Neural Keyphrase Generation for Social Media Understanding

WANG, Yue

A Thesis Submitted in Partial Fulfilment
of the Requirements for the Degree of
Doctor of Philosophy
in
Computer Science and Engineering

The Chinese University of Hong Kong
January 2021

Thesis Assessment Committee

Professor CHAN Lai Wan (Chair)

Professor LYU Rung Tsong Michael (Thesis Supervisor)

Professor KING Kuo Chin Irwin (Thesis Co-supervisor)

Professor YOUNG Fung Yu (Committee Member)

Professor LI Victor On Kwok (External Examiner)

Abstract of thesis entitled:

Neural Keyphrase Generation for Social Media Understanding

Submitted by WANG, Yue

for the degree of Doctor of Philosophy

at The Chinese University of Hong Kong in January 2021

Social media platforms, such as microblogging services and online forums, are becoming increasingly popular, profoundly revolutionizing how people share information and voice opinions. Due to the wide availability of mobile devices and easy connectivity, millions of user-generated messages are produced on a daily basis, leading us to the information explosion era. As a result, the current decade has witnessed a pressing demand for automatically digesting the large volume of social media data and discovering its crucial content. To this end, *keyphrase prediction*, which aims to summarize a social media post into a set of succinct keywords (or hashtags), receives growing attention in the social media research community.

Previous progress made in this field has mainly focused on either *extraction-based* or *classification-based* approaches, which are limited in that they cannot predict keyphrases absent in the source text or the predefined candidate list. To overcome this limitation, in this thesis, we study *neural keyphrase generation* methods that enable new keyphrases to be created for social media posts. In contrast to early methods relying on hand-

crafted features, we take advantage of recent advances in deep learning and employ neural network-based frameworks that allow effective representation learning in a data-driven manner. More importantly, to alleviate the *data sparsity* issue widely exhibited in unstructured social media posts, we propose to enrich contexts via either *implicitly* exploiting the post-level latent topics or *explicitly* leveraging the user replies or the accompanying images.

First, we propose a novel topic-aware sequence generation model that leverages *implicit* latent topics to guide the keyphrase generation. Specifically, we make use of unsupervised topic models to induce a topic representation and then incorporate it into a sequence-to-sequence (seq2seq) model for generating keyphrases. Our topic models are also built with a neural architecture and allow end-to-end training of both components. Experimental results on three datasets from Twitter, Weibo, and StackExchange show that our model outperforms existing methods in keyphrase prediction, meanwhile generating more coherent topics.

Second, we explore how to leverage external knowledge for keyphrase generation. We propose to *explicitly* exploit user conversations about the target post to alleviate the data sparsity issue and design a bi-attention module to better model the interactions between the post and its conversation contexts. Unlike most prior work using classification models for recommending keyphrases, our model employs a sequence generation framework that is able to generate rare and even unseen keyphrases, which is however not possible for these existing methods. Experiments on two large-scale datasets from Twitter and Weibo validate the superiority of our model over traditional methods.

Third, we focus on predicting keyphrases for *cross-media* posts, which additionally contains images to deliver auxiliary information from authors. Apart from the informal texts, images in cross-media posts usually cover diverse categories and have a complex text-image relationship, making it difficult to identify their core meanings. To cope with this, we propose to exploit the image wordings (OCR texts and image attributes) to bridge text-image semantic gap and design a novel Multi-Modality Multi-head Attention (M^3H-Att) to better capture the dense interactions between them. Moreover, we propose a unified framework to integrate the outputs of keyphrase classification and generation and couple their advantages. Experiments on a dataset of text-image tweets demonstrate the effectiveness of our model in predicting more precise keyphrases and attending indicative information from various aspects in both modalities with our multi-head attention.

Last but not least, to better leverage the visual cues from multi-modal social media posts, we take a further step to study how to effectively learn *visual and linguistic* representations in a more general setting. For this study, we focus on the visual dialog task, one of the most challenging vision-language tasks, and propose a unified vision-dialog Transformer with BERT (VD-BERT) for it. Our model captures the intricate interactions between image and dialog within a single-stream Transformer and achieves the effective fusion of features from the two modalities via simple visually grounded training. Besides, it supports both answer ranking and answer generation seamlessly through the same architecture. Our model achieves effective vision and language fusion within a unified Transformer encoder and yields a new state of the art for visual dialog tasks.

In summary, the thesis targets keyphrase generation to facilitate a quicker understanding of the target information for users when navigating the massive amount of noisy social media data. Extensive experiments on real-world datasets show that by exploring both implicit and explicit approaches to alleviate data sparsity in social media posts, our proposed models outperform state-of-the-art methods in keyphrase prediction with better accuracy for both text-only and cross-media posts. The last pilot study in visual dialog also points out an interesting future work of extending vision-language pretraining to benefit multi-modal social media understanding, which is becoming increasingly crucial with the advent of the mobile Internet era.

論文題目：針對社交媒體理解的神經網絡關鍵詞生成
作者　　：王樾
學校　　：香港中文大學
學系　　：計算機科學與工程學系
修讀學位：哲學博士
摘要　　：
諸如微博服務和在線論壇之類的社交媒體平台正變得越來越流行，深刻地革新了人們共享信息和發表觀點的方式。由於移動設備的廣泛可用性以及便捷的連接性，每天都會產生數百萬條用戶生成的消息，這使我們進入了信息爆炸的時代。於是在當前十年中，迫切需要能自動化理解大量社交媒體數據並發現其關鍵內容的技術。為此，關鍵詞預測旨在將社交媒體帖子概括為幾個簡潔的關鍵字或主題標籤，在最近的社交媒體研究社區中受到越來越多的關注。
該領域的先前進展主要集中在基於提取的或基於分類的方法上，它們的局限性在於它們無法預測源文本或預定義的候選列表中以外的關鍵詞。在本文中，我們研究了基於神經網絡的關鍵詞生成方法，這些方法可以為社交媒體帖子創建新的關鍵詞。與依靠手工特徵提取的傳統方法相比，我們利用了深度學習方面的最新進展並採用了基於神經網絡的框架，從而以數據驅動的方式來進行有效的特徵表示學習。更重要的是，為了緩解在非結構化社交媒體帖子中廣泛出現的數據稀疏問題，我們提出隱含地利用帖子的潛在主題或者顯性利用用戶回复或圖像來豐富數據內容。
首先，我們提出了一種新穎的主題感知的序列生成模型，該模

型利用隱式潛在主題來指導關鍵詞的生成。具體而言，我們採用非監督主題模型來得到主題表示並且把它傳入到基於序列到序列（seq2seq）的框架以此來生成關鍵詞。此外，我們的主題模型也是採用神經網絡框架，能夠與另一模型進行端到端的共同訓練。在Twitter，微博和StackExchange的三個數據集上的實驗結果表明，我們的模型在關鍵詞短語預測方面優於現有方法，並且能夠學習到更連貫的主題。

其次，我們探索如何利用外部知識來幫助關鍵詞生成。我們提出隱式利用有關目標帖子的用戶對話來緩解數據稀疏性問題，並設計一種雙向注意來更好地建模該帖子及其對話上下文之間的交互關係。與大多數使用分類模型來推薦關鍵詞的先前工作不同，我們採用了序列生成模型使之能夠生成稀有甚至未出現過的詞，而這對於現有方法而言是不可能的。在Twitter和微博上的兩個大規模數據集上進行的實驗驗證了我們模型相比與傳統方法的優越性。

第三，我們關注於為多模態社交媒體帖子生成關鍵詞，這種帖子額外包含圖片來傳遞作者的輔助信息。除了不正式的文本內容，多媒體帖子中的圖片通常涵蓋來多樣的種類，並且有著複雜的圖文關係，這使得很難確定他們的核心含義。為了解決這個問題，我們提出利用圖片語意特徵（光學識別的字符以及圖片屬性）彌補圖片與文本的語義差距，並且設計了一種創新的多模態多頭注意力機制來更好的捕捉圖文之間的密集交互。此外，我們提出了一個統一框架來整合分類模型和生成模型的輸出，以此來融合兩種模型的優勢。在Twitter的圖文數據集上的實驗表明，我們的模型能夠更準確的預測關鍵詞，並且我們提出的多頭注意力機制能夠捕捉到兩種模態交互中多樣的有效信息。

最後，為了更好地利用多模式社交媒體帖子中的視覺信息，我們更進一步研究如何在更廣義的場景中有效學習視覺和語言表示形式。在本研究中，我們專注於視覺對話任務，這是最具挑

戰性的視覺語言任務之一，並為此提出了一個帶有BERT的統一視覺對話Transformer模型（VD-BERT）。我們的模型捕獲了單流Transformer中圖像和多輪對話之間的複雜交互，並通過基於圖像的簡單預訓練實現了兩種模態特徵的有效融合。此外，它通過統一的架構無縫支持答案排名和答案生成。我們的模型展示了使用統一的Transformer編碼器的强大的視覺和語言融合能力，並在視覺對話任務上取得最先進的效果。

綜上所述，本文目標是研究關鍵詞生成，以幫助用戶在瀏覽大量嘈雜的社交媒體數據時更快地了解目標信息。在真實數據集上進行的大量實驗表明，通過探索隱式和顯式方式來緩解社交媒體帖子中的數據稀疏性，我們提出的模型在針對純文本或多模態帖子的關鍵詞預測中的表現優於最新方法，具有更高的準確性。最後一項針對視覺對話任務的試點研究還指明了一個有趣的未來工作，即拓展視覺語言預訓練以幫助更好的多模態社交媒體理解，這隨著移動互聯網時代的到來變得越來越重要。

Acknowledgement

I feel highly privileged to take this opportunity to express my sincere gratitude to the people who have been instrumental and helpful on my way to pursuing my Ph.D. degree. This thesis would not have been possible without many people.

First and foremost, I wish to thank my supervisors, Prof. Michael R. Lyu and Prof. Irwin King, for their supervision during my Ph.D. study at The Chinese University of Hong Kong. I am deeply grateful for all of their time and efforts devoted to my research, from finding research problems and designing innovative solutions, to technical writing and conference presentations. I am fortunate to have them as my supervisors and have learned so much from their guidance.

I would like to extend my gratitude to my thesis committee members, Prof. Lai Wan Chan and Prof. Fung Yu Young, for their helpful comments and suggestions to this thesis and all my term presentations. Great thanks to Prof. Victor O.K. Li from The University of Hong Kong, who kindly serves as the external examiner for this thesis and provides constructive feedback.

I would like to thank Prof. Jing Li, who mentored me when I interned at Tencent AI Lab and co-authored three papers with me. She unreservedly provides me valuable support for my research that helps me achieve essential results in this thesis. I also thank my co-author Shuming Shi and friends met there,

Xiaoxue Liu, Meng Li, Ning Li, Zhantu Zhu, Lu Ji, Lingzhi Wang, Ming Fan, Haisong Zhang, Guanlin Li, and Yang Zhao for their helpful discussion and kind support.

I would also like to thank Prof. Steven Hoi and Prof. Shafiq Joty, my mentors during the internship in Salesforce Research Singapore. They provide insightful discussion and constructive feedback to my research, and always encourage me to insist on the highest standard, which helps me to substantially improve the paper quality. I also thank my colleagues, Pan Zhou, Weishi Wang, Hualin Liu, Junnan Li, Jason Wu, Samson Tan, Jiashi Feng, and my SYSU Singapore Alumni, who have made my internship very enjoyable.

I sincerely thank Jian Li and Hou Pong Chan for their contributions to my research. I am also grateful to work with my excellent group fellows: Jieming Zhu, Yu Kang, Shenglin Zhao, Hongyi Zhang, Tong zhao, Xixian Chen, Pinjia He, Xiaotian Yu, Hui Xu, Cuiyun Gao, Jichuan Zeng, Jiani Zhang, Pengpeng Liu, Shilin He, Han Shao, Yaoman Li, Haoli Bai, Wenxiang Jiao, Yifan Gao, Jingjing Li, Weibin Wu, Zhuangbin Chen, Tianyi Yang, Xinyu Fu, Ziqiao Meng, Yankai Chen, Menglin Yang, Tianyu Liu, Wenchao Gu, and Jen-tse Huang.

I would like to thank my life-long friends, Shuzhi Yu, Yufeng Yu, Yuyuan Xu, Jinyuan Cai, Zhanpeng Liu, Yongsheng Ding, Peifeng Wang, Heng Wang, Guipeng Wang, and Shaozhuang Wang for their encouragements.

Last but most importantly, I would like to sincerely thank my parents, my sister, and my fiancée Ms. Yaqing Liu. Their endless love and constant support help me go through all the difficulties during my Ph.D. study.

To my family.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Overview

As social media continues its worldwide expansion, the last decade has witnessed the revolution of interpersonal communication, shifting from offline "kitchen table conversations" to public discussions on online platforms. Among them, microblogging services and online forums have become an essential outlet for individuals to voice opinions and exchange information. While empowering individuals with richer and fresher information, the flourish of social media also results in millions of posts generated on a daily basis. According to the current statistics from Twitter[1], there are around 500 million tweets generated per day. Facing a sheer quantity of texts, language understanding has become a daunting task for human beings. Under this circumstance, there exists a pressing need to develop automatic systems capable of digesting massive social media texts and figuring out what is essential.

In recent decades, numerous machine learning techniques have been studied towards social media understanding, which cov-

[1] https://twitter.com

ers a broad set of real-world applications, such as microblog search [37, 10], sentiment analysis [34, 146], summarization [177, 23], sarcasm detection [19], user profiling [152, 38], stock price prediction [17, 108], event extraction [87] and categorization [2], and so forth. In this thesis, we target *understanding social media by generating keyphrases using deep neural networks*. In general, keyphrases are formed with words or phrases and able to convey the main idea of the target posts quickly, thereby effectively helping users when navigating a large volume of noisy social data. Specifically, in microblogs, users employ hashtags (i.e., "#DeepLearning") prefixed with a "#" to represent their key topics, which are regarded as keyphrases in this thesis following the common practice [176, 179]. Keyphrase prediction has been shown to benefit a wide range of downstream tasks, such as instant detection of trending events [151], summarizing public opinions [101], and analyzing social behavior [124].

Despite the substantial efforts made in social media keyphrase prediction, most progress to date has focused on extracting phrases from source posts [176, 179] or selecting candidates from a predefined list [45, 175, 178]. However, social media keyphrases can often appear in neither the target posts nor the given candidate list mainly due to two reasons. For one thing, social media platforms allow large freedom for users to write whatever keyphrases they like and do not set any restrictions to let them include the keywords in the posts. For another thing, due to the wide range and rapid change of social media topics, a wide variety of keyphrases can be created daily, making it impossible to be covered by a fixed candidate list. Inspired by the recent advances in neural language generation, we approach social media keyphrase prediction with a sequence generation

framework, enabling new keyphrases to be created out of both the source post and the candidate list. Specifically, we regard the keyphrase as a short sequence of words (e.g., "#DeepLearning" to be "deep learning") instead of a discrete label like previous work did. Then we build on a sequence-to-sequence (seq2seq) framework [139] to generate the keyphrases in a word-by-word manner. The seq2seq learning has been widely adopted for improving a wide spectrum of language generation tasks, such as machine translation [8, 96], text summarization [127], dialog response generation [80, 164], and question generation [82, 42], so forth.

Recently, seq2seq models have been also applied to generate keyphrases for scientific articles [100, 24, 26, 27]. However, their performance would be compromised when directly applied to noisy social media data. Unlike conventional formal and well-edited texts in these previous studies, social media content suffers from the *data sparsity* issue and poses a unique hurdle for precisely identifying its main idea. On the one hand, social media texts are usually *short* in length and thus contain limited features for understanding them. For example, microblogging services like Twitter and Sina Weibo[2] initially restrict the content length to be less than 140 characters. Although such constraints might be relieved (e.g., changed to 280 characters in tweets) through the development, users still exhibit the habits of posting short messages. For example, the average length of a tweet is around 28 characters [1], and some studies further suggest that shorter posts tend to receive more likes, comments, and sharing. On the other hand, due to the informal and colloquial nature of user-generated content, social media posts

[2] `https://weibo.com`

usually contain lots of misspelling words, grammatical errors, abbreviations, emojis, and even slangs. For example, given the tweet, "Ok no Bunz by choice. But can I atleast get some head? Lmao", it is difficult to understand its meaning based on such a short sentence, which contains not only typos like "Bunz" and "atleast", but also the specific social media domain abbreviation "Lmao" (laughing my ass off).

To address the data sparsity challenge, we explore to enrich useful features by encoding either *implicit* contexts like latent topics (W1) or *explicit* contexts like user comments (W2) and accompanying images (W3). In W1, we explore the effects of latent topics inferred from unsupervised topic models [15, 102] in aiding keyphrase prediction. Intuitively, the learned topics can narrow down the search space and serve as auxiliary contexts to indicate the keyphrases. While latent topics have been shown to benefit short text classification [174], it is unclear how it can help keyphrase prediction, which has a much larger vocabulary space (thousands vs. up to 50 classes in [174]) and the more complex multi-step decoding process compared to the one-step classification. Our W1 aims to fill this gap by proposing a topic-aware keyphrase generation model, which intelligently leverages topic information to guide keyphrase generation.

In W2 and W3, we resort to explicitly exploiting external knowledge to enhance keyphrase prediction. Social media platforms like Twitter allow users to form conversations on interests by retweet with comments or replying to previous messages to voice their opinions. Table 1.1 shows an example target post and the corresponding conversation initiated by it. We can hardly identify its keyphrase as Super Bowl from the target post and might just know it is a comment for some sports games given

Target post for hashtag generation
Thank you fox for showing the good sposmanship segment! That's what it should always be like. ***#SuperBowl***
Replying messages forming a conversation
[**T1**] Bet you are happy dancing right about now lol! You are the biggest *Steelers* fan I know, so I have been thinking of you tonight.
[**T2**] Thank you! That's a huge compliment. They have won a lot this season. It would have been poetic to end the season that way.
[**T3**] Yes, just think of all the money you will save, not having to buy all the *SuperBowl* champions gear.

Table 1.1: A post and its conversation snippet about "Super Bowl" on Twitter. "*#SuperBowl*" is the user tagged keyphrase for the target post. *Words indicative of the keyphrase* are in blue and italic type.

such limited information. To deal with this, we leverage the user conversations to enrich contexts, which has been shown to benefit the understanding of the original post [23, 81]. As can be seen from the example, key content words in the conversation are useful to unveil the reason why it is tagged with "#SuperBowl", e.g., "Steelers" in the first reply message is a famous team in a Super Bowl football game, and even 'the keyphrase 'SuperBowl" directly appears in the third reply message. In W2, we explore how to make use of the user conversations to better understand the target post.

