the other for the conversation context, to capture the representations from the *two sources*. Furthermore, to capture their joint effects, we employ the bidirectional attention (**bi-attention**) (Seo et al., 2016) to explore the interactions between two encoders' outputs. Afterward, an attentive decoder is applied to generate the word sequence of the hashtag.

In experiments, we construct two large-scale datasets, one from English platform Twitter and the other from Chinese Weibo. Experimental results based on both information retrieval and text summarization metrics show that our model generates hashtags closer to human-annotated ones than all the comparison models. For example, our model achieves 45.03% ROUGE-1 F1 on Weibo, compared to 25.34% given by the state-of-the-art classification-based method. Further comparisons with classification-based models show that our model, in a sequence generation framework, can better produce rare and even new hashtags.

To summarize, our contributions are three-fold:

• We are the first to approach microblog hashtag annotation with *sequence generation* architecture.

• To alleviate data sparsity, we enrich context for short target posts with their *conversations* and employ a bi-attention mechanism for capturing their interactions.

• Our proposed model outperforms state-of-the-art models by large margins on two large-scale datasets, constructed as part of this work.

2 Neural Hashtag Generation Model

In this section, we describe our framework shown in Figure 1. There are two major modules: a dual encoder to encode both target posts and their conversations with a bi-attention to explore their interactions, and a decoder to generate hashtags.

[2]For instance, the eligible length of a post on Twitter or Weibo is up to 140 characters.

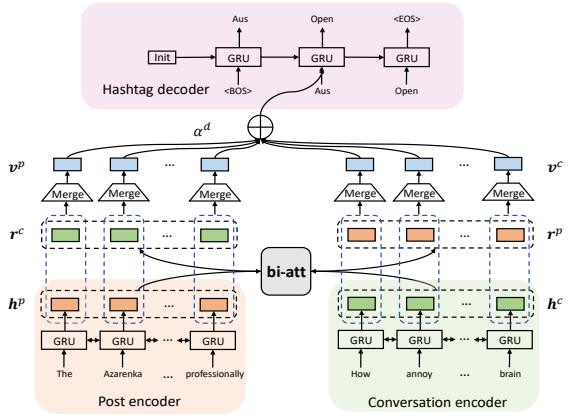

Figure 1: Our hashtag generation framework with a dual encoder, including a post encoder and a conversation encoder, where a bi-attention (bi-att) distills their salient features, followed by a merge layer to fuse them. An attentive decoder generates the hashtag sequence.

Input and Output. Formally, given a target post $\mathbf{x}^p$ formulated as word sequence $\langle x_1^p, x_2^p, ..., x_{|\mathbf{x}^p|}^p \rangle$ and its conversation context $\mathbf{x}^c$ formulated as word sequence $\langle x_1^c, x_2^c, ..., x_{|\mathbf{x}^c|}^c \rangle$, where $|\mathbf{x}^p|$ and $|\mathbf{x}^c|$ denote the number of words in the input target post and its conversation, respectively, our goal is to output a hashtag $\mathbf{y}$ represented by a word sequence $\langle y_1, y_2, ..., y_{|\mathbf{y}|} \rangle$. For training instances tagged with multiple gold-standard hashtags, we copy the instances multiple times, each with one gold-standard hashtag following Meng et al. (2017). All the input target posts, their conversations, and the hashtags share the same vocabulary V.

Dual Encoder. To capture representations from both target posts and conversation contexts, we design a dual encoder, composed of a post encoder and a conversation encoder, each taking the $\mathbf{x}^p$ and $\mathbf{x}^c$ as input, respectively.

For the post encoder, we use a bidirectional gated recurrent unit (Bi-GRU) (Cho et al., 2014) to encode the target post $\mathbf{x}^p$, where its embeddings $e(\mathbf{x}^p)$ are mapped into hidden states $\mathbf{h}^p = \langle \mathbf{h}_1^p, \mathbf{h}_2^p, ..., \mathbf{h}_{|\mathbf{x}^p|}^p \rangle$. Specifically, $\mathbf{h}_i^p = [\overrightarrow{\mathbf{h}_i^p}; \overleftarrow{\mathbf{h}_i^p}]$ is the concatenation of forward hidden state $\overrightarrow{\mathbf{h}_i^p}$ and backward hidden state $\overleftarrow{\mathbf{h}_i^p}$ for the i-th token:

$$\overrightarrow{\mathbf{h}_i^p} = GRU(e(\mathbf{x}_i^p), \overrightarrow{\mathbf{h}_{i-1}^p}), \quad (1)$$

$$\overleftarrow{\mathbf{h}_i^p} = GRU(e(\mathbf{x}_i^p), \overleftarrow{\mathbf{h}_{i+1}^p}). \quad (2)$$

Likewise, the conversation encoder converts conversations into hidden states $\mathbf{h}^c$ via another Bi-GRU. The dimensions of both $\mathbf{h}^p$ and $\mathbf{h}^c$ are d.