Thanks to improved smartphones and the flourish of mobile Internet, cross-media posts with matching images are becoming ubiquitous and bring additional difficulties for understanding them. Traditional keyphrase prediction methods relying only on the textual information would achieve suboptimal performance as they neglect the critical clues conveyed from the images. To illustrate our motivation, we depict two cross-media tweet examples in Figure 1.1. In the post (a), the text is an

Post (a): I was watching all the bees Honeybee collecting pollen on the flowers Bouquet *#Cats*

Post (b): Congrats producer of the year, non-classical winner - Williams *#Grammys*

Figure 1.1: Two multimedia posts from Twitter where texts offer limited help in identifying their keyphrases while images provide essential clues.

anthropomorphic description and hardly unveils the key content *cats*, which can be clearly signaled by the image. As for the post (b), the keyphrase *grammys* is directly reflected by the optical characters in the image. Inspired by these examples, in W3, we explore how to encode matching images for compensating the limited contexts exhibited in the texts. Notably, studying the combined effects of text and image in social media is more challenging than traditional vision-language tasks like image captioning or visual question answering (VQA) [6], where the two modalities often have most semantics shared and their images mostly are natural scene photos. By contrast, texts and images in cross-media posts are not necessarily connected in semantic space and can have a variety of relationships. A recent finding [142] points out there can be four diverse text-image semantic relations (depending on whether text is represented in image and whether image adds to semantics in text) on

Twitter. Besides, social media images tend to have a more diverse category apart from photos, e.g, the poster with texts in post (b). To handle these unique challenges, we propose novel methods for distilling more useful features from the image and capture its interaction with texts.

So far, while W1 and W2 focus on the single modality (text-only), W3 explores keyphrase generation in a more challenging multi-modal setting, where interactions between text and image should be effectively captured and exploited. To further investigate how to achieve better vision and language fusion, we focus on a more general vision-language task visual dialog (W4), where an agent is asked to answer a series of questions conditioned on an image and previous dialog turns. By pretraining with visually grounded self-supervised objectives in the visual dialog task, self-attention in Transformer [141] can capture the cross-modality interaction more effectively. Such findings provide a strong indication that vision-language pretraining would benefit the cross-media understanding for keyphrase generation in W3, where its proposed model is also built on a Transformer.

To better illustrate the structure of our thesis, we show the roadmap of our contributions in Figure 1.2. Our thesis targets at *social media keyphrase generation* and proposes to encode *implicit* contexts like latent topics [148] and *explicit* contexts like user conversations [150] and images [149] to alleviate the data sparsity in social media. To explore better ways for fusing multi-modal features, we take a further step to study a more challenging visual dialog task. We extensively investigate the effects of vision-language pretraining [147] for vision and dialog fusion, which points out an interesting future work to adapt such pretraining back to benefit cross-media understanding.

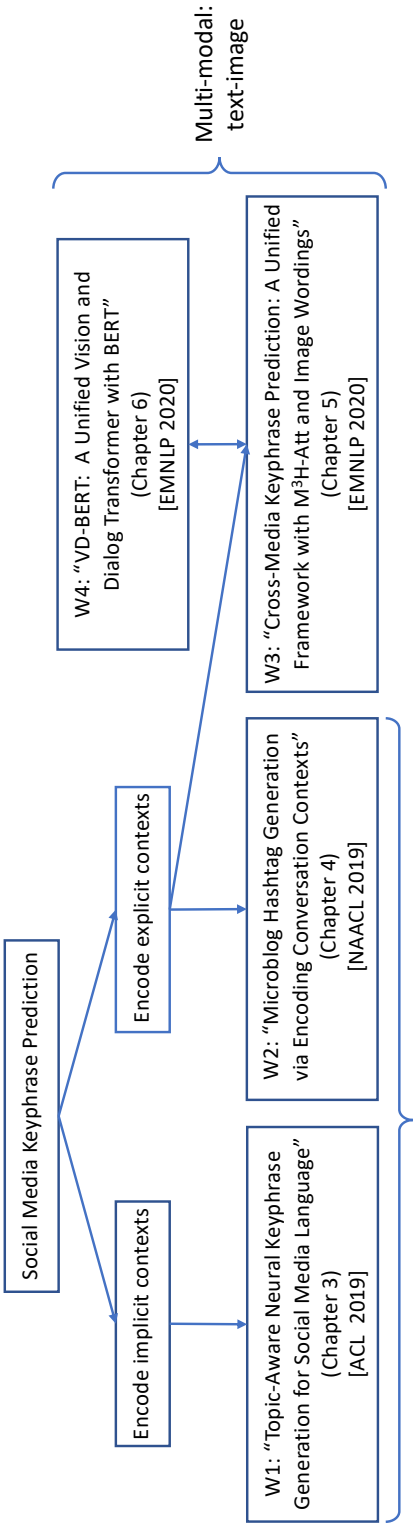


Figure 1.2: The roadmap of this thesis.

1.2 Thesis Contributions

In this thesis, we make contributions to neural keyphrase generation for social media understanding as follows.

- **Encoding Implicit Topics for Keyphrase Generation [148]**
  To mitigate the data sparsity in social media posts, we propose a novel topic-aware keyphrase generation model that leverages *implicit* latent topics to enrich useful features. Specifically, we propose a sequence-to-sequence (seq2seq) based framework that considers latent topics for better keyphrase prediction. Instead of employing traditional topic models, we exploit a neural topic model that can be seamlessly integrated into our seq2seq model for the end-to-end joint training. Experimental results on three newly constructed datasets from Twitter, Weibo, and StackExchange show that our model outperforms previous keyphrase prediction methods while generating more coherent topics.

- **Encoding Explicit Conversation for Keyphrase Generation [150]**
  In this work, we propose to *explicitly* exploit user conversations about the target post to better predict keyphrases for microblog posts. Unlike most prior work that regards keyphrase to be inseparable and employs classification models for keyphrase recommendation, we propose a sequence generation model to generate keyphrase in a word-by-word manner, enabling rare and even unseen keyphrases to be created. Moreover, we design a bi-attention module to

model the interactions between the post and its conversation contexts. Extensive experiments on two datasets from Twitter and Weibo validate our model's superiority over traditional methods with more accurate keyphrase predictions.

- **Cross-Media Keyphrase Prediction: A Unified Framework with Multi-Modality Multi-Head Attention and Image Wordings [149]**
  We explore another *explicit* knowledge from the visual modality, which is the ubiquitous accompanying images in *cross-media* tweets. Due to social media's informal style, tweet images often have an exceptionally diverse category and have a complicated relationship with the target texts. To distill indicative signals from the noisy cross-media posts, we propose to exploit the image wordings to bridge the text-image semantic gap and design a novel Multi-Modality Multi-head Attention (M^3H-Att) to better capture the dense interactions between them. Moreover, we propose a unified framework to leverage the outputs of keyphrase classification and generation and couple their advantages. Extensive experiments on a dataset of text-image tweets demonstrate the effectiveness of our model in predicting more precise keyphrases and being able to attend information from various aspects in both modalities with M^3H-Att.

- **Vision-Language Pretraining for Visual Dialog [147]**
  To better leverage the visual cues for understanding multi-modal social media posts, we take a further step to study how to effectively learn *visual and linguistic* representations

in a more general task visual dialog. In this task, an agent is asked to answer a series of questions based on the joint understanding of an image and the dialog history. We propose a unified vision-dialog Transformer with BERT (VD-BERT). Our model captures the intricate interactions between image and dialog and achieves the effective fusion of features from the two modalities via simple visually grounded training. Besides, it supports both answer ranking and answer generation seamlessly through the same architecture. Our model yields a new state of the art in discriminative settings and promising results in generative settings for visual dialog tasks.

1.3 Thesis Organization

The remainder of this thesis is organized as follows.

- **Chapter 2**
  This chapter presents a systematic review of the background knowledge and related work in neural keyphrase prediction and social media research. First, we briefly introduce some basic knowledge of deep neural networks, on which all the proposed models in the thesis are based. Then we review existing techniques for keyphrase prediction tasks, which can be divided into extraction, classification, and generation methods. Lastly, we review some recent advances in social media research for both text-only and multi-modal settings.

- **Chapter 3**
  In this chapter, we present a topic-aware neural keyphrase

generation model for social media posts. We first define the keyphrase generation problem and introduce our motivations in Section 3.1. Then we present the formulation of our proposed topic-aware keyphrase generation approach that consists of two components (neural topic model and keyphrase generation model) in Section 3.2. After that, we introduce our experiment setup including the collection of three social media datasets in Section 3.3. We comprehensively analyze the experimental results in Section 3.4 and conclude this work in Section 3.5.

- **Chapter 4**
  In this chapter, we propose to approach microblog keyphrase annotation as a sequence generation problem. We first give an overview of the task in Section 4.1 and introduce our neural keyphrase generation model for that in Section 3.2. Then we introduce how to construct the dataset and set up the experiments in Section 3.3. Lastly, Section 3.4 shows some empirical results compared to previous methods and Section 3.5 concludes this work.

- **Chapter 5**
  In this chapter, we propose a unified framework for cross-media keyphrase prediction. We first briefly introduce its unique challenges compared to conventional vision-language tasks and our motivations to address them in Section 5.1. Then Section 5.2 gives an overview of our proposed model, which consists of a multi-modality encoder to digest features from three modalities, a multi-modality multi-head attention to capture their complex interactions, and a unified keyphrase prediction module

to couple the advantages of keyphrase classification and generation. After that, we introduce and analyze the newly-constructed multi-modal tweet dataset together with experiment setup in Section 5.3. In the end, Section 5.4 shows the experimental results and Section 5.5 concludes this work.

- **Chapter 6**
  In this chapter, we focus on a more general and challenging multi-modal task visual dialog. We first introduce the task and our motivations to improve it in Section 6.1, followed by a brief review of its related work in Section 6.2. Then we introduce our VD-BERT model in Section 5.2, which employs a single-stream vision-dialog Transformer encoder to encode the image and its multi-turn dialogs and visual grounded training objectives to encourage their effective fusion, together with a ranking optimization module to fine-tune the final predictions. We introduce the experiment setup in Section 6.4 and show the empirical results in Section 6.5. Lastly, Section 6.6 concludes this work.

- **Chapter 7**
  The last chapter summarizes the contributions of this thesis and presents some potential future research directions about social media keyphrase prediction.

□ **End of chapter.**

Chapter 2

Background Review

In this chapter, we review the background knowledge and related work of our research contributions. We first introduce some basic knowledge of deep neural networks on which our proposed models are built in Section 2.1. Then we review the related work of keyphrase prediction in Section 2.2, which can be categorized into extraction-based methods, classification-based methods, and generation-based methods. After that, we review related research for social media understanding with text-only and multi-modal content in Section 2.3.

2.1 Neural Network Basics

A great number of recent approaches for keyphrase prediction are based on deep neural network models. These models avoid the need of feature engineering and allow effective representation learning via a purely data-driven manner. In this section, we review some background knowledge for most neural models, including sequence encoders with building blocks like word embeddings and Recurrent Neural Networks (RNNs), and the sequence-to-sequence (seq2seq) models with copy mechanisms

for generation tasks. We also review the recent advances of the powerful Transformer architecture and its exceptional use for pretraining. Throughout the thesis, we employ $\mathbf{W}$ and $\mathbf{b}$ to respectively denote a trainable projection matrix and a trainable bias vector in a neural network model.

2.1.1 Neural Sequence Encoders

Given an input sequence, neural sequence encoders aim to encode the sequential contexts via learning high-dimensional representations. In recent decades, with the expansion of deep learning in all kinds of areas, neural networks like RNNs have been widely adopted as the backbone for modeling a sequence of inputs. Typically, the encoding procedure consists of two steps: first, map the discrete input tokens into continuous vectors via an embedding lookup table; second, RNNs such as LSTMs and GRUs are employed to derive their contextual representations.

Word Embeddings

Formally, let us define a discrete input sequence as $\mathbf{x} = \{x_1, ..., x_n\}$, where n is the number of tokens and each token x_i is in a vocabulary V. Word embedding aims to map each x_i into a distributed vector $e_i \in \mathbb{R}^{d_e}$ with a lookup table $\mathbf{E} \in \mathbb{R}^{d_e \times |V|}$, where d_e denotes the embedding size. After that, these embeddings will be integrated into other neural modules and jointly trained. In general, word embedding is deemed as the foundation of the successful use of deep learning in NLP.
Apart from training the embedding matrix $\mathbf{E}$ from scratch, one can also load pretrained ones like word2vec [104] and GloVe [114] as a better start point. These embeddings are trained from

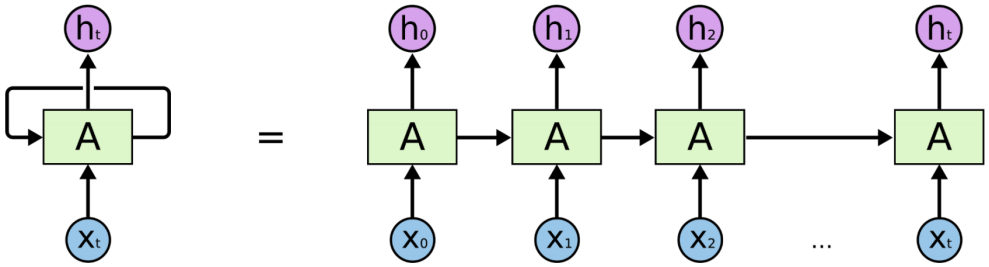


Figure 2.1: Illustration of RNNs. Figure is from [32].

some large corpus with self-supervised objectives and capture basic task-agnostic language representations. To further deal with the out-of-vocabulary problem when handling big corpus, character-level or sub-word representations are widely employed in many NLP applications, such as CharCNN [67], FastText [16] and Byte-Pair Encoding [128].

Recurrent Neural Networks

Recurrent Neural Networks (RNNs) [122] have been extensively adopted as the backbone for encoding sequences due to the unique recurrent structure. Specifically, as shown in Figure 2.1, such a unique recurrent module can be unrolled along multiple time steps, thereby enabling RNNs to encode arbitrarily long sequences. Moreover, RNNs employ parameter sharing for each time step and largely reduce the parameter numbers. Formally, an RNN cell is represented as:

$$\mathbf{h}_t = \text{RNN}(\mathbf{h}_{t-1}, \mathbf{x}_t; \theta), \tag{2.1}$$

where $\mathbf{x}_t$ and $\mathbf{h}_t$ are the embeddings of the input token and hidden state respectively at time step t, and θ is the shared parameters for all time steps, which will be learned by back propagating gradients. We omit the θ for simplicity below.

Despite its ability to encode arbitrarily long sequences, RNNs suffer from the long-range dependencies, where gradients propagated over many time steps tend to either vanish or explode [47]. The underlying reason is that training on long-term dependencies will produce exponentially smaller weights (due to the multiplication of many Jacobians) compared to the short-term ones. To alleviate such issues, researchers introduce a gating mechanism to better control the message propagation along long-term dependencies. Concretely, it dynamically determines how much of past information will be discarded or kept at each cell state. Among these methods, LSTMs and GRUs are two widely adopted RNN variants.

Formally, an LSTM cell employs three gates to update its hidden state at each time step via:

$$\mathbf{u}_t = \sigma(\mathbf{W}_{xu}\mathbf{x}_t + \mathbf{W}_{hu}\mathbf{h}_{t-1} + \mathbf{b}_u), \tag{2.2}$$

$$\mathbf{f}_t = \sigma(\mathbf{W}_{xf}\mathbf{x}_t + \mathbf{W}_{hf}\mathbf{h}_{t-1} + \mathbf{b}_f), \tag{2.3}$$

$$\mathbf{o}_t = \sigma(\mathbf{W}_{xo}\mathbf{x}_t + \mathbf{W}_{ho}\mathbf{h}_{t-1} + \mathbf{b}_o), \tag{2.4}$$

$$\mathbf{c'}_t = \tanh(\mathbf{W}_{xc'}\mathbf{x}_t + \mathbf{W}_{hc'}\mathbf{h}_{t-1} + \mathbf{b}_{c'}), \tag{2.5}$$

$$\mathbf{c}_t = \mathbf{f}_t \circ \mathbf{c}_{t-1} + \mathbf{u}_t \circ \mathbf{c'}_t, \tag{2.6}$$

$$\mathbf{h}_t = \mathbf{o}_i \circ \tanh(\mathbf{c}_t), \tag{2.7}$$

where $\sigma(.)$ is the sigmoid function and $\circ$ denotes element-wise multiplication. $\mathbf{c}_t \in \mathbb{R}^{d_h}$ is the current internal cell state with dimension size to be d_h. $\mathbf{u}_t, \mathbf{f}_t, \mathbf{o}_t \in \mathbb{R}^{d_h}$ are the input, forget, and output gates to decide how much of information will be added to the cell state, removed from the cell state, and passed to hidden states, respectively.

Compared to LSTMs, GRUs simplify the gating mechanism with only two gates, i.e., reset and update gate, and still achieve comparable performance. Formally, a GRU cell updates the

hidden state at each time step via:

$$\mathbf{r}_t = \sigma(\mathbf{W}_{xr}\mathbf{x}_t + \mathbf{W}_{hr}\mathbf{h}_{t-1} + \mathbf{b}_r), \tag{2.8}$$

$$\mathbf{z}_t = \sigma(\mathbf{W}_{xz}\mathbf{x}_t + \mathbf{W}_{hz}\mathbf{h}_{t-1} + \mathbf{b}_z), \tag{2.9}$$

$$\tilde{\mathbf{h}}_t = \tanh(\mathbf{W}_{x\tilde{h}}\mathbf{x}_t + \mathbf{W}_{h\tilde{h}}(\mathbf{r}_t \circ \mathbf{h}_{t-1}) + \mathbf{b}_{\tilde{h}}), \tag{2.10}$$

$$\mathbf{h}_t = \mathbf{z}_t \circ \mathbf{h}_{t-1} + (1 - \mathbf{z}_t) \circ \tilde{\mathbf{h}}_t, \tag{2.11}$$

where $\mathbf{r}_t$ is the reset gate that controls how much of past information will be neglected while $\mathbf{z}_t$ is the update gate that determines how much of past information will be reserved. GRUs can achieve comparable performance as LSTMs but with a simpler architecture and will be adopted in multiple approaches proposed in the thesis.

To encode more useful contexts, bi-directional RNN encoders have been widely adopted to model the sequential input from two directions. Specifically, it employs a forward RNN and a backward RNN to respectively read input sequence $\mathbf{x}$ from $\mathbf{x}_1$ to $\mathbf{x}_n$ and from $\mathbf{x}_n$ to $\mathbf{x}_1$:

$$\overrightarrow{\mathbf{h}}_t = \text{RNN}(\mathbf{x}_t, \overrightarrow{\mathbf{h}}_{t-1}), \tag{2.12}$$

$$\overleftarrow{\mathbf{h}}_t = \text{RNN}(\mathbf{x}_t, \overleftarrow{\mathbf{h}}_{t+1}). \tag{2.13}$$

The forward and backward hidden states $\overrightarrow{\mathbf{h}}_t$ and $\overleftarrow{\mathbf{h}}_t$ are concatenated to form a hidden representation $\mathbf{h}_i = [\overrightarrow{\mathbf{h}}_t; \overleftarrow{\mathbf{h}}_t]$ for the input $\mathbf{x}_t$. As such, $\mathbf{H} = \{\mathbf{h}_1, ..., \mathbf{h}_n\}$ can be deemed as the contextual representations for the whole sequence.

After obtaining the sequential representations, one can feed them for a sequence decoder to generate another sequence (Section 2.1.2), or directly make a prediction based on them. For the latter, the representations are usually aggregated into a vector via max or mean pooling and transformed to $\mathbf{v}$ via

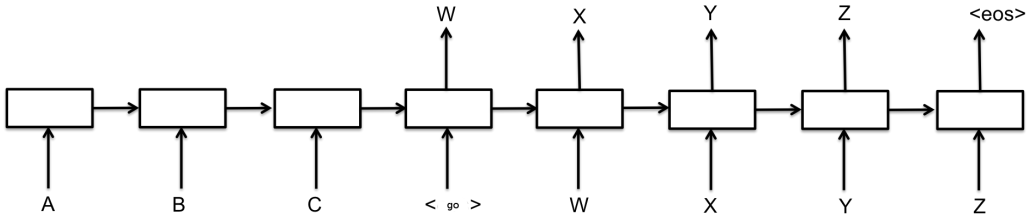


Figure 2.2: Illustration of seq2seq models.

multi-layer perceptron (MLP) for later use. Take the multi-class classification as an example. A softmax layer can directly operate on them to yield the probabilities:

$$\text{softmax}(\mathbf{v}) = \frac{\exp(\mathbf{v})}{\sum_{k=1} \exp(\mathbf{v}_k)}, \tag{2.14}$$

where k is the number of classes and the softmax function produces a normalized distribution over the class vocabulary.

2.1.2 Sequence to Sequence Models

Apart from making a discriminative prediction, one can also predict another target sequence based on the learned source sequence representations $\mathbf{H}$. This is well known as the sequence to sequence (seq2seq) learning that typically employs an encoder and decoder framework (as shown in Figure 2.2). The seq2seq models have been originally proposed for neural machine translation task and later widely adopted as the paradigm for a number of language generation tasks, e.g., dialog response generation, question generation, text summarization, and also the keyphrase generation [100, 26, 27, 22, 148]. RNNs are the most popular backbone for the seq2seq models.

Formally, given the source input sequence $\mathbf{x} = \{x_1, ..., x_n\}$, an encoder reads this sequential input and summarizes them into a context vector $\mathbf{c}$ (one popular choice is to employ the $\mathbf{h}_t$). Then

a decoder generates the output sequence $\mathbf{y} = \{y_1, ..., y_m\}$ based on the fixed-size context vector. At each decoding step t, the decoder calculate a hidden state $\mathbf{s} \in \mathbb{R}^{d_s}$:

$$\mathbf{s}_t = \text{RNN}(\mathbf{s}_{t-1}, \mathbf{y}_{t-1}), \tag{2.15}$$

where $\mathbf{s}_{t-1}$ is the previous decoder hidden state and $\mathbf{y}_{t-1}$ the embedded word predicted at the last time step. Usually, $\mathbf{c}$ is employed as the initial state $\mathbf{s}_0$ and a special token `<BOS>` (begin of sentence) is inserted as the first token y_0 to trigger the decoding process. To ensure the autoregressive property, the decoder is built on uni-directional RNNs.
Based on the hidden state $\mathbf{s}_t$, the decoder employs a MLP with softmax to derive a probability distribution over words in a predefined vocabulary V:

$$P(y_t|\mathbf{y}_{<t}, \mathbf{x}) = \text{softmax}(\mathbf{W}_V \mathbf{s}_t + \mathbf{b}_V), \tag{2.16}$$

where $\mathbf{y}_{<t}$ refers to $\{y_1, y_2, ..., y_{t-1}\}$. The decoding process is usually terminated when a special token `<EOS>` (end of sentence) is emitted.

Attention Mechanism

However, the traditional encoder-decoder frameworks often suffer from the so-called *hidden state bottleneck* that is caused by attempting to summarize an arbitrarily long sequence into a fixed-size vector, which is unrealistic and inevitably restricts the capability of capturing long-range dependencies. To address this issue, attention mechanisms have been widely adopted to allow the decoder to fully make use of all the contexts in the source sequence.

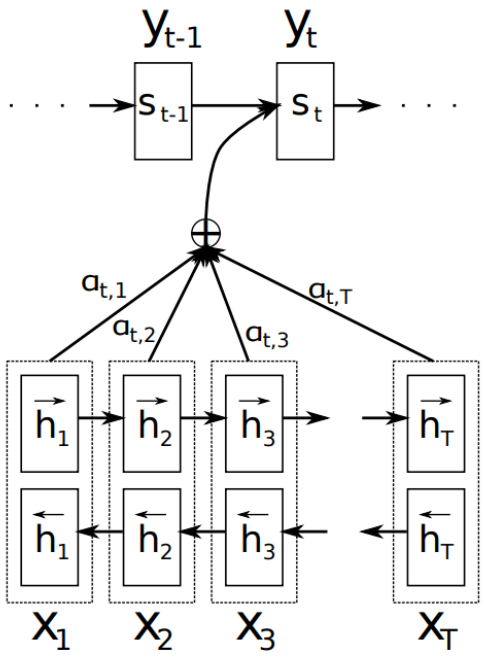


Figure 2.3: Illustration of attention mechanism.

As illustrated in Figure 2.3, attention mechanisms add shortcut connections from each decoder state $\mathbf{s}_t$ to all the encoder states in $\mathbf{H}$. Specifically, the decoder computes an attention score $\alpha_{t,i}$ using the following equations:

$$\alpha_{t,i} = \mathbf{v}^T \tanh(\mathbf{W}_h \mathbf{h}_i + \mathbf{W}_s \mathbf{s}_t + \mathbf{b}_\alpha), \tag{2.17}$$

$$a_{t,i} = \frac{\exp(\alpha_{t,i})}{\sum_{j=1}^{n} \exp(\alpha_{t,j})}, \tag{2.18}$$

where $\mathbf{v} \in \mathbb{R}^{d_\alpha \times 1}, \mathbf{W}_s \in \mathbb{R}^{d_\alpha \times d_s}, \mathbf{W}_h \in \mathbb{R}^{d_\alpha \times d_h}, \mathbf{b}_\alpha \in \mathbb{R}^{d_\alpha}$. The attention weight $\alpha_{t,i}$ measures the compatibility score between the decoder state $\mathbf{s}_t$ and encoder state $\mathbf{h}_i$, which will be used to compute the context vector $\mathbf{c}_t$ via:

$$\mathbf{c}_t = \sum_{i=1}^{n} \alpha_{t,i} \mathbf{h}_i. \tag{2.19}$$

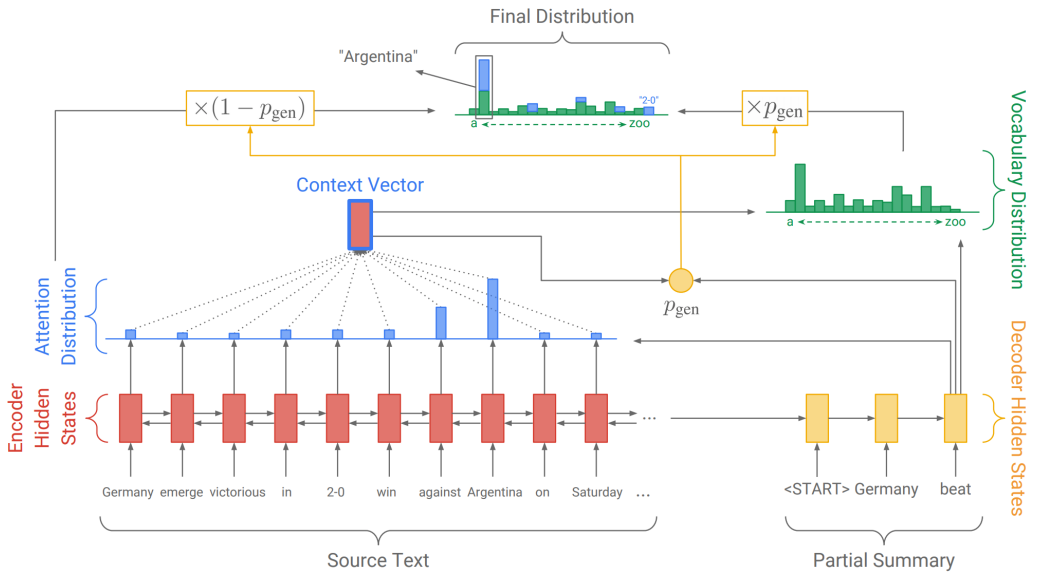


Figure 2.4: Illustration of copy mechanism. Figure is from [127]

Here $\mathbf{c}_t$ is a dynamic context vector depending on the decoding state and represents the relevant information distilled from the source sequence. At each time step t, the decoder takes the context vector into account and predict the output y_t via:

$$P(y_t|\mathbf{y}_{<t}, \mathbf{x}) = \text{softmax}(\mathbf{W}_V[\mathbf{s}_t; \mathbf{c}_t] + \mathbf{b}_V), \tag{2.20}$$

where $[;]$ denotes the concatenation operation. By incorporating the context vector $\mathbf{c}_t$, attentional seq2seq models overcome the hidden state bottleneck issue and are more capable of capturing long-range dependencies.

Copy Mechanism

In many language generation tasks, the output sequence often contains some shared contexts with the source sequence, e.g., text summarization and the keyphrase generation. In such cases, copy mechanism [106, 48, 127] that allows the decoder to directly extract source words as the predicted outputs has been widely

employed for improving the overall performance. Another major advantage of the copy mechanism is to help handle the out-of-vocabulary (OOV) problem, which is a well-known phenomenon in NLP, especially when processing large-scale corpus. As it is impractical to maintain a large vocabulary for all the words, a common approach is to set a fixed vocabulary size for most frequent words and regard the rest long-tail words as unknown words (often marked as `<UNK>`). Copy mechanism brings an extra opportunity to recover these words if they directly appear in the source sequence.

Figure 2.4 illustrates one of the most popular copy mechanisms proposed by See et al. [127], where they devise a pointer and generator model for text summarization tasks. Specifically, to select words from the source sequence, the copy mechanism often employs the attention scores α in Eq. (2.18) as the extractive probabilities. Besides, it computes a *soft switch* p_{gen} to determine whether to copy from the local source sequence or generate from the global vocabulary:

$$p_{gen} = \sigma(\mathbf{u}_g^T[\mathbf{c}_t; \mathbf{s}_t; \mathbf{y}_{t-1}]), \tag{2.21}$$

where σ is a sigmoid function that maps to $p_{gen} \in [0, 1]$. As such, the final prediction is computed by linearly combining both probabilities:

$$P_{final}(y_t) = p_{gen}P(y_t) + (1 - p_{gen}) \sum_{i:x_i=y_t}^{n} \alpha_{t,i}, \tag{2.22}$$

where p_{gen} controls the percentage of contribution that each module makes to final predictions, e.g., $p_{gen} = 1$ represents the original model without a copy mechanism.

2.1.3 Transformer and Pretraining

Recently, Transformers [141] relying only on attention mechanism have received growing attention and revolutionized numerous NLP tasks, including both natural language generation (NLG) like machine translation and natural language understanding (NLU) like the GLUE benchmarks [143]. Compared to RNNs that employ a recurrent structure to encode sequences, Transformers get rid of such sequential nature and utilize fully self-attention networks, thereby enabling better parallelization and largely improving efficiency. Moreover, they are more capable of capturing long-range dependencies by adding direct shortcut connections between any tokens in the sequence. Due to its strong representation learning and great efficiency, Transformers tend to be the new paradigm for encoding texts, and even any other types of sequential data, such as speech [31] or video [138].

Transformer Architecture

Figure 2.5 illustrates the overview of the Transformer architecture, which consists of a Transformer encoder and Transformer decoder. Formally, let $\mathbf{H}^l$ be a matrix with rows $\{\mathbf{h}_1^l, ..., \mathbf{h}_T^l\}$ corresponding to the intermediate representations after the l-layer. Multi-head attention is applied to compute each $\mathbf{h}_t^l$ from the $l-1$ layer's outputs and each head is defined as:

$$\mathbf{A}_i = \text{softmax}(\frac{\mathbf{Q}_i\mathbf{K}_i^T}{\sqrt{d_k}})\mathbf{V}_i, \tag{2.23}$$

$$\mathbf{Q}_i = h_t^{l-1}\mathbf{W}_i^Q, \mathbf{K}_i = h_t^{l-1}\mathbf{W}_i^K, \mathbf{V}_i = h_t^{l-1}\mathbf{W}_i^V, \tag{2.24}$$

where $\{\mathbf{Q}_i, \mathbf{K}_i, \mathbf{V}_i\}(i \in [1, m])$ is a set of queries, keys, and values for computing the i-head $\mathbf{A}_i \in d_k$ and m is the number of heads.

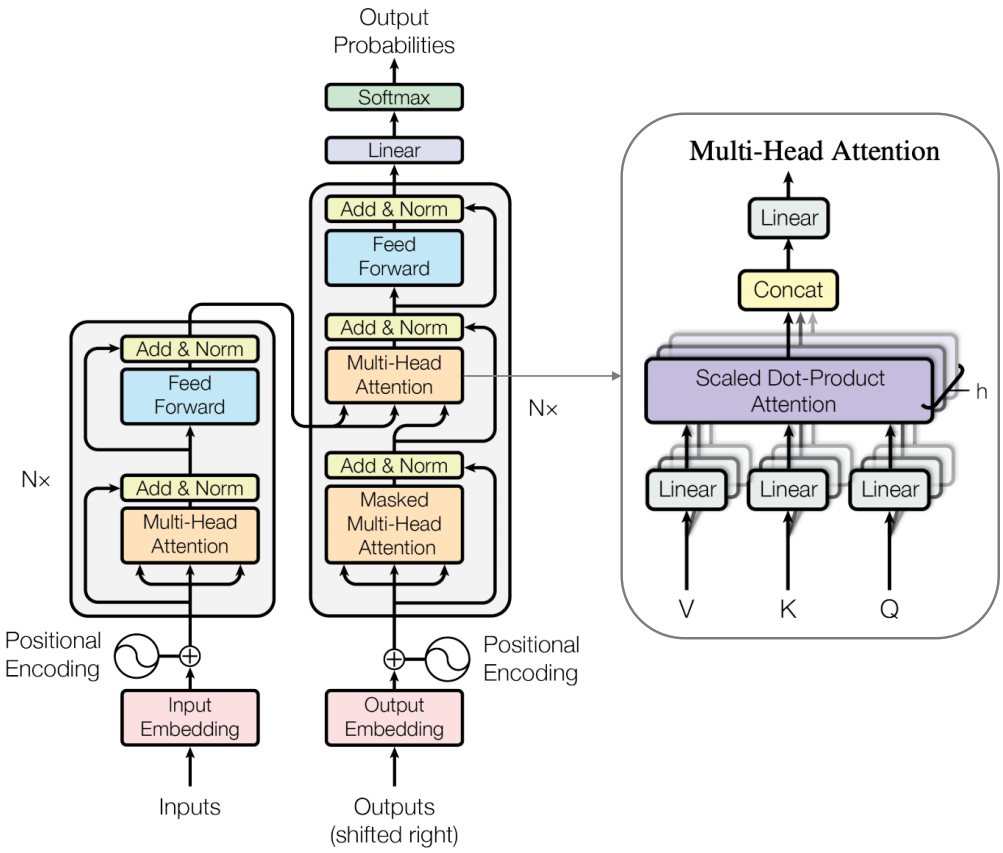


Figure 2.5: Illustration of Transformer and multi-head attention. Figure is from [141].

$\mathbf{W}_i^Q$, $\mathbf{W}_i^K$, and $\mathbf{W}_i^V$ are the trainable projection weights. Compared to traditional single-head attention, multi-head attention is able to attend information from various representation spaces and hence exhibits better representation learning capability.

Next, outputs from all the heads are concatenated and passed to a Feed-Forward Network (FFN) with residual connection [52],

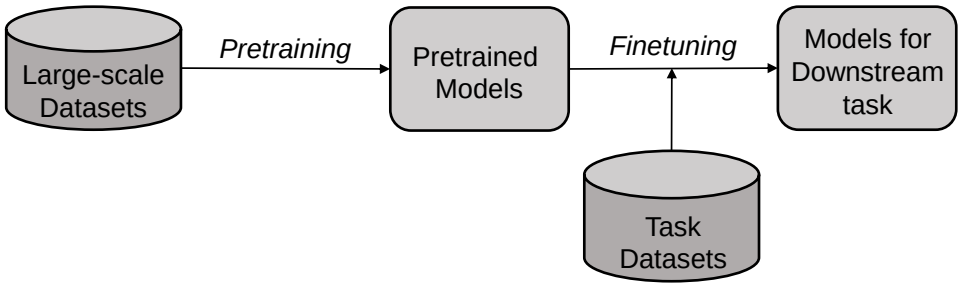


Figure 2.6: Illustration of pretrain-then-finetune paradigm.

followed by layer normalization [7]:

$$\mathbf{h}_t^l = \text{Concat}(\mathbf{A}_1, ..., \mathbf{A}_m)\mathbf{W}^O, \tag{2.25}$$

$$\mathbf{h}_t^l = \text{LayerNorm}(\mathbf{h}_t^{l-1} + \mathbf{h}_t^l), \tag{2.26}$$

$$\tilde{\mathbf{h}}_t^l = \max(0, \mathbf{h}_t^l\mathbf{W}_1 + \mathbf{b}_1)\mathbf{W}_2 + \mathbf{b}_2, \tag{2.27}$$

$$\mathbf{h}_t^l = \text{LayerNorm}(\tilde{\mathbf{h}}_t^l + \mathbf{h}_t^l), \tag{2.28}$$

where W^O is the projection weights to combine various head's outputs, and $\mathbf{W}_1, \mathbf{W}_2, \mathbf{b}_1, \mathbf{b}_2$ are trainable weights and biases in FFN layer. The outputs $\mathbf{H}^L$ at the final layer from the encoder will be based to make a discriminative prediction for NLU tasks or generate a target sequence with a decoder for NLG tasks.