Bi-attention. To further distill useful representations from our two encoders, we employ the bi-attention to explore the interactions between the target posts and their conversations. The adoption of bi-attention is inspired by Seo et al. (2016), where the bi-attention was applied to extract query-aware contexts for machine comprehension. Our intuition is that the content concerning the key points in target posts might have their relevant words frequently appearing in their conversation contexts, and vice versa. In general, such content can reflect what the target posts focus on and hence effectively indicate what hashtags should be generated. For instance, in Table 1, names of tennis players (e.g., "*Azarenka*", "*Nadal*", and "*Tomic*") are mentioned many times in both target posts and their conversations, which reveals why the hashtag is "*#AusOpen*".

To this end, we first put a *post-aware* attention on the conversation encoder with coefficients:

$$\alpha_{ij}^c = \frac{\exp(f_{score}(\mathbf{h}_i^p, \mathbf{h}_j^c))}{\sum_{j'=1}^{|\mathbf{x}^c|} \exp(f_{score}(\mathbf{h}_i^p, \mathbf{h}_{j'}^c))}, \quad (3)$$

where the alignment score function $f_{score}(\mathbf{h}_i^p, \mathbf{h}_j^c) = \mathbf{h}_i^p \mathbf{W}_{bi-att} \mathbf{h}_j^c$ captures the similarity of the i-th word in the target post and the j-th word in its conversation. Here $\mathbf{W}_{bi-att} \in \mathbb{R}^{d \times d}$ is a weight matrix to be learned. Then, we compute a context vector $\mathbf{r}^c$ conveying post-aware conversation representations, where the i-th value is defined as:

$$\mathbf{r}_i^c = \sum_{j=1}^{|\mathbf{x}^c|} \alpha_{ij}^c \mathbf{h}_j^c. \quad (4)$$

Analogously, a *conversation-aware* attention on post encoder is used to capture the conversation-aware post representations as $\mathbf{r}^p$.

Merge Layer. Next, to further fuse representations distilled by the bi-attention on each encoder, we design a *merge* layer, a multilayer perceptron (MLP) activated by hyperbolic function:

$$\mathbf{v}^p = \tanh(\mathbf{W}_p[\mathbf{h}^p; \mathbf{r}^c] + \mathbf{b}_p), \quad (5)$$

$$\mathbf{v}^c = \tanh(\mathbf{W}_c[\mathbf{h}^c; \mathbf{r}^p] + \mathbf{b}_c), \quad (6)$$

where $\mathbf{W}_p, \mathbf{W}_c \in \mathbb{R}^{d \times 2d}$ and $\mathbf{b}_p, \mathbf{b}_c \in \mathbb{R}^d$ are trainable parameters.

Note that either $\mathbf{v}^p$ or $\mathbf{v}^c$ conveys the information from both posts and conversations, but with a

different emphasis. Specifically, $\mathbf{v}^p$ mainly retains the contexts of posts with the auxiliary information from conversations, while $\mathbf{v}^c$ does the opposite. Finally, vectors $\mathbf{v}^p$ and $\mathbf{v}^c$ are concatenated and fed into the decoder for hashtag generation.

Decoder. Given the representations $\mathbf{v} = [\mathbf{v}^p; \mathbf{v}^c]$ produced by our dual encoder with bi-attention, we apply an attention-based GRU decoder to generate a word sequence $\mathbf{y}$ as the hashtag. The probability to generate the hashtag conditioned on a target post and its conversation is defined as:

$$Pr(\mathbf{y}|\mathbf{x}^p, \mathbf{x}^c) = \prod_{t=1}^{|\mathbf{y}|} Pr(y_t|\mathbf{y}_{<t}, \mathbf{x}^p, \mathbf{x}^c), \quad (7)$$

where $\mathbf{y}_{<t}$ refers to $(y_1, y_2, ..., y_{t-1})$.

Concretely, when generating the t-th word in hashtag, the decoder emits a hidden state vector $\mathbf{s}_t \in \mathbb{R}^d$ and puts a global attention over $\mathbf{v}$. The attention aims to exploit indicative representations from the encoder outputs $\mathbf{v}$ and summarizes them into a context vector $\mathbf{c}_t$ defined as:

$$\mathbf{c}_t = \sum_{i=1}^{|\mathbf{x}^p|+|\mathbf{x}^c|} \alpha_{ti}^d \mathbf{v}_i, \quad (8)$$

$$\alpha_{ti}^d = \frac{\exp(g_{score}(\mathbf{s}_t, \mathbf{v}_i))}{\sum_{i'=1}^{|\mathbf{x}^p|+|\mathbf{x}^c|} \exp(g_{score}(\mathbf{s}_t, \mathbf{v}_{i'})}, \quad (9)$$

where $g_{score}(\mathbf{s}_t, \mathbf{v}_i) = \mathbf{s}_t \mathbf{W}_{att} \mathbf{v}_i$ is another alignment function ($\mathbf{W}_{att} \in \mathbb{R}^{d \times d}$) to measure the similarity between $\mathbf{s}_t$ and $\mathbf{v}_i$.