Pretraining Tasks

To fully unleash the potential of Transformer models, it requires a large amount of data for sufficient training. As for some tasks with limited data, one can leverage the large-scale out-of-domain data for pretraining and then finetune it with small task datasets. Generally, pretraining aims to learn generic representations that can be transferred to downstream tasks. It can improve generalization especially when the target domain has scarce data. Such *pretrain-then-finetune* paradigm (Figure 2.6) has been ubiquitously applied in numerous applications in NLP,

CV, and also their intersection.

As it is very expensive to obtain huge annotated data for pretraining, researchers in NLP resort to applying unsupervised learning (or self-supervised learning) to derive generic representations from the abundant text data like Wikipedia or book corpus. Among them, BERT [35], short for bidirectional encoder representations from Transformers, is one of the most popular pretrained language models based on a multi-layer bidirectional Transformer. The BERT model is pretrained on a large language-corpus in an end-to-end fashion under two tasks: *masked language modelling* (MLM) and *next sentence prediction* (NSP).

In masked language modelling, tokens in x are randomly masked out with a probability of 15%. Each of the masked tokens will be replaced with (1) a special `[MASK]` token 80% of the time, (2) a random token 10% of the time, (3) the unchanged one 10% of the time. Next, the BERT model is teached to recover the masked tokens based on the observed set with cross entropy loss:

$$\mathcal{L}_{MLM}(\theta) = -E_{\mathbf{w}\sim D,\ t\sim T} \log P_\theta(w_t|\mathbf{w}_{\setminus t}), \tag{2.29}$$

where θ represents all the trainable parameters and $\mathbf{w}$ is sampled from the whole training set D. $\mathbf{w}_{\setminus t}$ is defined as $\{w_1, ..., w_{t-1}, \texttt{[MASK]}, w_{t+1}, ..., w_T\}$, and $P_\theta(w_t|\mathbf{w}_{\setminus t})$ is implemented by mapping h_t^L (the representation of w_t at the final Transformer layer) to a distribution over the vocabulary with a linear layer.

In next sentence prediction, a pair of sentences (`A`, `B`) are sampled from the input document D, and the model is trained to predict whether or not sentence `B` follows sentence `A` in the source text. Specifically, the two sentences are passed it into the

BERT following the format:

$$\{\texttt{[CLS]}, w_{\texttt{A}1}, ..., w_{\texttt{A}T}, \texttt{[SEP]}, w_{\texttt{B}1}, ..., w_{\texttt{B}T}, \texttt{[SEP]}\}.$$

A sigmoid classifier operating on the final output representation for the [CLS] token is trained to minimize a binary cross-entropy loss:

$$\begin{aligned}\mathcal{L}_{NSP}(\theta) = -E_{(\texttt{A},\texttt{B})\sim D}[y \log(S_\theta(\texttt{A},\texttt{B})) \\ + (1-y)log(1 - S_\theta(\texttt{A},\texttt{B}))],\end{aligned} \tag{2.30}$$

where $S_\theta(\texttt{A},\texttt{B})$ is the matching score of the sentences A and B from the classifier and $y \in \{0, 1\}$ indicates the relationship between the two sentences. Both positive and negative sentence pairs are sampled with the equivalent probability (i.e., 50%) to achieve a balanced label distribution.

Inspired by its success in NLP, recent work attempts to extend pretrained Transformer models to the vision and language domain. They employ similar pretraining tasks on a language-vision input and achieve prominent improvements in various visual and linguistic tasks, such as image/video captioning, question answering, cross-modal retrieval, etc.

2.2 Keyphrase Prediction

The goal of keyphrase prediction is to predict a set of concise keyphrases that summarize the main ideas of the input document. It can be considered as a special case of text summarization but with a different granularity. Formally, given an input document $\mathbf{x}$, keyphrase prediction aims to output a set of keyphrases $\mathcal{Y} = \{\mathbf{y}^{(1)}, \ldots, \mathbf{y}^{(|\mathcal{Y}|)}\}$, where $|\mathcal{Y}|$ is the

number of keyphrases and each keyphrase $\mathbf{y}^{(i)}$ is a phrase consisting of several words. The document $\mathbf{x}$ and a keyphrase $\mathbf{y}^{(i)}$ are formulated as word sequences $\mathbf{x} = \langle x_1, ..., x_{l_x} \rangle$ and $\mathbf{y} = \langle y_1^{(i)}, ..., y_{l_y}^{(i)} \rangle$, where l_x and l_y denote the number of words in $\mathbf{x}$ and $\mathbf{y}^{(i)}$ respectively.

Generally, keyphrase prediction methods can be divided into extraction-based methods, classification-based methods (specifically in social media), and generation-based methods. In the social media domain, hashtags that prefixed with "#" conveying the main topic are often regarded as the keyphrases for a post [176]. Apart from regarding each hashtag as a discrete label, one can employ segmentation rules to split it into several words, e.g., "DeepLearning" to "Deep" and "Learning".

2.2.1 Extraction-based Methods

Traditional keyphrase prediction studies mainly focus on handcrafted features, which select key words or phrases from the document as the predicted keyphrases. It typically adopts a two-step pipeline: candidates are first extracted with handcrafted features and then ranked by various scoring functions. At the candidate extraction step, these methods identify a set of keyphrase candidates based on handcrafted features, such as Part-of-Speech (POS) tags [90, 144, 76] and TF-IDF scores [99]. At the candidate ranking step, there are mainly two kinds of algorithms: unsupervised and supervised learning. Unsupervised learning algorithms are primarily built on a text graph, where they first regard each candidate as a node and then calculate its importance score in the graph [39, 156, 58, 107, 98, 91, 43]. As for supervised learning algorithms, they build a binary classifier to determine whether each candidate is a keyphrase, and then

the predicted scores are employed to rank the candidates. These methods also rely on handcrafted features such as phrase position and length, as well as the TF-IDF scores.
These methods undergo labor-intensive feature engineering and hence lead to the growing popularity of adopting fully data-driven methods using deep neural networks. Most efforts are based on sequence tagging style extraction [95, 44, 176] and combine the traditional two-step pipeline together into one step. Specifically, sequence tagging methods predict a label for each token in the document indicating whether it belongs to a keyphrase or not. Apart from the binary label, these methods often employ a more fine-grained categorization, i.e., (B) beginning of a keyphrase; (E) ending of keyphrase; (I) inside a keyphrase; (S) single-word keyphrase; or (O) otherwise. As for the social media domain, Zhang et al. [176] and Zhang et al. [179] also employ sequence tagging methods to extract keyphrases.

2.2.2 Classification-based Methods

Classification-based keyphrase prediction methods [54, 153, 126, 45, 57, 175] are mainly employed in the social media domain, where they usually regard each keyphrase as a discrete label and build classifier to predict it. Specifically, classification-based methods first construct a predefined candidate list and then select some of them based on the classifier's scores.
As for deep neural classifiers, Gong et al. [45] propose attention-based convolutional neural networks (CNNs) [77] consisting of a local attention channel and global channel to recommend hashtags. Huang et al. [57] employ end-to-end memory networks [137, 154] for this task, where they incorporate the histories of users into the external memory and leverage a hi-

erarchical attention mechanism to select more related histories. Recently, due to the wide availability of mobile devices and easy connectivity, multimedia contents are more prevalent in various social media platforms. To encoding more contexts, Zhang et al. [175] incorporate visual signals from the matching images in Twitter posts and employ co-attention networks [94, 163] to capture the text-image relationship.

2.2.3 Generation-based Methods

However, both extraction-based methods and classification-based methods have limitations in that they cannot produce keyphrases out of the source documents or the predefined candidate list. Inspired by the recent success of seq2seq learning in language generation tasks like text summarization, Meng et al. [100] first introduce sequence generation methods that predict keyphrase in a word-by-word manner for keyphrase prediction tasks. Generation-based methods overcome the drawbacks of the above two types of methods and enable new keyphrases beyond the source document or predefined list to be flexibly created. Most generation-based methods are proposed for predicting keyphrases from scientific articles.

As a pioneer work, Meng et al. [100] employ the pointer and generator framework [127] to either generate a word from the global vocabulary or copy it from the source sequence, which yields remarkable improvements over traditional extraction-based methods. Inspired by its success, a number of generation-based models [27, 24, 169, 26, 22, 171] have been proposed for this task. Chen et al. [27] propose a TG-Net that differentiates the importance of the title and the document, and explicitly makes use of the title to guide the understanding of the docu-

ment, while Chen et al. [24] propose a CorrRNN that applies the coverage mechanism in [127] to avoid generating repetitive keyphrases. Chen et al. [26] propose a hybrid approach that integrates keyphrase extraction, keyphrase retrieval, and keyphrase generation with a merging module and then returns the top-ranked candidates as the final predictions.

Some of them explore the keyphrase generation from some new perspectives. Ye et al. [169] investigate a different scenario where the amount of labeled data is limited and propose to leverage semi-supervised methods for improving the performance. Yuan et al. [171] consider to let the model itself determine the number of keyphrases that should be generated for a document. Along this line, Chan et al. [22] further introduce reinforcement learning (RL) to encourages a model to generate both sufficient and accurate keyphrases with an adaptive reward function.

In this thesis, we are the first to introduce sequence generation models to predict keyphrases for social media posts. Due to the informal nature of social media, the posts usually are short in length and contain lots of misspellings, making it difficult to process them effectively. To deal with such data sparsity in social media keyphrase generation, we propose to encode explicit contexts from user comments [150] or implicit latent topics that can be learned from a corpus in an unsupervised manner [148]. We further leverage the matching images to enrich the contexts and propose a unified model to couple the advantages of keyphrase classification and generation [149]. Similar to this, Chen et al. [26] also exploits the power of classification for keyphrase generation but in a separated retrieval manner, where we elegantly integrate them with a tailored copy mechanism and allow for the end-to-end joint training.

2.3 Social Media Understanding

The recent decades witness the flourish of social media, revolutionizing the ways people share information and interact with others. As a result, millions of user-generated data can be produced daily, leading us to the era of information explosion. To effectively process such a large volume of data and distill useful knowledge, social media understanding with automated techniques has received growing attention. In this section, we categorize current approaches for social media understanding into two groups based on the type of social media data: text-only and cross-media research.

2.3.1 Text-only Research

The abundance of user-generated texts fertilizes a broad set of real-world applications, such as microblog search [37, 10], sentiment analysis [34, 146], summarization [177, 23], user profiling [152, 38], stock price prediction [17, 108], and so forth. Among them, text classification and topic modeling are popular base approaches for language understanding proven to be useful in a variety of downstream tasks. Recently, with the success of seq2seq models for language generation, keyphrase prediction that summarizes a document into a set of concise keyphrases receives increasing attention due to its flexibility in creating multiple keyphrases in a large space. Hence, automatic keyphrase prediction serves as an important research topic for social media understanding.

Previous progress made in keyphrase prediction has mainly focused on either *extraction-based* or *classification-based* approaches, which are limited in that they cannot predict keyphrases

absent in the source text or the predefined candidate list. To overcome their shortcomings, in this thesis, we propose neural keyphrase generation models that enable new keyphrases to be flexibly created for social media posts. Although seq2seq-based generation models have demonstrated their effectiveness in keyphrase generation for scientific articles, their performance will be inevitably compromised when directly applied to noisy social media texts. The inferior performance is attributed to the severe data sparsity widely exhibited in short and informal social media posts. To deal with this, we propose to enrich contexts via either *implicitly* exploiting the post-level latent topics or *explicitly* leveraging conservation contexts from other users.

Our first work is also closely related to topic models that discover latent topics from word co-occurrence at the document level. They are commonly in the fashion of latent Dirichlet allocation (LDA) based on Bayesian graphical models [15]. These models, however, rely on the expertise involvement to customize model inference algorithms. Our framework exploits the recently proposed neural topic models [102, 134] to infer latent topics, which facilitate end-to-end training with other neural models and do not require model-specific derivation. It has proven useful for citation recommendation [9] and conversation understanding [173]. In particular, Zeng et al. [174] propose to jointly train topic models and short text classification, which cannot fit our scenario due to the large diversity of the keyphrases [150]. Different from them, our latent topics are learned together with language generation, whose effects on keyphrase generation have never been explored before in existing work.

2.3.2 Cross-media Research

"A picture is worth a thousand words". With the improved bandwidth and smartphones, cross-media posts are becoming ubiquitous as they can convey more diverse and complex information from the authors. For example, Twitter allows users to create tweets with multiple images and even videos. Some recent studies also find that tweets with images take up an increasing faction and receive significantly more engagement than tweets without images, approximately 22.8% more retweets, favorites, replies compared to text-only tweets [18]. However, cross-media posts bring more challenges for automatic understanding as they contain multi-modal features that involve complex interactions and require effective fusion.

In recent decades, numerous multi-modal machine learning techniques have been studied towards cross-media understanding, which covers a broad set of real-world applications, such as personalized image captioning [111], event extraction [87], sarcasm detection [19], possession extraction [29], and crisis event categorization [2]. Closest to our work, [175, 178] study multimedia hashtag classification and employ co-attention [94, 163] to model the text-image associations, where a single attention function is concurrently performed to infer either visual or textual distributions. We argue that they might be suboptimal to model intricate text-image associations, as a recent finding [142] points out there can be four diverse semantic relations held by images and texts on Twitter.

To allow for better modeling, in our fourth work, we take advantage of the recent advance of multi-head attention [141] capable of learning from different representation subspaces and extend it to capture diverse cross-media interactions. While

multi-head attention has been widely exploited in many vision-language (VL) tasks, such as image captioning [181], visual question answering [140, 92], and visual dialog [60], its potential benefit to model flexible cross-media posts has been previously ignored. Moreover, to well align the images' semantics to texts', we propose to encode *image wordings* and define two forms for that — explicit *optical characters* detected from the optical character reader (OCR) and implicit *image attributes* [157], high-level text labels predicted to summarize the image's semantic concepts. Some prior work has pointed out the usefulness of OCR texts [25] and image attributes [158] to endow images with higher-level semantics beyond visual features, where we are the first to study how OCR texts and image attributes work together to indicate keyphrases.

Cross-media research usually benefits from the development of more general multi-modal research, where conventional vision-language (VL) tasks like image captioning, visual question answering [6], and visual dialog [33] are extensively studied. Their core goal is to derive a generic visual and linguistic representation that achieves effective fusion from two modalities. Differently, cross-media studies can bring unique difficulties mainly due to the informal style in social media. For one thing, the text-image relationship in cross-media posts is rather complicated [142], while in conventional VL tasks the two modalities have most semantics shared. For another thing, social media images usually exhibit a more diverse distribution and a much higher probability of containing OCR tokens, thereby posing a hurdle for effectively processing. In the future, we will explore how to extend powerful visual and linguistic representation learning methods for improving cross-media understanding.

Chapter 3

Encoding Implicit Topics for Keyphrase Generation

This chapter presents our study in implicitly leveraging latent topics for social media keyphrase generation. Latent topics learned from a corpus via unsupervised methods like topic models can provide additional clues for understanding documents. While topic information has been proven useful in short text classification, we are the first to investigate its effects in keyphrase generation. The main points of this chapter are as follows. (1) We propose a topic-aware keyphrase generation model that incorporates corpus-level topics to enrich useful features for short social media posts. (2) Our topic-aware keyphrase generation model consists of a seq2seq model and a neural topic model that are elegantly integrated and jointly trained in an end-to-end manner. (3) We evaluate our models on three newly-constructed social media datasets from Twitter, Weibo, and StackExchange. The results show our model outperforms existing methods in keyphrase prediction, meanwhile generating more coherent topics.

Source post with keyphrase "*super bowl*":
[S]: Somewhere, a wife that is not paying attention to the *game*, says "I want the *team* in *yellow pants* to *win*."
Relevant tweets:
[T_1]: I been a *steelers fan* way before *black* & *yellow* and this *super bowl*!
[T_2]: I will bet you the *team* with *yellow pants wins*.
[T_3]: Wiz Khalifa song '*black* and *yellow*" to spur the *pittsburgh steelers* and Lil Wayne is to sing "*green* and *yellow*' for the *packers*.

Table 3.1: Sample tweets tagged with "*super bowl*" as their keyphrases. *Blue and italic words* can indicate the topic of super bowl.

3.1 Introduction

As social media continues its worldwide expansion, the last decade has witnessed the revolution of interpersonal communication. While empowering individuals with richer and fresher information, the flourish of social media also results in millions of posts generated on a daily basis. Facing a sheer quantity of texts, language understanding has become a daunting task for human beings. Under this circumstance, there exists a pressing need for developing automatic systems capable of absorbing massive social media texts and figuring out what is important.
In this work, we study the prediction of *keyphrases*, generally formed with words or phrases reflecting main topics conveyed in input texts [179]. Particularly, we focus on producing keyphrases for social media language, proven to be beneficial to a broad range of applications, such as instant detection of trending events [78], summarizing public opinions [101], analyzing social behavior [124], and so forth.
In spite of the substantial efforts made in social media keyphrase identification, most progress to date has focused on *extracting* words or phrases from source posts, thus failing to yield

keyphrases containing absent words (i.e., words do not appear in the post). Such cases are indeed prominent on social media, mostly attributed to the informal writing styles of users therein. For example, Table 3.1 shows a tweet S tagged with keyphrase "*super bowl*" by its author, though neither "*super*" nor "*bowl*" appears in it.[1] In our work, distinguishing from previous studies, we approach social media keyphrase prediction with a *sequence generation* framework, which is able to create absent keyphrases beyond source posts.

Our work is built on the success of deep keyphrase generation models based on neural sequence-to-sequence (seq2seq) framework [100]. However, existing models, though effective on well-edited documents (e.g., scientific articles), will inevitably encounter the data sparsity issue when adapted to social media. It is essentially due to the informal and colloquial nature of social media language, which results in limited features available in the noisy data. For instance, only given the words in S (Table 3.1), it is difficult to figure out why "*super bowl*" is its keyphrase. However, by looking at tweets T_1 to T_3, we can see "*yellow pants*" is relevant to "*steelers*", a *super bowl* team. As "*yellow*" and "*pants*" widely appear in tweets tagged with "*super bowl*', it becomes possible to identify "*super bowl*" as S's keyphrase.

Here we propose a novel *topic-aware neural keyphrase generation model* that leverages latent topics to enrich useful features. Our model is able to identify topic words, naturally indicative of keyphrases, via exploring post-level word co-occurrence patterns, such as "*yellow*" and "*pants*" in S. Previous work have shown that corpus-level latent topics can effectively alleviate

[1]Following common practice [176, 179], we consider author-annotated hashtags as tweets' keyphrases.

data sparsity in other tasks [174, 84]. The effects of latent topics, nevertheless, have never been explored in existing keyphrase generation research, particularly in the social media domain. To the best of our knowledge, *our work is the first to study the benefit of leveraging latent topics on social media keyphrase generation.* Also, our model, taking advantage of the recent advance of neural topic models [102], enables end-to-end training of latent topic modeling and keyphrase generation.

We experiment on three newly constructed social media datasets. Two are from English platform Twitter and StackExchange, and the other from Chinese microblog Weibo. The comparison results over both extraction and generation methods show that our model can better produce keyphrases, significantly outperforming all the comparison models without exploiting latent topics. For example, on Weibo dataset, our model achieves 34.99% F1@1 compared with 32.01% yielded by a state-of-the-art keyphrase generation model [100]. We also probe into our outputs and find that meaningful latent topics can be learned, which can usefully indicate keyphrases. At last, a preliminary study on scientific articles shows that latent topics work better on text genres with informal language style.

3.2 Topic-Aware Neural Keyphrase Generation Model

In this section, we describe our framework that leverages latent topics in neural keyphrase generation. Figure 3.1 shows our overall architecture consisting of two modules — a neural topic model for exploring latent topics (Section 3.2.1) and a seq2seq-based model for keyphrase generation (Section 3.2.2).

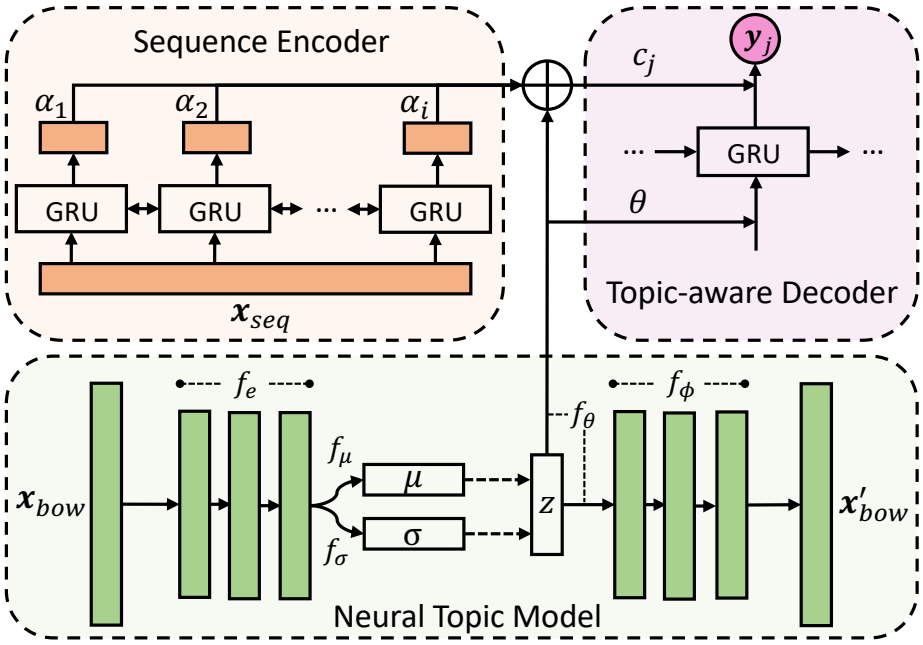


Figure 3.1: Our topic-aware neural keyphrase generation framework.

Before starting with more details, we first introduce the formulations of inputs. Formally, given a collection $\mathcal{C}$ with $|C|$ social media posts $\{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_{|C|}\}$ as input, we process each post $\mathbf{x}$ into bag-of-words (BoW) term vector $\mathbf{x}_{bow}$ and word index sequence vector $\mathbf{x}_{seq}$. $\mathbf{x}_{bow}$ is a V-dim vector over the vocabulary (V being the vocabulary size). It is fed into the neural topic model following the BoW assumption [102]. $\mathbf{x}_{seq}$ serves as the input for the seq2seq-based keyphrase generation model.

Below we first introduce our two modules and then describe how they are jointly trained.

3.2.1 Neural Topic Model

Our neural topic model (NTM) module is inspired by Miao et al. [102] based on variational auto-encoder [69], which consists

of an encoder and a decoder to resemble the data reconstruction process.
Specifically, the input $\mathbf{x}_{bow}$ is first encoded into a continuous latent variable $\mathbf{z}$ (representing $\mathbf{x}$'s topic) by a BoW encoder. Then the BoW decoder, conditioned on $\mathbf{z}$, attempts to reconstruct $\mathbf{x}$ and outputs a BoW vector $\mathbf{x}'_{bow}$. Particularly, the decoder simulates topic model's generation process. We then describe their division of labor.

BoW Encoder. The BoW encoder is responsible for estimating prior variables μ and σ, which will be used to induce intermediate topic representation $\mathbf{z}$. We adopt the following formula:

$$\mu = f_\mu(f_e(\mathbf{x}_{bow})),\ log\,\sigma = f_\sigma(f_e(\mathbf{x}_{bow})), \tag{3.1}$$

where $f_*(\cdot)$ is a neural perceptron with an ReLU-activated function following Zeng et al. [174].

BoW Decoder. Analogous to LDA-style topic models, it is assumed that there are K topics underlying the given corpus $\mathcal{C}$. Each topic k is represented with a topic-word distribution ϕ_k over the vocabulary, and each post $\mathbf{x} \in \mathcal{C}$ has a topic mixture denoted by θ, a K-dim distributional vector. Specifically in neural topic model, θ is constructed by Gaussian softmax [102]. The decoder hence takes the following steps to simulate how each post $\mathbf{x}$ is generated:

- Draw latent topic variable $\mathbf{z} \sim \mathcal{N}(\mu, \sigma^2)$
- Topic mixture $\theta = softmax(f_\theta(\mathbf{z}))$
- For each word $w \in \mathbf{x}$
  - Draw $w \sim softmax(f_\phi(\theta))$

Here $f_*(\cdot)$ is also a ReLU-activated neural perceptron for inputs. In particular, we employ the weight matrix of $f_\phi(\cdot)$ as the topic-word distributions $(\phi_1, \phi_2, ..., \phi_K)$. In the following, we adopt the topic mixture θ as the topic representations to guide keyphrase generation.

3.2.2 Neural Keyphrase Generation Model

Here we describe how we generate keyphrases with a topic-aware seq2seq model, which incorporates latent topics (learned by NTM) in its generation process. Below comes more details.

Overview. The keyphrase generation module (KG model) is fed with source post $\mathbf{x}$ in its word sequence form $\mathbf{x}_{seq} = \langle w_1, w_2, ..., w_{|\mathbf{x}|} \rangle$ ($|\mathbf{x}|$ is the number of words in $\mathbf{x}$). Its target is to output a word sequence $\mathbf{y}$ as $\mathbf{x}$'s keyphrase. Particularly, for a source post with multiple gold-standard keyphrases, we follow the practice in [100] to pair its copies with each of the gold standards to form a training instance.
To generate keyphrases for source posts, the KG model employs a seq2seq model. The *sequence encoder* distills indicative features from an input source post. The decoder then generates its keyphrase, conditioned on the encoded features and the latent topics yielded by NTM (henceforth *topic-aware sequence decoder*).

Sequence Encoder. We employ a bidirectional gated recurrent unit (Bi-GRU) [30] to encode the input source sequence. Each word $w_i \in \mathbf{x}_{seq}$ $(i = 1, 2, ..., |\mathbf{x}|)$ is first embedded into an embedding vector ν_i, and then mapped into forward and backward hidden states (denoted as $\overrightarrow{\mathbf{h}_i}$ and $\overleftarrow{\mathbf{h}_i}$) with the following defined

operations:

$$\overrightarrow{\mathbf{h}_i} = \text{GRU}(\nu_i, \mathbf{h}_{i-1}), \tag{3.2}$$

$$\overleftarrow{\mathbf{h}_i} = \text{GRU}(\nu_i, \mathbf{h}_{i+1}). \tag{3.3}$$

The concatenation of $\overrightarrow{\mathbf{h}_i}$ and $\overleftarrow{\mathbf{h}_i}$, $[\overrightarrow{\mathbf{h}_i}; \overleftarrow{\mathbf{h}_i}]$, serves as w_i's hidden state in encoder, denoted as $\mathbf{h}_i$. Finally, we construct a memory bank: $\mathbf{M} = \langle \mathbf{h}_1, \mathbf{h}_2, ..., \mathbf{h}_{|\mathbf{x}|} \rangle$, for decoder's attentive retrieval.

Topic-Aware Sequence Decoder. In general, conditioned on the memory bank $\mathbf{M}$ and latent topic θ from NTM, we define the process to generate its keyphrase $\mathbf{y}$ with the following probability:

$$Pr(\mathbf{y} \,|\, \mathbf{x}) = \prod_{j=1}^{|\mathbf{y}|} Pr(y_j \,|\, \mathbf{y}_{<j}, \mathbf{M}, \theta), \tag{3.4}$$

where $\mathbf{y}_{<j} = \langle y_1, y_2, ..., y_{j-1} \rangle$. And $Pr(y_j|\mathbf{y}_{<j}, \mathbf{M}, \theta)$, denoted as p_j, is a word distribution over vocabulary, reflecting how likely a word to fill in the j-th slot in target keyphrase. Below we describe the procedure to obtain p_j.

Our sequence decoder employs a unidirectional GRU layer. Apart from the general state update, the j-th hidden state $\mathbf{s}_j$ is further designed to take input $\mathbf{x}$'s topic mixture θ into consideration:

$$\mathbf{s}_j = \text{GRU}([\mathbf{u}_j; \theta], \mathbf{s}_{j-1}), \tag{3.5}$$

where $\mathbf{u}_j$ is the j-th embedded decoder input[2] and $\mathbf{s}_{j-1}$ is the previous hidden state. Here $[;]$ denotes the concatenation operation.

The decoder also looks at $\mathbf{M}$ (learned by sequence encoder) and puts an attention on it to capture important information. When

[2]We take the previous word from gold standards in training by teacher forcing and from the predicted word in test.

predicting the j-th word in keyphrase, the attention weights on $w_i \in \mathbf{x}_{seq}$ is defined as:

$$\alpha_{ij} = \frac{\exp(f_\alpha(\mathbf{h}_i, \mathbf{s}_j, \theta))}{\sum_{i'=1}^{|\mathbf{x}|} \exp(f_\alpha(\mathbf{h}_{i'}, \mathbf{s}_j, \theta))}, \tag{3.6}$$

where

$$f_\alpha(\mathbf{h}_i, \mathbf{s}_j, \theta) = \mathbf{v}_\alpha^T \ tanh(\mathbf{W}_\alpha[\mathbf{h}_i; \mathbf{s}_j; \theta] + \mathbf{b}_\alpha). \tag{3.7}$$

Here $\mathbf{v}_\alpha$, $\mathbf{W}_\alpha$, and $\mathbf{b}_\alpha$ are trainable parameters. $f_\alpha(\cdot)$ measures the semantic relations between the i-th word in the source and the j-th target word to be predicted. Such relations are also calibrated with the input's latent topic θ in order to explore and highlight topic words. We hence obtain the topic sensitive context vector $\mathbf{c}_j$ with:

$$\mathbf{c}_j = \sum_{i=1}^{|\mathbf{x}|} \alpha_{ij} \mathbf{h}_i. \tag{3.8}$$

Further, conditioned on $\mathbf{c}_j$, we generate the j-th word over the global vocabulary according to:

$$p_{gen} = softmax(\mathbf{W}_{gen}[\mathbf{s}_j; \mathbf{c}_j] + \mathbf{b}_{gen}). \tag{3.9}$$

In addition, we adopt copy mechanism [127] following Meng et al. [100], which allows keywords to be directly extracted from the source input. Specifically, we adopt a soft switcher $\lambda_j \in [0, 1]$ to determine whether to copy a word from source as the j-th target word:

$$\lambda_j = sigmoid(\mathbf{W}_\lambda[\mathbf{u}_j; \mathbf{s}_j; \mathbf{c}_j; \theta] + \mathbf{b}_\lambda), \tag{3.10}$$

with $\mathbf{W}_\lambda$ and $\mathbf{b}_\lambda$ being learnable parameters. Topic information θ is also injected here to guide the switch decision.

Finally, we obtain distribution p_j for predicting the j-th target word with the formula below:

$$p_j = \lambda_j \cdot p_{gen} + (1 - \lambda_j) \cdot \sum_{i=1}^{|\mathbf{x}|} \alpha_{ij}, \tag{3.11}$$

where attention scores $\{\alpha_{ij}\}_{i=1}^{|\mathbf{x}|}$ serve as the extractive distribution over the source input.

3.2.3 Jointly Learning Topics and Keyphrases

Our neural framework allows end-to-end learning of latent topic modeling and keyphrase generation. We first define objective functions for the two modules respectively.
For NTM, the objective function is defined based on negative variational lower bound [14]. Here due to space limitation, we omit the derivation details already described in [102], and directly give its loss function:

$$\mathcal{L}_{NTM} = D_{KL}(p(\mathbf{z}) \,||\, q(\mathbf{z} \,|\, \mathbf{x})) - \mathbb{E}_{q(\mathbf{z} \,|\, \mathbf{x})}[p(\mathbf{x} \,|\, \mathbf{z})], \tag{3.12}$$

where the first term is the Kullback-Leibler divergence loss and the second term reflects the reconstruction loss. $p(\mathbf{z})$ denotes a standard normal prior. $q(\mathbf{z} \,|\, \mathbf{x})$ and $p(\mathbf{x} \,|\, \mathbf{z})$ represent the process of BoW encoder and BoW decoder respectively.
For KG model, we minimize the cross entropy loss over all training instances:

$$\mathcal{L}_{KG} = -\sum_{n=1}^{N} \log(Pr(\mathbf{y}_n \,|\, \mathbf{x}_n, \theta_n)), \tag{3.13}$$

where N denotes the number of training instances and θ_n is $\mathbf{x}_n$'s latent topics induced from NTM.

Finally, we define the entire framework's training objective with the linear combination of $\mathcal{L}_{NTM}$ and $\mathcal{L}_{KG}$:

$$\mathcal{L} = \mathcal{L}_{NTM} + \gamma \cdot \mathcal{L}_{KG}, \tag{3.14}$$

where the hyper-parameter γ balances the effects of NTM and KG model. Our two modules can be jointly trained with their parameters updated simultaneously. For inference, we adopt beam search and generate a ranking list of output keyphrases following Meng et al. [100].

3.3 Experimental Setup

3.3.1 Datasets

We conduct experiments on three social media datasets collected from two English online platforms, Twitter and StackExchange, and a Chinese microblog website, Weibo. Twitter and Weibo are microblogs encouraging users to freely post with a wide range of topics, while StackExchange, an online Q&A forum, are mainly for question asking (with a title and a description) and seeking answers from others.

The Twitter dataset contains tweets from TREC 2011 microblog track.[3] For Weibo dataset, we first tracked the real-time trending hashtags in Jan-Aug 2014,[4] and then used them as keywords to search posts with hashtag-search API.[5] And the StackExchange dataset is randomly sampled from a publicly available raw corpus.[6]

[3] http://trec.nist.gov/data/tweets/
[4] http://open.weibo.com/wiki/Trends/
[5] http://www.open.weibo.com/wiki/2/
[6] https://archive.org/details/stackexchange

For the target keyphrases, we employ user-annotated hashtags for Twitter and Weibo following Zhang et al. [176], and author-assigned tags (e.g., "*artificial-intelligence*") for StackExchange. Posts without such keyphrase tags are hence removed from the datasets. Particularly, for StackExchange, we concatenate the question title together with its description as the source input. For Twitter and Weibo source posts, we retain tokens in hashtags (without # symbols) for those appearing in the middle of posts, since they generally act as semantic elements and thus considered as present keyphrases [176]. For those appearing before or after a post, we remove the entire hashtags and regard them as absent keyphrases as is done in [150].

For model training and evaluation, we split the data into three subsets with 80%, 10%, and 10%, corresponding to training, development, and test set. The statistics of the three datasets are shown in Table 3.2. As can be seen, over 50% of the keyphrases do not appear in their source posts, thus extractive approaches will fail in dealing with these posts. We also observe that StackExchange exhibits different keyphrase statistics compared to either Twitter or Weibo, with more keyphrases appearing in one post and more diverse keyphrases.

3.3.2 Preprocessing

For Twitter dataset, we employed Twitter preprocessing toolkit in [11] for source post and hashtag (keyphrase) tokenization. Chinese Weibo data was preprocessed with Jieba toolkit[7] for word segmentation, and English StackExchange data with natural language toolkit (NLTK) for tokenization.[8]

[7] https://github.com/fxsjy/jieba

[8] https://www.nltk.org/

Source posts	# of posts	Avg len per post	# of KP per post	Source vocab
Twitter	44,113	19.52	1.13	34,010
Weibo	46,296	33.07	1.06	98,310
StackExchange	49,447	87.94	2.43	99,775
Target KP	|KP|	Avg len per KP	% of abs KP	Target vocab
Twitter	4,347	1.92	71.35	4,171
Weibo	2,136	2.55	75.74	2,833
StackExchange	12,114	1.41	54.32	10,852

Table 3.2: Data statistics of source posts (on the top) and target keyphrases (on the bottom). Avg len: the average number of tokens. KP: keyphrases. Abs KP: absent keyphrases. |KP|: the number of distinct keyphrases.

We further take the following preprocessing steps for each of the three datasets: First, posts with meaningless keyphrases (e.g., single-character ones) were filtered out; also removed were non-alphabetic (for English data) and retweet-only (e.g., "*RT*") posts. Second, links, mentions (@username), and digits were replaced with generic tags "*URL*", "*MENT*", and "*DIGIT*" following Wang et al. [150]. Third, a vocabulary was maintained, with 30K most frequent words for Twitter, and 50K for Weibo and StackExchange each. For BoW vocabulary of the input $\mathbf{x}_{bow}$ for NTM, stop words and punctuation were removed.