Finally, we map the current hidden state $\mathbf{s}_t$ of the decoder together with the context vector $\mathbf{c}_t$ to a word distribution over the vocabulary V via:

$$Pr(y_t|\mathbf{y}_{<t}, \mathbf{x}^p, \mathbf{x}^c) = softmax(\mathbf{W}_v[\mathbf{s}_t; \mathbf{c}_t] + \mathbf{b}_v), \quad (10)$$

which reflects how likely a word to be the t-th word in the generated hashtag sequence. Here $\mathbf{W}_v \in \mathbb{R}^{V \times 2d}$ and $\mathbf{b}_v \in \mathbb{R}^V$ are trainable weights.

Learning and Inferring Hashtags. During the training stage, we apply stochastic gradient descent to minimize the loss function of our entire framework, which is defined as:

$$\mathcal{L}(\Theta) = -\sum_{n=1}^{N} \log(Pr(\mathbf{y}_n|\mathbf{x}_n^p, \mathbf{x}_n^c; \Theta)). \quad (11)$$

Here N is the number of training instances and Θ denotes the set of all the learnable parameters.

Datasets	# of posts	Avg len of posts	Avg len of convs	Avg len of tags	# of tags per post
Twitter	44,793	13.27	29.94	1.69	1.14
Weibo	40,171	32.64	70.61	2.70	1.11

Table 2: Statistics of our datasets. Avg len of posts, convs, tags refer to the average number of words in posts, conversations, and hashtags, respectively.

Datasets	|Tagset|	$\mathcal{P}$	$\mathcal{C}$	$\mathcal{P} \cup \mathcal{C}$
Twitter	4,188	2.72%	5.58%	7.69%
Weibo	5,027	8.29%	6.21%	12.52%

Table 3: Statistics of the hashtags. |Tagset|: the number of distinct hashtags. $\mathcal{P}$, $\mathcal{C}$, and $\mathcal{P} \cup \mathcal{C}$: the percentage of hashtags appearing in their corresponding posts, conversations, and the union set of them, respectively.

In hashtag inference, based on the produced word distribution at each time step, word selection is conducted using beam search. In doing so, we generate a ranking list of output hashtags, where the top K hashtags serve as our final output.

3 Experiment Setup

Here we describe how we set up our experiments.

Datasets and Statistic Analysis. Two large-scale experiment datasets are *newly collected* from popular microblog platforms: an English Twitter dataset and a Chinese Weibo dataset. The Twitter dataset was built based on the TREC 2011 microblog track.[3] To recover the conversations, we used Tweet Search API to fetch "in-reply-to" relations in a recursive way. The Weibo dataset was collected from January to August 2014 using Weibo Search API via searching messages with the trending queries[4] as keywords. For gold-standard hashtags, we take the user-annotated hashtags, appearing before or after a post, as the reference.[5] The statistics of our datasets are shown in Table 2. We randomly split both datasets into three subsets, where 80%, 10%, and 10% of the data corresponds to training, development, and test sets, respectively.

To further investigate how challenging our problem is, we show some statistics of the hashtags in Table 3 and the distributions of hashtag frequency in Figure 2. In Table 3, we observe

[3] https://trec.nist.gov/data/tweets/

[4] http://open.weibo.com/wiki/Trends/

[5] Hashtags in the middle of a post are not considered here as they generally act as semantic elements (Zhang et al., 2016, 2018).

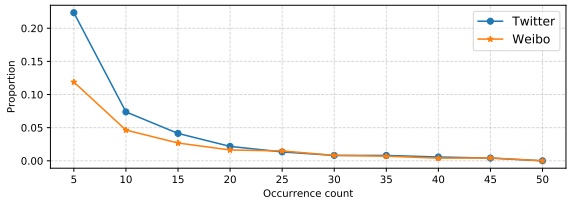

Figure 2: Distribution of hashtag frequency. The horizontal axis refers to the occurrence count of hashtags (shown with maximum 50 and bin 5) and the vertical axis denotes the data proportion.

the large size of hashtags in both datasets. Moreover, Figure 2 indicates that most hashtags only appear a few times. Given such a large and imbalanced hashtag space, hashtag selection from a candidate list, as many existing methods do, might not perform well. Table 3 also shows that only a small proportion of hashtags appearing in their posts, conversations, and either of them, making it inappropriate to directly extract words from the two sources to form hashtags.

Preprocessing. For tokenization and word segmentation, we employed the tweet preprocessing toolkit released by Baziotis et al. (2017) for Twitter, and the Jieba toolkit[6] for Weibo. Then, for both Twitter and Weibo, we further take the following preprocessing steps: First, single-character hashtags were filtered out for not being meaningful. Second, generic tags, i.e., links, mentions (@username), and numbers, were replaced with "URL" "MENTION", and "DIGIT", respectively. Third, inappropriate replies (e.g., retweet-only messages) were removed, and the remainder were chronologically ordered to form a sequence as conversation contexts. Last, a vocabulary was maintained with the $30K$ and $50K$ most frequent words, for Twitter and Weibo, respectively.