3.3.3 Model Settings

We implement our model based on the pytorch framework in [113]. For NTM, we implement it following the design[9] in [174] and set topic number K to 50. The KG model is set up mostly based on [100]. For its sequence encoder, we adopt two layers of

[9] `https://github.com/zengjichuan/TMN`

bidirectional GRU and one layer of unidirectional GRU for its decoder. The hidden size of the GRU is 300 (for bi-GRU, 150 for each direction). For the embedding, its size is set to 150 and values are randomly initialized. We apply Adam [68] with initial learning rate as $1e-3$. In training, gradient clipping = 1.0 is conducted to stabilize the training. Early-stopping strategy [21] is adopted based on the validation loss. Before joint training, we pretrain NTM for 100 epochs and KG model for 1 epoch as the convergence speed of NTM is much slower than the KG model. We empirically set the $\gamma = 1.0$ for balancing NTM and KG loss (Eq. (3.14)) and iteratively update the parameters in each module and then their combination in turn.

3.3.4 Comparisons

In comparison, we first consider a simple baseline selecting majority keyphrases (henceforth MAJORITY) — the top K keyphrases ranked by their frequency in training data are used as the keyphrases for all test instances. We also compare with the following extractive baselines, where n-grams ($n = 1, 2, 3$) in source posts are ranked by TF-IDF scores (henceforth TF-IDF), TextRank algorighm [103] (henceforth TEXTRANK), and KEA system [156] (henceforth KEA). We also compare with a neural state-of-the-art keyphrase extraction model based on sequence tagging [176] (henceforth SEQ-TAG). In addition, we take the following state-of-the-art keyphrase generation models into consideration: seq2seq model with copy mechanism [100] (henceforth SEQ2SEQ-COPY) and its variation SEQ2SEQ without copy mechanism, SEQ2SEQ-CORR [24] exploiting keyphrase correlations, and TG-NET [27] jointly modeling of titles and descriptions (thereby only tested on StackExchange).

3.4 Results and Analysis

In the experiment, we first evaluate our performance on keyphrase prediction in Section 3.4.1. Then, we study whether jointly learning keyphrase generation can in turn help produce coherent topics in Section 3.4.2. At last, further discussions are presented with an ablation study, a case study, and an analysis for varying text genres.

3.4.1 Keyphrase Prediction Results

In this subsection, we examine our performance in predicting keyphrases for social media. We first discuss the main comparison results, followed by a discussion for present and absent keyphrases.

Popular information retrieval metrics macro-average F1@K and mean average precision (MAP) are adopted for evaluation. Here for Twitter and Weibo, most posts are tagged with one keyphrase on average (Table 3.2), thus F1@1 and F1@3 are reported. For StackExchange, we report F1@3 and F1@5, because on average, posts have 2.4 keyphrases. MAP is measured over the top 5 predictions for all three datasets. For keyphrase matching, we consider keyphases after stemmed by Porter Stemmer following Meng et al. [100].

Model	Twitter			Weibo			StackExchange		
	F1@1	F1@3	MAP	F1@1	F1@3	MAP	F1@3	F1@5	MAP
Baselines									
MAJORITY	9.36	11.85	15.22	4.16	3.31	5.47	1.79	1.89	1.59
TF-IDF	1.16	1.14	1.89	1.90	1.51	2.46	13.50	12.74	12.61
TEXTRANK	1.73	1.94	1.89	0.18	0.49	0.57	6.03	8.28	4.76
KEA	0.50	0.56	0.50	0.20	0.20	0.20	15.80	15.23	14.25
State of the arts									
SEQ-TAG	22.79±0.3	12.27±0.2	22.44±0.3	16.34±0.2	8.99±0.1	16.53±0.3	17.58±1.6	12.82±1.2	19.03±1.3
SEQ2SEQ	34.10±0.5	26.01±0.3	41.11±0.3	28.17±1.7	20.59±0.9	34.19±1.7	22.99±0.3	20.65±0.2	23.95±0.3
SEQ2SEQ-COPY	36.60±1.1	26.79±0.5	43.12±1.2	32.01±0.3	22.69±0.2	38.01±0.1	31.53±0.1	27.41±0.2	33.45±0.1
SEQ2SEQ-CORR	34.97±0.8	26.13±0.4	41.64±0.5	31.64±0.7	22.24±0.5	37.47±0.8	30.89±0.3	26.97±0.2	32.87±0.6
TG-NET	-	-	-	-	-	-	32.02±0.3	27.84±0.3	34.05±0.4
Our model	**38.49**±0.3	**27.84**±0.0	**45.12**±0.2	**34.99**±0.3	**24.42**±0.2	**41.29**±0.4	**33.41**±0.2	**29.16**±0.1	**35.52**±0.1

Table 3.3: Main comparison results displayed with average scores (in %) and their standard deviations over the results with 5 sets of random initialization seeds. Boldface scores in each column indicate the best results. Our model significantly outperforms all comparisons on all three datasets ($p < 0.05$, paired t-test).

Main Comparison Discussion. Table 3.3 shows the main comparison results on our three datasets, where higher scores indicate better performance. From all three datasets, we observe:

- *Social media keyphrase prediction is challenging.* As can be seen, all simple baselines give poor performance. This indicates that predicting keyphrases for social media language is a challenging task. It is impossible to rely on simple statistics or rules to yield good results.
- *Seq2seq-based keyphrase generation models are effective.* Compared to the extractive baselines and SEQ-TAG, seq2seq-based models perform much better. It is because social media's informal language style results in a large amount of absent keyphrases (Table 3.2), which is impossible for extractive methods to make correct predictions. We also find SEQ2SEQ-COPY better than SEQ2SEQ, suggesting the effectiveness to combine source word extraction with word generation when predicting keyphrases.
- *Latent topics are consistently helpful for indicating keyphrases.* It is observed that our model achieves the best results, significantly outperforming all comparisons by a large margin. This shows the usefulness of leveraging latent topics in keyphrase prediction. Interestingly, compared with StackExchange, we achieve larger improvements for Twitter and Weibo, both exhibiting more informal nature and prominent word order misuse. For such text genres, latent topics, learned under BoW assumption, are more helpful.

Also, the following interesting points can be observed by comparing results across datasets:

- *Keyphrase generation is more challenging for StackExchange.* When MAP scores of seq2seq-based methods are compared over the three datasets, we find that the scores on StackExchange are generally lower. It is probably attributed to the data characteristics of more diverse keyphrases and larger target vocabulary (Table 3.2).

- *Twitter and Weibo data is noisier.* We notice that TF-IDF, TextRank, and KEA perform much worse than Majority, while the opposite is observed on StackExchange. It is because Twitter and Weibo, as microblogs, contain shorter posts (Table 3.2) and exhibit more informal language styles. In general, models relying on simple word statistics would suffer from such noisy data.

Present and Absent Keyphrase Prediction. We further discuss how our model performs in producing present and absent keyphrases. The comparison results with all neural-based models are shown in Figure 3.2. Here F1@1 is adopted for evaluating the prediction of present keyphrases and recall@5 for absent keyphrases.

The results indicate that our model consistently outperforms comparison models in predicting either absent or present keyphrases. Also, interestingly, copy mechanism seems to somehow sacrifice the performance on absent keyphrase generation for correctly extracting the present ones. Such side effects, however, are not observed on our model. It is probably attributed to our ability to associate posts with corpus-level topics, hence enabling absent keywords from other posts to be "copied". This observation also demonstrates the latent topics can help our model to better decide whether to copy (Eq. (3.10)).

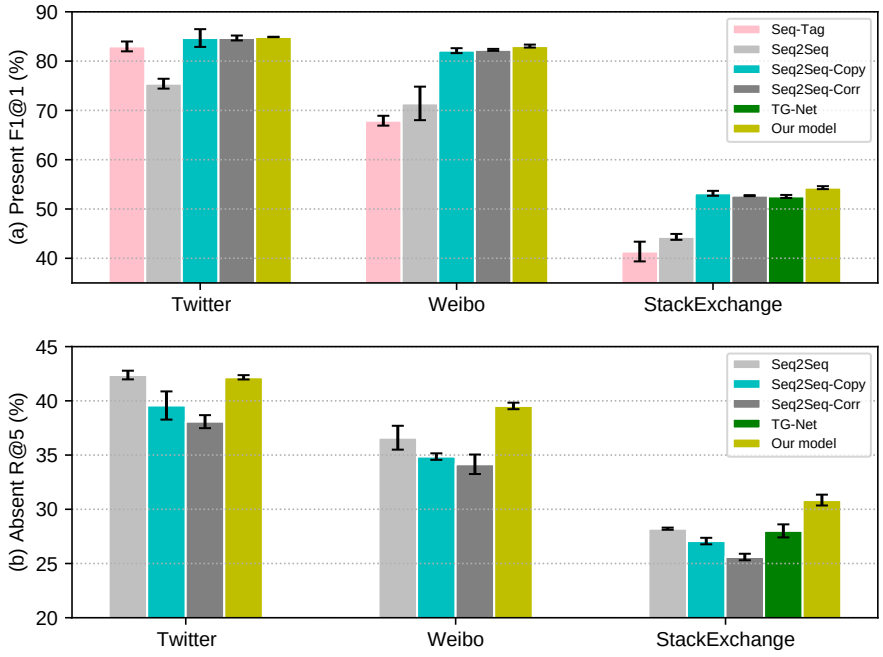


Figure 3.2: The prediction results for present (on the top) and absent keyphrases (on the bottom, R@5: recall@5). For present cases, from left to right shows the results of SEQ-TAG, SEQ2SEQ, SEQ2SEQ-COPY, SEQ2SEQ-CORR, TG-NET (only for StackExchange), and our model. For absent cases, models (except SEQ-TAG) are shown in the same order.

3.4.2 Latent Topic Analysis

We have shown latent topics useful for social media keyphrase generation above. Here we analyze whether our model can learn meaningful topics.

Coherence Score Comparison. We first evaluate topic coherence with an automatic C_V measure. Here we employ Palmetto toolkit[10] [121] on the top 10 words from each latent topic following Zeng et al. [174]. The results are only reported on English Twitter and StackExchange because Palmetto does not support

[10] https://github.com/dice-group/Palmetto/

Datasets	Twitter	StackExchange
LDA	41.12	35.13
BTM	43.12	43.52
NTM	43.82	43.04
Our model	**46.28**	**45.12**

Table 3.4: C_V topic coherence score comparison on our two English datasets. Higher scores indicate better coherence. Our model produces the best scores.

LDA	bowl super quote steeler jan watching egypt playing glee girl
BTM	bowl super anthem national christina aguilera fail word brand playing
NTM	super bowl eye protester winning watch halftime ship sport mena
Our model	bowl super yellow green packer steeler nom commercial win winner

Table 3.5: Top 10 terms for latent topics "*super bowl*". Red and underlined words indicate non-topic words.

Chinese. For comparisons, we consider LDA (implemented with a gensim LdaMulticore package[11]), BTM[12] [165] (a state-of-the-art topic model specifically for short texts), and NTM [102]. For LDA and BTM, we run Gibbs sampling with $1,000$ iterations to ensure convergence. From the results in Table 3.4, we observe that our model outperforms all the comparison topic models by large margins, which implies that jointly exploring keyphrase generation can in turn help produce coherent topics.

Sample Topics. To further evaluate whether our model can produce coherent topics qualitatively, we probe into some sample words (Table 3.5) reflecting the topic "*super bowl*" discovered

[11] `https://pypi.org/project/gensim/`
[12] `https://github.com/xiaohuiyan/BTM`

Model	Twitter	Weibo	SE
SEQ2SEQ-COPY	36.60	32.01	31.53
Our model (*separate train*)	36.75	32.75	31.78
Our model (*w/o topic-attn*)	37.24	32.42	32.34
Our model (*w/o topic-state*)	37.44	33.48	31.98
Our full model	**38.49**	**34.99**	**33.41**

Table 3.6: Comparison results of our ablation models on three datasets (SE: StackExchange) — *separate train*: our model with pretrained latent topics; *w/o topic-attn*: decoder attention without topics (Eq. (3.7)); *w/o topic-state*: decoder hidden states without topics (Eq. (3.5)). We report F1@1 for Twitter and Weibo, F1@3 for StackExchange. Best results are in bold.

by various models from Twitter. As can be seen, there are mixed non-topic words [13] in LDA's, BTM's, and NTM's sample topic. Compared with them, our inferred topic looks more coherent. For example, "*steeler*" and "*packer*", names of *super bowl* teams, are correctly included into the cluster.

3.4.3 Ablation Study

We compare the results of our full model and its four ablated variants to analyze the relative contributions of topics on different components. The results in Table 3.6 indicate the competitive effect of topics on decoder attention and that on hidden states, but combining them both help our full model achieve the best performance. We also observe that pretrained topics only bring a small boost, indicated by the close scores yielded by our model (*separate train*) and SEQ2SEQ-COPY. This suggests that the joint training is crucial to better absorb latent topics.

[13] Non-topic words refer to words that cannot clearly indicate the corresponding topic, including off-topic words more likely to reflect other topics.

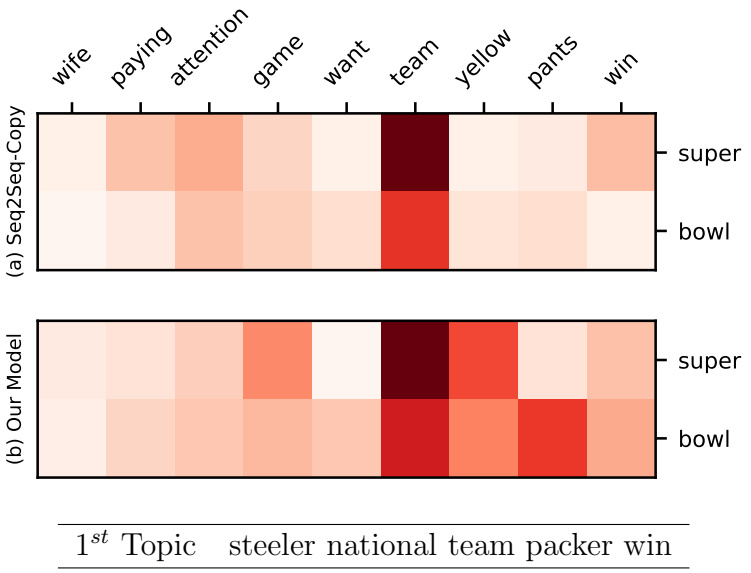


Figure 3.3: Attention visualization for the sample post in Table 3.1. Only non-stopwords are selected. The table below shows the top five words for the 1^{st} topic.

3.4.4 Case Study

We feed the tweet S in Table 3.1 into both SEQ2SEQ-COPY and our model. Eventually our model correctly predicts the keyphrase as "*super bowl*" while SEQ2SEQ-COPY gives a wrong prediction "*team follow back*" (posted to ask other to follow back). To analyze the reason behind, we visualize the attention weights of two models in Figure 3.3. It can be seen that both models highlight the common word "*team*", which frequently appears in "*team follow back*"-tagged tweets. By joint modeling of latent topics, our model additionally emphasizes topic words "*yellow*" and "*pants*", which are signals indicating a super bowl team *steeler* (also reflected in the 1^{st} topic) and thus helpful to correctly generate "*super bowl*" as its keyphrase. Without such topic guidance, SEQ2SEQ-COPY wrongly predicts a common but

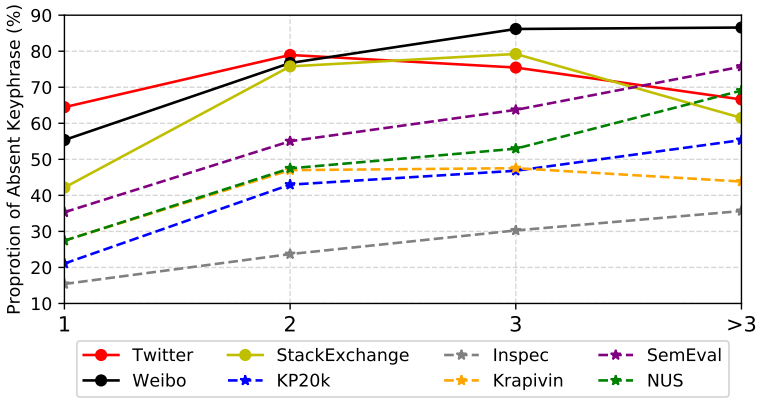


Figure 3.4: Proportion of absent n-gram keyphrases (n: $1, 2, 3, > 3$). The dashed lines with '*' marks: the five scientific article datasets used in [100].

unrelated term "*team follow back*".

3.4.5 Topic-Aware KG for Other Text Genres

We have shown the effectiveness of latent topics on social media keyphrase generation. To examine how they affect in identifying keyphrases for well-edited language, we also experiment on the traditional scientific article datasets [100], but limited improvements are observed. Latent topics can better help keyphrase generation on social media, probably because there are larger proportion of keyphrases with absent words (Figure 3.4), where latent topics can cluster relevant posts and enrich the source contexts. Another possible reason lies in that social media language exhibits prominent arbitrary word orders. Thus latent topics, learned under BoW assumption, can better provide useful auxiliary features.

3.5 Summary

In this chapter, we have presented a novel topic-aware keyphrase generation model for social media language. Unlike prior methods based on extraction or classification, our keyphrase generation model can create new keyphrases that do not appear in the source post. In order to alleviate the data sparsity issue in social media, we exploit the corpus-level latent topics to enrich features, thereby benefiting the keyphrase prediction. Particularly, our model allows the joint learning of latent topic representations in an end-to-end manner. Experimental results on three newly constructed social media datasets show that our model significantly outperforms state-of-the-art methods in keyphrase prediction, meanwhile producing more coherent topics. Further analysis interprets our superiority to discover key information from noisy social media data. We release our code and datasets to benefit future research on text analysis and topic modeling in social media.

Chapter 4

Encoding Explicit Conversation for Keyphrase Generation

Social media platforms like microblogging services allow users to form conversations on issues of interests by replying to target posts for voicing their opinions. Such conversation contexts can enrich the limited features conveyed from the short target posts and thus are useful for identifying their key ideas. This chapter explores how to improve keyphrase generation by explicitly encoding conversation contexts for social media posts. The main points of this chapter are as follows. (1) Unlike most prior work relying on classification-based methods, we employ a sequence generation model that can generate rare and even unseen keyphrases. (2) We propose to leverage the user conversation with a bi-attention mechanism to model its interactions with the target post. (3) Experimental results on English Twitter and Chinese Weibo datasets validate our model's superiority over traditional classification methods.

4.1 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 topics [166, 55, 83]. Following the common practice in [176, 179], we regard hashtags as keyphrases for a social media post. By tagging keyphrases for social media posts, it can further benefit downstream applications, such as microblog search [37, 10], summarization [177, 23], sentiment analysis [34, 146], and so forth. Despite the widespread use of keyphrases, there are a significant fraction of microblog messages without any user-provided keyphrases. For example, less than 15% tweets contain at least one hashtag [146, 64]. Consequently, for a multitude of posts without human-annotated hashtags, there exists a pressing need for automating the keyphrase annotation process for them. Most previous work in this field focuses on extracting phrases from target posts [176, 179] or selecting candidates from a predefined list [45, 57, 175]. However, keyphrases 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 keyphrases they like. For another, due to the wide range and rapid change of social media topics, a vast variety of keyphrases can be daily created, making

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 *NadalTomic* 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 4.1: A post and its conversation snippet about "Australian Open" on Twitter. "*#AusOpen*" is the human-annotated keyphrase for the target post. *Words indicative of the keyphrase* are in blue and italic type.

it impossible to be covered by a fixed candidate list. Prior research from another line employs topic models to generate topic words as keyphrases [46, 176]. These methods, ascribed to the limitation of most topic models, are nevertheless incapable of producing phrase-level keyphrases.

In this work, we approach keyphrase annotation from a novel *sequence generation* framework. In doing so, we enable phrase-level keyphrases beyond the target posts or the given candidates to be created. Here, keyphrases 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 [139], we present a neural seq2seq model to generate keyphrases in a *word-by-word* manner. To the best of our knowledge, *we are the first to deal with keyphrase 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.[1] To illustrate such challenge, Table 4.1 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 keyphrases 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 [23, 81]. The effects of conversation contexts, useful on topic modeling [83, 85] and keyphrase extraction [179], have never been explored on microblog keyphrase generation. To show why conversation contexts are useful, we display in Table 4.1 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 keyphrase "*#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 a bidirectional attention (**bi-attention**) mechanism [129] to explore the interactions between two encoders' outputs. Afterward, an attentive decoder

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

is applied to generate the word sequence of the keyphrase.
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 keyphrases 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 keyphrases.
To summarize, our contributions are three-fold:

- We are the first to approach microblog keyphrase 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.

4.2 Conv-aware Neural Keyphrase Generation Model

In this section, we describe our framework shown in Figure 4.1, which is a conv-aware (short for conversation-aware) keyphrase generation model. There are two major modules: a dual encoder

to encode both target posts and their conversations with a bi-attention module to explore their interactions, and a decoder to generate keyphrases.

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 keyphrase $\mathbf{y}$ represented by a word sequence $\langle y_1, y_2, ..., y_{|\mathbf{y}|} \rangle$. For training instances tagged with multiple gold-standard keyphrases, we copy the instances multiple times, each with one gold-standard keyphrase following Meng et al. [100]. All the input target posts, conversations, and keyphrases share the same vocabulary V.

4.2.1 Post-Conversation 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) [30] 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:

$$\overrightarrow{\mathbf{h}_i^p} = \text{GRU}(e(\mathbf{x}_i^p), \overrightarrow{\mathbf{h}_{i-1}^p}), \tag{4.1}$$

$$\overleftarrow{\mathbf{h}_i^p} = \text{GRU}(e(\mathbf{x}_i^p), \overleftarrow{\mathbf{h}_{i+1}^p}). \tag{4.2}$$

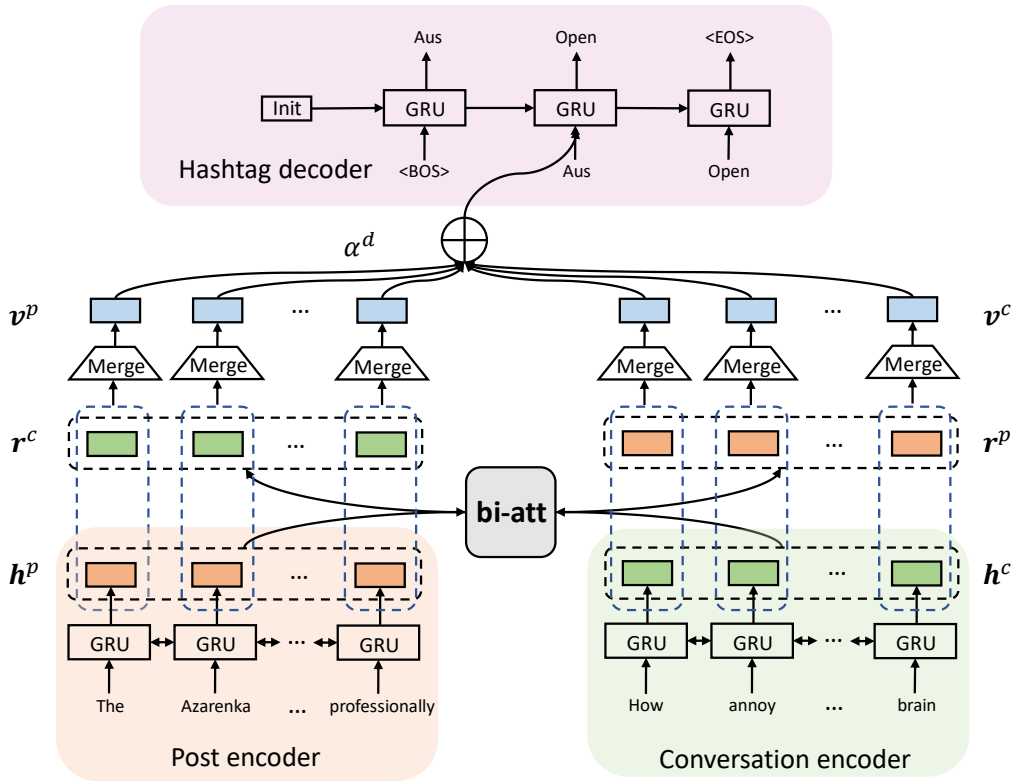


Figure 4.1: Our Conv-aware keyphrase generation framework with a dual encoder, including a post encoder and a conversation encoder, where a bi-attention (bi-att) module distills their salient features, followed by a merge layer to fuse them. An attentive decoder generates the keyphrase sequence.

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 module to explore the interactions between the target posts and their conversations. The adoption of bi-attention mechanism is inspired by Seo et al. [129], 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 keyphrases should be generated. For instance, in Table 4.1, 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 keyphrase is "*#AusOpen*".

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

$$\alpha_{ij}^{c} = \frac{\exp(f_{score}(\mathbf{h}_i^p, \mathbf{h}_j^c))}{\sum_{j'=1}^{|\mathbf{x}^c|} \exp(f_{score}(\mathbf{h}_i^p, \mathbf{h}_{j'}^c))}, \tag{4.3}$$

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:

$$\mathbf{r}_i^c = \sum_{j=1}^{|\mathbf{x}^c|} \alpha_{ij}^c \mathbf{h}_j^c. \tag{4.4}$$

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 module on each encoder, we design a *merge* layer, a multilayer perceptron (MLP) activated by hyperbolic

function:

$$\mathbf{v}^p = \tanh(\mathbf{W}_p[\mathbf{h}^p; \mathbf{r}^c] + \mathbf{b}_p), \tag{4.5}$$

$$\mathbf{v}^c = \tanh(\mathbf{W}_c[\mathbf{h}^c; \mathbf{r}^p] + \mathbf{b}_c), \tag{4.6}$$

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 keyphrase generation.

4.2.2 Sequence 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 keyphrase. The probability to generate the keyphrase conditioned on a target post and its conversation is defined as:

$$Pr(\mathbf{y}|\mathbf{x}^p, \mathbf{x}^c) = \prod_{t=1}^{|\mathbf{y}|} Pr(y_t|\mathbf{y}_{<t}, \mathbf{x}^p, \mathbf{x}^c), \tag{4.7}$$

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

Concretely, when generating the t-th word in keyphrase, 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:

$$\mathbf{c}_t = \sum_{i=1}^{|\mathbf{x}^p|+|\mathbf{x}^c|} \alpha_{ti}^d \mathbf{v}_i, \tag{4.8}$$

$$\alpha_{ti}^{d} = \frac{\exp(g_{score}(\mathbf{s}_t, \mathbf{v}_i))}{\sum_{i'=1}^{|\mathbf{x}^p|+|\mathbf{x}^c|} \exp(g_{score}(\mathbf{s}_t, \mathbf{v}_{i'})}, \tag{4.9}$$

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:

$$Pr(y_t|\mathbf{y}_{<t}, \mathbf{x}^p, \mathbf{x}^c) = softmax(\mathbf{W}_v[\mathbf{s}_t; \mathbf{c}_t] + \mathbf{b}_v), \tag{4.10}$$

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

4.2.3 Learning and Inferring Keyphrases

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

$$\mathcal{L}(\Theta) = -\sum_{n=1}^{N} \log(Pr(\mathbf{y}_n|\mathbf{x}_n^p, \mathbf{x}_n^c; \Theta)). \tag{4.11}$$

Here N is the number of training instances and Θ denotes the set of all the learnable parameters.
In keyphrase 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 keyphrases, where the top K keyphrases serve as our final output.

4.3 Experimental Setup

Here we describe how we set up our experiments.

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

4.3.1 Datasets

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.[2] 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[3] as keywords. For gold-standard keyphrases, we take the user-annotated keyphrases, appearing before or after a post, as the reference.[4] The statistics of our datasets are shown in Table 4.2. 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 keyphrases in Table 4.3 and the distributions of keyphrase frequency in Figure 4.2. In Table 4.3, we observe the large size of keyphrases in both datasets. Moreover, Figure 4.2 indicates that most keyphrases only appear

[2] `https://trec.nist.gov/data/tweets/`

[3] `http://open.weibo.com/wiki/Trends/`

[4] keyphrases in the middle of a post are not considered here as they generally act as semantic elements [176, 179].

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

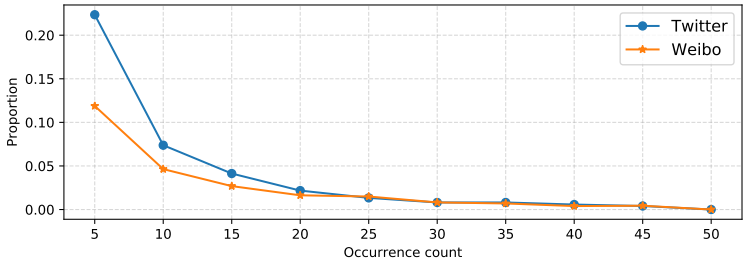


Figure 4.2: Distribution of keyphrase frequency. The horizontal axis refers to the occurrence count of keyphrases (shown with maximum 50 and bin 5) and the vertical axis denotes the data proportion.

a few times. Given such a large and imbalanced keyphrase space, keyphrase selection from a candidate list, as many existing methods do, might not perform well. Table 4.3 also shows that only a small proportion of keyphrases appearing in their posts, conversations, and either of them, making it inappropriate to directly extract words from the two sources to form keyphrases.

4.3.2 Preprocessing

For tokenization and word segmentation, we employed the tweet preprocessing toolkit [11] for Twitter, and the Jieba toolkit[5] for Weibo. Then, for both Twitter and Weibo, we further take the following preprocessing steps: First, single-character keyphrases were filtered out for not being meaningful. Second, generic

[5]`https://pypi.python.org/pypi/jieba/`

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.

4.3.3 Comparisons

For experiment comparisons, we first consider a weak baseline RANDOM that randomly ranks keyphrases 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. [179], we extract phrases from target posts as keyphrases via sequence tagging and encode conversations with memory networks [137].
- CLASSIFIER: We compare with the state-of-the-art model based on classification [45], where keyphrases 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) [139] is applied to generate keyphrases given a target post.

We also consider its variant augmented with copy mechanism [48] (henceforth SEQ2SEQ-COPY), which has proven effective in keyphrase generation [100] 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.

4.3.4 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 [70]. 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 [68] 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 keyphrase to 10.
For CLASSIFIER and EXTRACTOR, lacking publicly available codes, we reimplement the models using Keras.[6] Their results are reproduced in their original experiment settings. For LDA, we employ an open source toolkit lda.[7]

[6] https://keras.io/
[7] https://pypi.org/project/lda/

Evaluation Metrics. Popular *information retrival* evaluation metrics F1 scores at K (F1@K) and mean average precision (MAP) scores [97] 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 keyphrase as a sequence of words, ROUGE metrics for *summarization* evaluation [88] are also adopted. Here, we use ROUGE F1 for the top-ranked keyphrase prediction computed by an open source toolkit pythonrouge,[8] with Porter stemmer used for English tweets. For Weibo posts, scores calculated at the Chinese character level following Li et al. [85]. We report the average scores for multiple gold-standard keyphrases on ROUGE evaluation.

4.4 Results and Analysis

In this section, we first report the main comparison results in Section 4.4.1, followed by an in-depth comparative study between classification and sequence generation models in Section 4.4.2. Further discussions are then presented to analyze our superiority and errors.

[8] `https://github.com/tagucci/pythonrouge`

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 4.4: 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.

4.4.1 Main Comparison Results

Table 4.4 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 keyphrases and thus no results are reported for F1@5 and MAP. For LDA, as it cannot generate bigram keyphrases, no results are presented for ROUGE-SU4. In general, we have the following observations:

- *keyphrase 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 keyphrases (e.g., abbreviations are prominent like "*aus*" in "*#AusOpen*"). For statistical reasons, Twitter keyphrases are more likely to be absent in either posts or conversations (Table 4.3), and have a more severe imbalanced distribution (Figure 4.2).

- *Topic models and extractive models are ineffective for keyphrase annotation.* The poor performance of all baseline models indicates that keyphrase annotation is a challenging problem. LDA sometimes performs even worse than RANDOM due to its inability to produce phrase-level keyphrases. For extractive models, both TF-IDF and EXTRACTOR fail to achieve good results. It is because most keyphrases are absent in target posts, as we see in Table 4.3 that only 2.72% keyphrases 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

keyphrase 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 keyphrases 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 keyphrase space (reflected in Table 4.3 and Figure 4.2).

- *Conversations are useful for keyphrase 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 keyphrase 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 keyphrase, which is however not possible for classification models (considering keyphrases to be inseparable).

4.4.2 Classification vs. Generation

From Table 4.4, we observe that the classifiers outperform topic models and extractive models by a large margin but exhibit

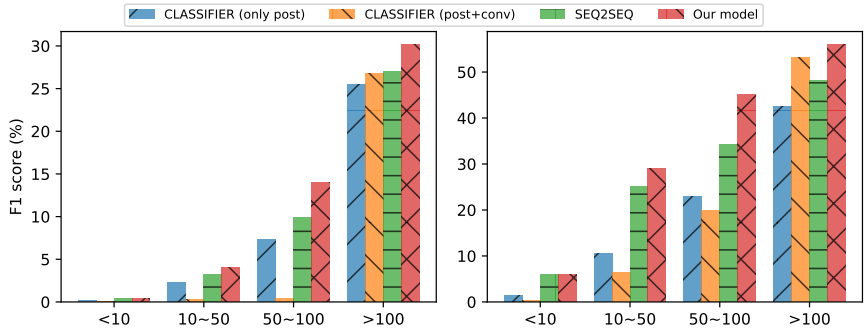


Figure 4.3: F1@1 on Twitter (the left subfigure) and Weibo (the right subfigure) in inferring keyphrases with varying frequency. In each subfigure, from left to right shows the results of CLASSIFIER (*post only*), CLASSIFIER (*post+conv*), SEQ2SEQ, and OUR MODEL. Generation models consistently perform better.

generally worse results than sequence generation models. Here, we present a thorough study to compare keyphrase 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 keyphrases.

Rare keyphrases. According to the keyphrase distributions in Figure 4.2, we can see a large proportion of keyphrases appearing only a few times in the data. To study how models perform to predict such keyphrases, in Figure 4.3, we display their F1@1 scores in inferring keyphrases with varying frequency. The lower F1 score on less frequent keyphrases indicates the difficulty to yield rare keyphrases. 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 keyphrases with varying sparsity

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 4.5: ROUGE-1 F1 scores (%) in producing unseen keyphrases. Best results are in bold.

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 keyphrase generation, which enables the internal structure of keyphrases (how words form a keyphrase) to be exploited.

New keyphrases. To further explore the extreme situation where keyphrases are absent in the training set, we experiment to see how models perform in handling new keyphrases. To this end, we additionally collect instances tagged with keyphrases 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 4.5.

As can be seen, creating unseen keyphrases 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 keyphrase annotation on Twitter is more difficult

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 4.6: F1@1 scores (%) for our variants. Best results are in **bold**.

due to the noisier data characteristics. In particular, compared to SEQ2SEQ, OUR MODEL achieves an additional performance gain in producing new keyphrases by leveraging conversations with the bi-attention module.

4.4.3 Ablation Study

We report the ablation study results in Table 4.6 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 keyphrases 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. (4.5) and Eq. (4.6), 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	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 4.7: Model outputs for the target post in Table 4.1. "*aus open*" matches the gold-standard keyphrase.

4.4.4 Case Study

We further present a case study on the target post shown in Table 4.1, where the top five outputs of some comparison models are displayed in Table 4.7. 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 keyphrases, e.g., sport-related terms like "*bbc football*", "*arsenal*", and "*murray*". On the contrary, CLASSIFIER and SEQ2SEQ tend to yield frequent keyphrases, 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 keyphrase is however absent.