Comparisons. For experiment comparisons, we first consider a weak baseline RANDOM that randomly ranks hashtags seen from training data. Two unsupervised baselines are also considered, where words are ranked by latent topics induced with the latent Dirichlet allocation topic model (henceforth LDA), and by their TF-IDF scores (henceforth TF-IDF). Here for TF-IDF scores, we consider the N-gram TF-IDF ($N \leq 5$). Besides, we compare with *supervised* models below:

• EXTRACTOR: Following Zhang et al. (2018), we extract phrases from target posts as hashtags via sequence tagging and encode conversations with memory networks (Sukhbaatar et al., 2015).

• CLASSIFIER: We compare with the state-of-the-art model based on classification (Gong and Zhang, 2016), where hashtags are selected from candidates seen in training data. Here two versions of their classifier are considered, one only taking a target post as input (henceforth CLASSIFIER (*post only*)) and the other taking the concatenation of a target post and its conversation as input (henceforth CLASSIFIER (*post+conv*)).

• GENERATOR: A seq2seq generator (henceforth SEQ2SEQ) (Sutskever et al., 2014) is applied to generate hashtags given a target post. We also consider its variant augmented with copy mechanism (Gu et al., 2016) (henceforth SEQ2SEQ-COPY), which has proven effective in keyphrase generation (Meng et al., 2017) and also takes the post as input. The proposed seq2seq with the bi-attention to encode both the post and its conversation is denoted as OUR MODEL for simplicity.

Model Settings. We conduct model tunings on the development set based on grid search, where the hyper-parameters that give the lowest objective loss are selected. For the sequence generation models, the implementations are based on the OpenNMT framework (Klein et al., 2017). The word embeddings, with dimension set to 200, are randomly initialized. For encoders, we employ two layers of Bi-GRU cells, and for decoders, one layer of GRU cell is used. The hidden size of all GRUs is set to 300. In learning, we use the Adam optimizer (Kingma and Ba, 2014) with the learning rate initialized to 0.001. We adopt the early-stop strategy: the learning rate decreases by a decay rate of 0.5 till either it is below $1e^{-6}$ or the validation loss stops decreasing. The norm of gradients is rescaled to 1 if the $L2$-norm > 1 is observed. The dropout rate is 0.1 and the batch size is 64. In inference, we set the beam-size to 20 and the maximum sequence length of a hashtag to 10.

For CLASSIFIER and EXTRACTOR, lacking publicly available codes, we reimplement the models using Keras.[7] Their results are reproduced in their original experiment settings. For LDA, we employ an open source toolkit lda.[8]

Evaluation Metrics. Popular *information retrival* evaluation metrics F1 scores at K (F1@K)

[6] https://pypi.python.org/pypi/jieba/

[7] https://keras.io/

[8] https://pypi.org/project/lda/

Model	Twitter					Weibo				
	F1@1	F1@5	MAP	RG-1	RG-4	F1@1	F1@5	MAP	RG-1	RG-4
Baselines										
RANDOM	0.37	0.63	0.89	0.56	0.16	0.43	0.67	0.97	2.14	1.13
LDA	0.13	0.25	0.35	0.60	-	0.10	0.86	0.94	3.89	-
TF-IDF	0.02	0.02	0.03	0.54	0.14	0.85	0.73	1.30	8.04	4.29
EXTRACTOR	0.44	-	-	1.14	0.14	2.53	-	-	7.64	5.20
State of the arts										
CLASSIFIER (*post only*)	9.44	6.36	12.71	10.75	4.00	16.92	10.48	22.29	25.34	21.95
CLASSIFIER (*post+conv*)	8.54	6.28	12.10	10.00	2.47	17.25	11.03	23.11	25.16	22.09
GENERATORS										
SEQ2SEQ	10.44	6.73	14.00	10.52	4.08	26.00	14.43	32.74	37.37	32.67
SEQ2SEQ-COPY	10.63	6.87	14.21	12.05	4.36	25.29	14.10	31.63	37.58	32.69
OUR MODEL	**12.29***	**8.29***	**15.94***	**13.73***	**4.45**	**31.96***	**17.39***	**38.79***	**45.03***	**39.73***

Table 4: Comparison results on Twitter and Weibo datasets (in %). RG-1 and RG-4 refer to ROUGE-1 and ROUGE-SU4 respectively. The best results in each column are in bold. The "*" after numbers indicates significantly better results than all the other models ($p < 0.05$, paired t-test). Higher values indicate better performance.

and mean average precision (MAP) scores (Manning et al., 2008) are reported. Here, different K values are tested on F1@K and result in a similar trend, so only F1@1 and F1@5 are reported. MAP scores are also computed given the top 5 outputs. Besides, as we consider a hashtag as a sequence of words, ROUGE metrics for *summarization* evaluation (Lin, 2004) are also adopted. Here, we use ROUGE F1 for the top-ranked hashtag prediction computed by an open source toolkit pythonrouge,[9] with Porter stemmer used for English tweets. For Weibo posts, scores calculated at the Chinese character level following Li et al. (2018). We report the average scores for multiple gold-standard hashtags on ROUGE evaluation.

4 Experimental Results

In this section, we first report the main comparison results in Section 4.1, followed by an in-depth comparative study between classification and sequence generation models in Section 4.2. Further discussions are then presented to analyze our superiority and errors in Section 4.3.

4.1 Main Comparison Results

Table 4 reports the main comparison results. For CLASSIFIER, their outputs are ranked according to the logits after a $softmax$ layer. For EXTRACTOR, it is unable to produce ranked hashtags and thus no results are reported for F1@5 and MAP. For LDA, as it cannot generate bigram hashtags, no results are presented for ROUGE-SU4. In general, we have the following observations:

[9] https://github.com/tagucci/pythonrouge

• ***Hashtag annotation is more challenging for Twitter than Weibo.*** Generally, all models perform worse on Twitter measured by different metrics. The intrinsic reason is the essential language difference between English and Chinese microblogs. English allows higher freedom in writing, resulting in more variety in Twitter hashtags (e.g., abbreviations are prominent like "*aus*" in "*#AusOpen*"). For statistical reasons, Twitter hashtags are more likely to be absent in either posts or conversations (Table 3), and have a more severe imbalanced distribution (Figure 2).

• ***Topic models and extractive models are ineffective for hashtag annotation***. The poor performance of all baseline models indicates that hashtag annotation is a challenging problem. LDA sometimes performs even worse than RANDOM due to its inability to produce phrase-level hashtags. For extractive models, both TF-IDF and EXTRACTOR fail to achieve good results. It is because most hashtags are absent in target posts, as we see in Table 3 that only 2.72% hashtags on Twitter and 8.29% on Weibo appear in target posts. This confirms that extractive models, relying on word selection from target posts, cannot well fit the hashtag annotation scenario. For the same reason, copy mechanism fails to bring noticeable improvements for the seq2seq generator on both datasets.

• ***Sequence generation models outperform other counterparts.*** When comparing GENERATORS with other models, we find the former uniformly achieve better results, showing the superiority to produce hashtags with sequence generation framework. Classification models, though as

the state of the art, expose their inferiority as they select labels from the large and imbalanced hashtag space (reflected in Table 3 and Figure 2).

• ***Conversations are useful for hashtag generation.*** Among the sequence generation models, OUR MODEL achieves the best performance across all the metrics. The observation indicates the usefulness of bi-attention in exploiting the joint effects of target posts and their conversations, which further helps in identifying indicative features from both sources for hashtag generation. However, interestingly, incorporating conversations fails to boost the classification performance. The reason why OUR MODEL better exploits conversations than CLASSIFIER (*post+conv*) might be that we can attend the indicative features when decoding each word in the hashtag, which is however not possible for classification models (considering hashtags to be inseparable).

4.2 Classification vs. Generation

From Table 4, we observe that the classifiers outperform topic models and extractive models by a large margin but exhibit generally worse results than sequence generation models. Here, we present a thorough study to compare hashtag classification and generation. Four models are selected for comparison: two classifiers, CLASSIFIER (*post only*) and CLASSIFIER (*post+conv*), and two sequence generation models, SEQ2SEQ and OUR MODEL. Below, we explore how they perform to predict rare and new hashtags.

Rare Hashtags. According to the hashtag distributions in Figure 2, we can see a large proportion of hashtags appearing only a few times in the data. To study how models perform to predict such hashtags, in Figure 3, we display their F1@1 scores in inferring hashtags with varying frequency. The lower F1 score on less frequent hashtags indicates the difficulty to yield rare hashtags. The reason probably comes from the overfitting issue caused by limited data to learn from.

We also observe that sequence generation models achieve consistently better F1@1 scores on hashtags with varying sparsity degree, while classification models suffer from the label sparsity issue and obtain worse results. The better performance of the former might result from the word-by-word generation manner in hashtag generation, which enables the internal structure of hashtags (how words form a hashtag) to be exploited.

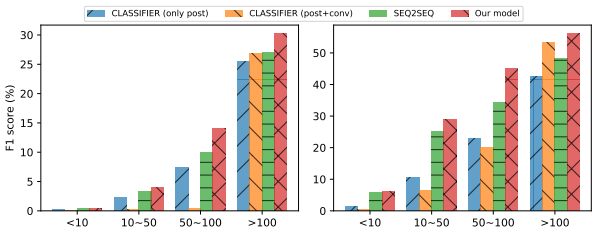

Figure 3: F1@1 on Twitter (the left subfigure) and Weibo (the right subfigure) in inferring hashtags with varying frequency. In each subfigure, from left to right shows the results of CLASSIFIER (*post only*), CLASSIFIER (*post+conv*), SEQ2SEQ, and OUR MODEL. Generation models consistently perform better.

New Hashtags. To further explore the extreme situation where hashtags are absent in the training set, we experiment to see how models perform in handling new hashtags. To this end, we additionally collect instances tagged with hashtags absent in training data and construct an external test set, with the same size as our original test set. Considering that classifiers will never predict unseen labels, to ensure comparable performance, we only adopt summarization metrics here for evaluation and report ROUGE-1 F1 scores in Table 5.

As can be seen, creating unseen hashtags is a challenging task, where unsurprisingly, all models perform poorly on this task. Nevertheless, sequence generation models perform much better on both datasets, e.g., at least 6.5x improvements over classification models observed on Weibo dataset. For Twitter dataset, the improvements are not that large, which confirms again that hashtag annotation on Twitter is more difficult due to the noisier data characteristics. In particular, compared to SEQ2SEQ, OUR MODEL achieves an additional performance gain in producing new hashtags by leveraging conversations with the bi-attention.

Model	Twitter	Weibo
CLASSIFIER (*post only*)	1.15	1.65
CLASSIFIER (*post+conv*)	1.13	1.52
SEQ2SEQ	1.33	10.84
OUR MODEL	**1.48**	**12.55**

Table 5: ROUGE-1 F1 scores (%) in producing unseen hashtags. Best results are in bold.

4.3 Further Discussions on Our Model

To further analyze our model, we conduct a quantitative ablation study, a qualitative case study, and an error analysis. We then discuss them in turn.

Ablation Study. We report the ablation study results in Table 6 to examine the relative contributions of the target posts and the conversation contexts. To this end, our model is compared with its five variants below: SEQ2SEQ (*post only*), SEQ2SEQ (*conv only*), and SEQ2SEQ (*post+conv*), using standard seq2seq to generate hashtags from their target posts, conversation contexts, and their concatenation, respectively; OUR MODEL (*post-att only*) and OUR MODEL (*conv-att only*), whose decoder only takes $\mathbf{v}^p$ and $\mathbf{v}^c$ defined in Eq. (5) and Eq. (6), respectively. The results show that solely encoding target posts is more effective than modeling the conversations alone, but exploring their joint effects can further boost the performance, especially combined with a bi-attention mechanism over them.

Model	Twitter	Weibo
SEQ2SEQ (*post only*)	10.44	26.00
SEQ2SEQ (*conv only*)	6.27	18.57
SEQ2SEQ (*post + conv*)	11.24	29.85
OUR MODEL (*post-att only*)	11.18	28.67
OUR MODEL (*conv-att only*)	10.61	28.06
OUR MODEL (*full*)	**12.29**	**31.96**

Table 6: F1@1 scores (%) for our variants.

Case Study. We further present a case study on the target post shown in Table 1, where the top five outputs of some comparison models are displayed in Table 7. As can be seen, only our model successfully generates "*aus open*", the gold standard. Particularly, it not only ranks the correct answer as the top prediction, but also outputs other semantically similar hashtags, e.g., sport-related terms like "*bbc football*", "*arsenal*", and "*murray*". On the contrary, CLASSIFIER and SEQ2SEQ tend to yield frequent hashtags, such as "*just saying*" and "*jan 25*". Baseline models also perform poorly: LDA produces some common single word, and TF-IDF extracts phrases in the target post, where the gold-standard hashtag is however absent.

Model	Top five outputs
LDA	found; stated; excited; card; apparently
TF-IDF	inappropes; umpire; woman need; azarenka woman; the umpire
CLASSIFIER	fail; facebook; just saying; quote; pro choice
SEQ2SEQ	fail; jan 25; yr; eastenders; facebook
OUR MODEL	***aus open*** ; bbc football ; bbc aus ; arsenal ; murray

Table 7: Model outputs for the target post in Table 1. "*aus open*" matches the gold-standard hashtag.

To analyze why our model obtains superior results in this case, we display the heatmap in Figure 4 to visualize our bi-attention weight matrix $\mathbf{W}_{bi-att}$. As we can see, bi-attention can identify the indicative word "*Azarenka*" in the target post, via highlighting its other pertinent words in conversations, e.g., "*Nadal*" and "*tennis*". In doing so, salient words in both the post and its conversations can be unveiled, facilitating the correct hashtag "*aus open*" to be generated.

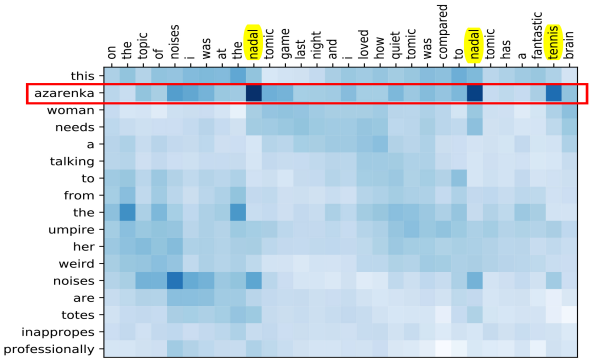

Figure 4: Visualization of bi-attention given the input case in Table 1. The horizontal axis denotes a snippet of a truncated conversation. The vertical axis shows the target post. Salient words are highlighted.

Error Analysis. Taking a closer look at our outputs, we find that one type of major errors comes from the unmatched outputs with gold standards, even as a close guess. For example, our model predicts "*super bowl*" for a post tagged with "*#steelers*", a team in super bowl. In future work, the semantic similarity should be considered in hashtag evaluation. Another primary type of error is caused by the non-topic hashtags, such as "*#fb*" (indicating the messages forwarded from Facebook). Such non-topic hashtags cannot reflect any content information from target posts and should be distinguished from topic hashtags in the future.

5 Related Work

Our work mainly builds on two streams of previous work — microblog hashtag annotation and neural language generation.

We are in the line of microblog hashtag annotation. Some prior work extracts phrases from target posts with sequence tagging models (Zhang et al., 2016, 2018). Another popular approach is to apply classifiers and select hashtags from a candidate list (Heymann et al., 2008; Weston et al., 2014; Sedhai and Sun, 2014; Gong and Zhang, 2016;

Huang et al., 2016; Zhang et al., 2017). Unlike them, we generate hashtags with a language generation framework, where hashtags in neither the target posts nor the pre-defined candidate list can be created. Topic models are also widely applied to induce topic words as hashtags (Krestel et al., 2009; Ding et al., 2012; Godin et al., 2013; Gong et al., 2015; Zhang et al., 2016). However, these models are usually unable to produce phrase-level hashtags, which can be achieved by ours via generating hashtag word sequences with a decoder.

Our work is also closely related to neural language generation, where the encoder-decoder framework (Sutskever et al., 2014) acts as a springboard for many sequence generation models. In particular, we are inspired by the keyphrase generation studies for scientific articles (Meng et al., 2017; Ye and Wang, 2018; Chen et al., 2018, 2019), incorporating word extraction and generation using a seq2seq model with copy mechanism. However, our hashtag generation task is inherently different from theirs. As we can see from Table 4, it is suboptimal to directly apply keyphrase generation models on our data. The reason mostly lies in the informal language style of microblog users in writing both target posts and their hashtags. To adapt our model on microblog data, we explore the effects of conversation contexts on hashtag generation, which has never been studied in any prior work before.

6 Conclusion

We have presented a novel framework of hashtag generation via jointly modeling of target posts and conversation contexts. To this end, we have proposed a neural seq2seq model with bi-attention over a dual encoder for capturing indicative representations from the two sources. Experimental results on two newly collected datasets have demonstrated that our proposed model significantly outperforms existing state-of-the-art models. Further studies have shown that our model can effectively generate rare and even unseen hashtags.

Acknowledgements

This work is supported by the Research Grants Council of the Hong Kong Special Administrative Region, China (No. CUHK 14208815 and No. CUHK 14210717 of the General Research Fund). We thank NAACL reviewers for their insightful suggestions on various aspects of this work.

References

Piyush Bansal, Somay Jain, and Vasudeva Varma. 2015. Towards semantic retrieval of hashtags in microblogs. In *World Wide Web Conference*.

Christos Baziotis, Nikos Pelekis, and Christos Doulkeridis. 2017. Datastories at semeval-2017 task 4: Deep LSTM with attention for message-level and topic-based sentiment analysis. In *North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.

Yi Chang, Xuanhui Wang, Qiaozhu Mei, and Yan Liu. 2013. Towards twitter context summarization with user influence models. In *International Conference on Web Search and Data Mining*.

Jun Chen, Xiaoming Zhang, Yu Wu, Zhao Yan, and Zhoujun Li. 2018. Keyphrase generation with correlation constraints. In *Empirical Methods in Natural Language Processing*.

Wang Chen, Yifan Gao, Jiani Zhang, Irwin King, and Michael R. Lyu. 2019. Title-guided encoding for keyphrase generation. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence*.

Kyunghyun Cho, Bart van Merrienboer, Çaglar Gülçehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using RNN encoder-decoder for statistical machine translation. In *Empirical Methods in Natural Language Processing*.

Dmitry Davidov, Oren Tsur, and Ari Rappoport. 2010. Enhanced sentiment learning using twitter hashtags and smileys. In *International Conference on Computational Linguistics*.

Zhuoye Ding, Qi Zhang, and Xuanjing Huang. 2012. Automatic hashtag recommendation for microblogs using topic-specific translation model. In *International Conference on Computational Linguistics*.

Miles Efron. 2010. Hashtag retrieval in a microblogging environment. In *Conference on Research and Development in Information Retrieval*.

Fréderic Godin, Viktor Slavkovikj, Wesley De Neve, Benjamin Schrauwen, and Rik Van de Walle. 2013. Using topic models for twitter hashtag recommendation. In *World Wide Web Conference*.

Yeyun Gong, Qi Zhang, and Xuanjing Huang. 2015. Hashtag recommendation using dirichlet process mixture models incorporating types of hashtags. In *Empirical Methods in Natural Language Processing*.

Yuyun Gong and Qi Zhang. 2016. Hashtag recommendation using attention-based convolutional neural network. In *International Joint Conference on Artificial Intelligence*.

Jiatao Gu, Zhengdong Lu, Hang Li, and Victor O. K. Li. 2016. Incorporating copying mechanism in sequence-to-sequence learning. In *Association for Computational Linguistics*.

Paul Heymann, Daniel Ramage, and Hector Garcia-Molina. 2008. Social tag prediction. In *Conference on Research and Development in Information Retrieval*.

Liangjie Hong, Amr Ahmed, Siva Gurumurthy, Alexander J. Smola, and Kostas Tsioutsiouliklis. 2012. Discovering geographical topics in the twitter stream. In *World Wide Web Conference*.

Haoran Huang, Qi Zhang, Yeyun Gong, and Xuanjing Huang. 2016. Hashtag recommendation using end-to-end memory networks with hierarchical attention. In *International Conference on Computational Linguistics*.

Elham Khabiri, James Caverlee, and Krishna Yeswanth Kamath. 2012. Predicting semantic annotations on the real-time web. In *ACM Conference on Hypertext and Social Media*.

Diederik P. Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*.

Guillaume Klein, Yoon Kim, Yuntian Deng, Jean Senellart, and Alexander M. Rush. 2017. Opennmt: Open-source toolkit for neural machine translation. In *Association for Computational Linguistics*.

Ralf Krestel, Peter Fankhauser, and Wolfgang Nejdl. 2009. Latent dirichlet allocation for tag recommendation. In *ACM Conference on Recommender Systems*.

Jing Li, Wei Gao, Zhongyu Wei, Baolin Peng, and Kam-Fai Wong. 2015. Using content-level structures for summarizing microblog repost trees. In *Empirical Methods in Natural Language Processing*.

Jing Li, Ming Liao, Wei Gao, Yulan He, and Kam-Fai Wong. 2016. Topic extraction from microblog posts using conversation structures. In *Association for Computational Linguistics*.

Jing Li, Yan Song, Zhongyu Wei, and Kam-Fai Wong. 2018. A joint model of conversational discourse and latent topics on microblogs. *Computational Linguistics*.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text Summarization Branches Out: Proceedings of the Association for Computational Linguistics-04 Workshop*.

Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. 2008. *Introduction to information retrieval*. Cambridge University Press.

Rui Meng, Sanqiang Zhao, Shuguang Han, Daqing He, Peter Brusilovsky, and Yu Chi. 2017. Deep keyphrase generation. In *Association for Computational Linguistics*.

Surendra Sedhai and Aixin Sun. 2014. Hashtag recommendation for hyperlinked tweets. In *Conference on Research and Development in Information Retrieval*.

Min Joon Seo, Aniruddha Kembhavi, Ali Farhadi, and Hannaneh Hajishirzi. 2016. Bidirectional attention flow for machine comprehension. In *International Conference on Learning Representations*.

Sainbayar Sukhbaatar, Arthur Szlam, Jason Weston, and Rob Fergus. 2015. End-to-end memory networks. In *Neural Information Processing Systems*.

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

Xiaolong Wang, Furu Wei, Xiaohua Liu, Ming Zhou, and Ming Zhang. 2011. Topic sentiment analysis in twitter: a graph-based hashtag sentiment classification approach. In *Conference on Information and Knowledge Management*.

Jason Weston, Sumit Chopra, and Keith Adams. 2014. #tagspace: Semantic embeddings from hashtags. In *Association for Computational Linguistics*.

Xiaohui Yan, Jiafeng Guo, Yanyan Lan, and Xueqi Cheng. 2013. A biterm topic model for short texts. In *World Wide Web Conference*.

Hai Ye and Lu Wang. 2018. Semi-supervised learning for neural keyphrase generation. In *Empirical Methods in Natural Language Processing*.

Qi Zhang, Jiawen Wang, Haoran Huang, Xuanjing Huang, and Yeyun Gong. 2017. Hashtag recommendation for multimodal microblog using co-attention network. In *International Joint Conference on Artificial Intelligence*.

Qi Zhang, Yang Wang, Yeyun Gong, and Xuanjing Huang. 2016. Keyphrase extraction using deep recurrent neural networks on twitter. In *Empirical Methods in Natural Language Processing*.

Renxian Zhang, Wenjie Li, Dehong Gao, and Ouyang You. 2013. Automatic twitter topic summarization with speech acts. *IEEE Trans. Audio, Speech & Language Processing*.

Yingyi Zhang, Jing Li, Yan Song, and Chengzhi Zhang. 2018. Encoding conversation context for neural keyphrase extraction from microblog posts. In *North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.