To analyze why our model obtains superior results in this case, we display the heatmap in Figure 4.4 to visualize our bi-attention weight matrix $\mathbf{W}_{bi-att}$. As we can see, the bi-attention mechanism 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,

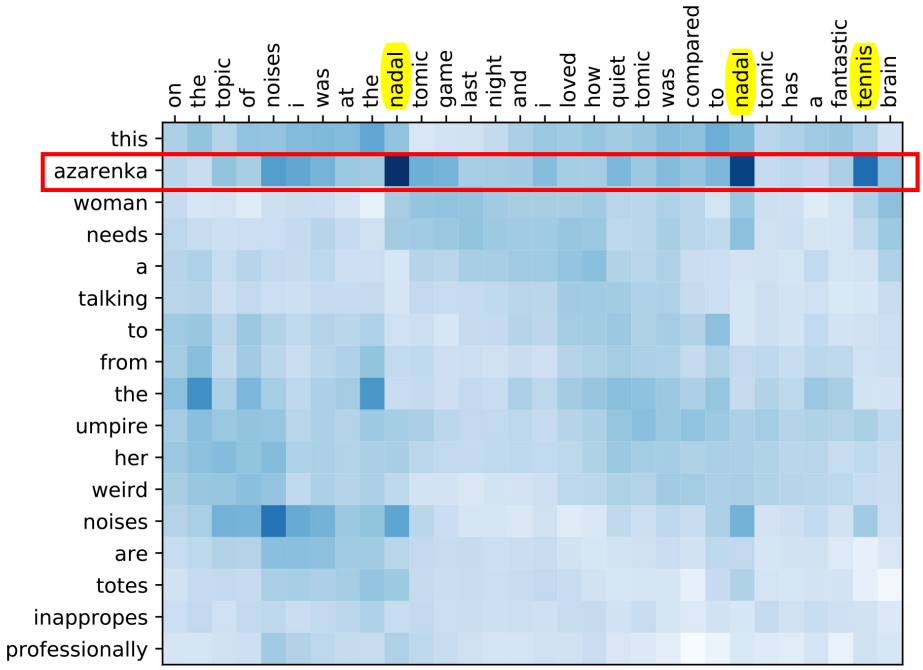


Figure 4.4: Visualization of the bi-attention module given the input case in Table 4.1. The horizontal axis denotes a snippet of a truncated conversation. The vertical axis shows the target post. Salient words are highlighted.

facilitating the correct keyphrase "*aus open*" to be generated.

4.4.5 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 keyphrase evaluation. Another primary type of error is caused by the non-topic keyphrases, such as "*#fb*" (indicating the messages forwarded from Facebook). Such non-topic keyphrases cannot reflect any content information from

target posts and should be distinguished from topic keyphrases in the future.

4.5 Summary

In this chapter, we have presented a novel framework of keyphrase 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 keyphrases.

Chapter 5

Cross-Media Keyphrase Prediction: A Unified Framework with Multi-Modality Multi-Head Attention and Image Wordings

With the advent of mobile Internet, more and more social media posts contain images to convey more diverse and complex information from the authors. Such images can provide complementary knowledge to the target post and thus should be exploited for better cross-media understanding. This chapter investigates the combined effects of texts and images for indicating keyphrases for a multimedia post. The main points of this chapter are as follows. (1) We propose to exploit image wordings to bridge the text-image semantic gap and design a novel M3H-Att to capture the dense interactions between them better. (2) We propose a unified framework to integrate the outputs of keyphrase classification and generation and couple their advantages. (3) Experiments on a text-image Twitter dataset demonstrate the effectiveness of our model.

Post (a): Contemplating the mysteries of life from inside my egg carton...☺
#cat #cats #CatsOfTwitter

Post (b): The <*mention*> have the slight lead at halftime!
#NBAFinals

Figure 5.1: Two multimedia posts from Twitter, where texts offer limited help in identifying their keyphrases while images provide essential clues.

5.1 Introduction

The prominent use of social media platforms (such as Twitter) exposes individuals with an abundance of fresh information in a wide variety of forms such as texts, images, videos, etc. Meanwhile, the explosive growth of multimedia data has far outpaced individuals' capability to understand them, presenting a concrete challenge to digest massive amounts of data, distill the salient contents therein, and provide users with quick access to the information they need when navigating noisy online data. To that end, extensive efforts have been made to *social media keyphrase prediction* — aiming to produce a sequence of words that reflect a post's key concern. Nevertheless, previous work mostly focuses on the use of textual signals [179, 148, 150], which sometimes provide limited features as social media language is

essentially informal and fragmented. To enrich the contexts, here we resort to exploiting the matching images, which are widely used in social media posts to deliver auxiliary information from authors (e.g., opinions, feelings, topics, etc.), primarily due to the flourish of mobile Internet.

To illustrate our motivation, Figure 5.1 shows the texts and images of two Twitter posts (tweets). The left is tagged with a keyphrase "cat", which can be clearly signaled with its image while the paired text is an anthropomorphic description and hardly unveils its real semantics. For the right, the image depicts a basketball game scene with optical characters "2019 NBA FINALS", directly indicating its keyphrase, which is difficult to identify from the texts. In both examples, images play a more vital role in reflecting the key information. These points motivate our cross-media keyphrase prediction study that examines how the salient contents can be indicated by the coupled effects of post texts and their matching images.

Previous work [175, 178] employs co-attention networks [94, 163] to encode multimedia posts, where a single attention function is concurrently performed to infer either visual or textual distributions. We argue that they might be suboptimal to model intricate text-image associations, as a recent finding [142] points out there can be four diverse semantic relations held by images and texts on Twitter. To allow for better modeling, in this work, we take advantage of the recent advance of multi-head attention [141] capable of learning from different representation subspaces and extend it to capture diverse cross-media interactions, named as *Multi-Modality Multi-Head Attention* (M^3H-Att). Moreover, to well align the images' semantics to texts', we adopt *image wordings* and define two forms for that — explicit

optical characters (such as "NBA Finals" in post (b)) detected from the optical character reader (OCR) and implicit *image attributes* [157], high-level text labels predicted to summarize the image's semantic concepts (such as a "cat" label for post (a)).

Furthermore, unlike prior work employing either classification [45] or generation models [148], we propose a *unified* framework to couple the advantages of keyphrase classification and generation. Specifically, in addition to the joint training of both modules, we further extend the copy mechanism [127] to explicitly aggregate classification outputs together with tokens from the source input. Empirical results show that integrating classification outputs not only keeps classification's superiority to predict common keyphrases (Figure 5.7(c)) while enables keyphrase creation beyond a predefined candidate list, but also largely benefits the keyphrase generation with better absent keyphrase prediction (Figure 5.7(b)).

For experiments, we collect large-scale tweets with texts and images, which is presented as part of our work. The empirical results show that our model significantly outperforms the state-of-the-art (SOTA) methods using traditional attention mechanisms. For example, we obtain 47.06% F1@1 compared with 43.17% by [148] (keyphrase generation from texts only) and 42.12% by [175] (multi-modal keyphrase classification). We then examine how we perform to handle absent and present keyphrases, and varying keyphrase frequency and post length. The results indicate the consistent performance boost brought by our M^{3}H-Att design in diverse scenarios and the significant benefit to absent keyphrase prediction offered from our unified framework (Section. 5.4.2). We further quantify the effects of

different settings of multi-head attention and image wordings to see when and how they work the best (Section. 5.4.3). A qualitative analysis is given at last to interpret why our model results in superior multimedia understanding (Section. 5.4.4). In summary, our contributions are three-fold:

- We extensively study the joint effects of texts and images for social media keyphrase prediction and present a large-scale Twitter dataset for that.
- A novel design of Multi-Modality Multi-Head Attention (M^3H-Att) and image wordings are proposed to effectively capture dense interactions between texts and images in social media styles.
- To the best of our knowledge, we are the first to propose a unified framework coupling classification and generation models for keyphrase prediction, which shows promising empirical results.

5.2 Unified Cross-Media Keyphrase Prediction Model

Given a collection $\mathcal{C}$ with $|C|$ text-image post pairs $\{(\mathbf{x}^n, I^n)\}_{n=1}^{|C|}$ as input, we aim to predict a keyphrase set $\mathcal{Y} = \{\mathbf{y}^i\}_{i=1}^{|\mathcal{Y}|}$ for each of them. Following Meng et al. [100], we copy the source input pair multiple times to allow each paired to have one keyphrase. We represent each input as a triplet $(\mathbf{x}, I, \mathbf{y})$, where $\mathbf{x}$ and $\mathbf{y}$ are formulated as word sequences $\mathbf{x} = \langle x_1, ..., x_{l_x} \rangle$ and $\mathbf{y} = \langle y_1, ..., y_{l_y} \rangle$ (l_x and l_y denote the number of words).
We show the overview of our proposed cross-media keyphrase prediction model in Figure 5.2. We first encode a text-image

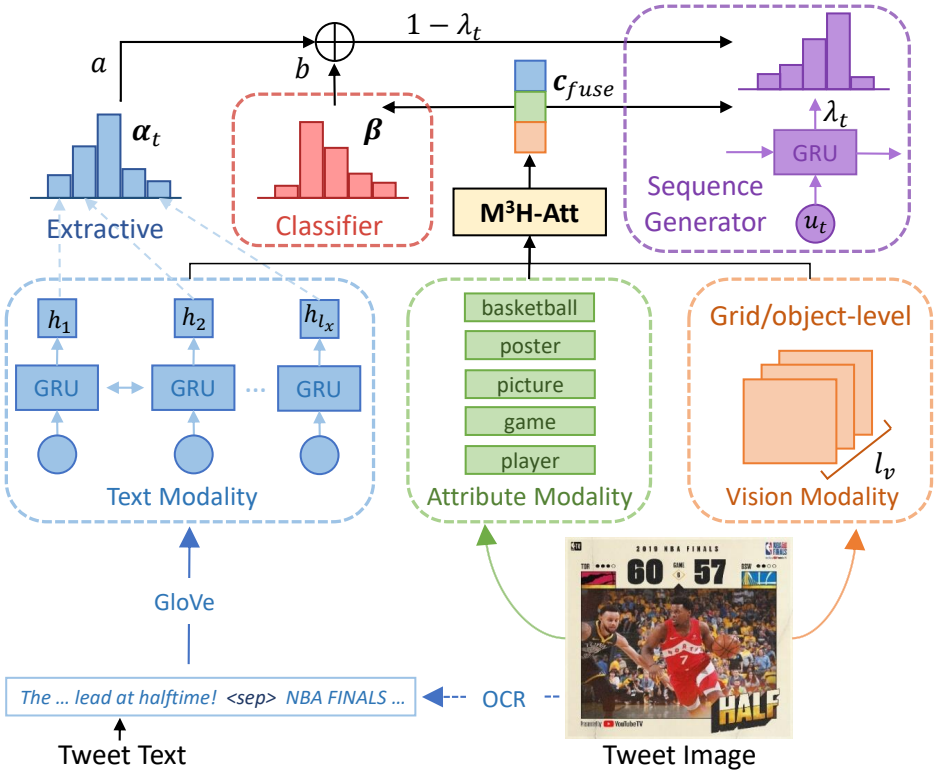


Figure 5.2: The overview of our unified cross-media keyphrase prediction model. Work flow: (1) a text-image post is encoded into text, attribute, and vision modalities; (2) the encoded features are fused with M^{3}H-Att; (3) the output of a keyphrase classifier and generator are aggregated for a unified prediction.

tweet into three modalities: *text*, *attribute*, and *vision* (Section 5.2.1), and propose a Multi-Modality Multi-Head Attention (M^{3}H-Att) to capture their intricate interactions (Section 5.2.2). Then, we feed the learned multi-modality representations for either keyphrase classification or generation. At last, a tailored aggregator is devised to combine their outputs (Section 5.2.3) and the entire framework can be jointly trained in an end-to-end manner (Section 5.2.4).

5.2.1 Multi-modality Encoder

Learning Text Representation. We first embed each token x_i from the input sequence into a high-dimensional vector via a pretrained lookup table, and then employ bidirectional gated recurrent unit (Bi-GRU) [30] to encode the embedded input token $e(x_i)$:

$$\overrightarrow{\mathbf{h}_i} = \text{GRU}(e(x_i), \quad \overrightarrow{\mathbf{h}_{i-1}}), \tag{5.1}$$

$$\overleftarrow{\mathbf{h}_i} = \text{GRU}(e(x_i), \quad \overleftarrow{\mathbf{h}_{i+1}}). \tag{5.2}$$

Forward hidden state $\overrightarrow{\mathbf{h}_i}$ and backward one $\overleftarrow{\mathbf{h}_i}$ are later concatenated into $\mathbf{h}_i = [\overrightarrow{\mathbf{h}_i}; \overleftarrow{\mathbf{h}_i}]$. We employ it as the context-aware representation of x_i and pack all of them in the input sequence into a textual memory bank $\mathbf{M}_{text} = \{\mathbf{h}_i, ..., \mathbf{h}_{l_x}\} \in \mathbb{R}^{l_x \times d}$, where d denotes the hidden state dimension.

Encoding OCR Text. To detect optical characters from images, we use an open-source toolkit [133] to extract OCR texts in form of a word sequence. It is then appended into the post text with a delimited token $\langle sep \rangle$ to notify the change of text genres, which is shown to be a simple yet effective design to combine OCR features (Table 5.4).

Learning Image Representation. We consider two types of image representations: *grid-level* or *object-level* visual features. For the former, we apply a pretrained VGG-16 Net [132] to extract 7×7 convolutional feature maps for each image I. For the latter, inspired by bottom-up attention [5], we use the Faster-RCNN [120] pretrained on Visual Genome [72] to detect the objects and extract their features. Each feature map is further

transformed into a new vector $\mathbf{v}_i$ through a linear projection layer. As such, we construct a visual memory bank as $\mathbf{M}_{vis} = \{\mathbf{v}_1, ..., \mathbf{v}_{l_v}\} \in \mathbb{R}^{l_v \times d}$, where l_v denotes the number of image regions or objects.

Encoding Image Attribute. Following Cai et al. [19], we first train an attribute predictor based on the Resnet-152 [53] features on Microsoft COCO 2014 caption dataset [89]. Specifically, we extract noun and adjective tokens from the image captions as the attribute labels. Afterwards, the top five attributes of each image are mapped with another linear layer to produce the attribute memory bank $\mathbf{M}_{attr} = \{\mathbf{a}_1, ..., \mathbf{a}_5\} \in \mathbb{R}^{5 \times d}$, which aims to capture images' high-level semantic concepts.

5.2.2 Multi-modality Multi-Head Attention

Our design of multi-head attention is inspired by its prototype in Transformer [141]. We extend it to capture multiple forms of cross-modality interactions for a multimedia post, which is therefore named as M^{3}H-Att, short for Multi-Modality Multi-Head Attention. The three modalities (text, attribute, and vision) are modeled in a *pairwise* co-attention manner, compared with its original form as a self-attention over texts only.
For each co-attention, we perform scaled dot attention $\mathcal{A}$ on a set of {*Query*, *Key*, *Value*}:

$$\mathcal{A}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{softmax}(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{d_K}})\mathbf{V}, \tag{5.3}$$

$$\mathcal{A}^{\text{Multi-head}}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = [head_1; ...; head_H]\mathbf{W}^O, \tag{5.4}$$

$$\text{where} \quad head_h = \mathcal{A}(\mathbf{Q}\mathbf{W}_h^Q, \mathbf{K}\mathbf{W}_h^K, \mathbf{V}\mathbf{W}_h^V). \tag{5.5}$$

$\mathbf{W}_h^Q, \mathbf{W}_h^K, \mathbf{W}_h^V \in \mathbb{R}^{d \times d_H}$ are learnable weights to project the query, key, and value from dimension d to a lower space of d_H-dimension and H is the head number. Outputs from all the heads are concatenated in $\mathcal{A}^{\text{Multi-head}}$ and passed to a feedforward network with residual connections [53] and layer normalization [7].

Specifically, we employ the text features as a query to attend to the vision/attribute modality and vice versa.[1] Here max/average-pooling is adopted to obtain one holistic query vector for each modality instead of token-level queries considering the noisy nature of social media data. Moreover, we stack multiple co-attention layers to empower its modeling capability, where $L_{text}, L_{attr}, L_{vis}$ denote the number of stacked layers for text, attribute, and vision queries. After that, the outputs from all co-attention layers are summed up with a linear multi-modal fusion layer to produce a context vector $\mathbf{c}_{fuse} \in \mathbb{R}^d$. It will be fed into a keyphrase classifier and generator for the unified prediction. Notably, this indicates that our M^{3}H-Att's great potential to serve as a generic module for benefiting other cross-media applications.

5.2.3 Unified Keyphrase Prediction

We describe how we combine the keyphrase classification and generation for the unified prediction.

Keyphrase Classification. As each keyphrase $\mathbf{y}$ usually consists of only several tokens, it can be considered as a discrete integral label and predicted it with a keyphrase classifier. Here we

[1] We also try other combinations, e.g., M^{3}H-Att between the vision and attribute, but the improvements are negligible.

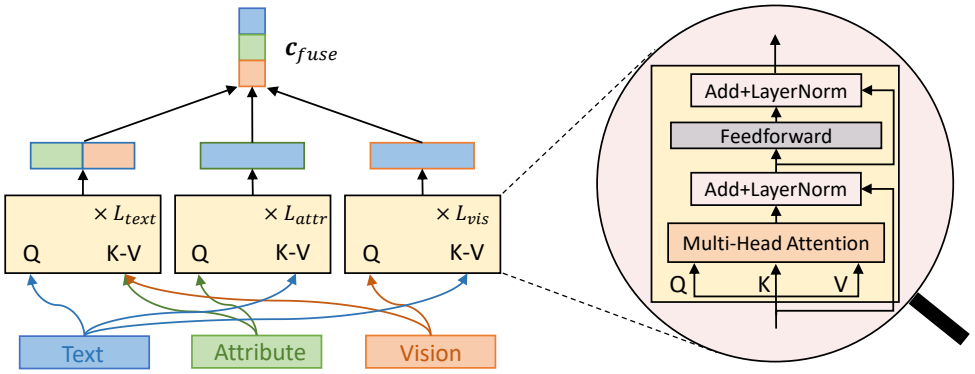


Figure 5.3: Overview of M^{3}H-Att to fuse multi-modal features from text, attribute, and vision modalities.

directly pass the multi-modal context vector $\mathbf{c}_{fuse}$ into a two-layer of multi-layer perceptron (MLP) and map it to $\hat{\mathbf{y}}$ in the label vocabulary space V_{cls}:

$$P_{cls}(\mathbf{y}) = \text{softmax}(\text{MLP}_{cls}(\mathbf{c}_{fuse})). \tag{5.6}$$

Keyphrase Generation with Pointer. For keyphrase generation, we base on a sequence-to-sequence framework to predict the keyphrase word sequence $\mathbf{y} = \langle y_1, ..., y_{l_y} \rangle$, where the generation probability is defined as $\prod_{t=1}^{l_y} P(y_t \,|\, \mathbf{y}_{<t})$.
Concretely, we use an unidirectional GRU decoder to model the generation process, which emits the hidden state $\mathbf{s}_t = \text{GRU}(\mathbf{s}_{t-1}, \mathbf{u}_t) \in \mathbb{R}^d$ based on the previous hidden state $\mathbf{s}_{t-1}$ and the embedded decoder input $\mathbf{u}_t$. The decoder state is initialized by the last hidden state $\mathbf{h}_{l_x}$ of the text encoder. Here an attention mechanism [8] is adopted to obtain a textual context

$\mathbf{c}_{text}$:

$$\mathbf{c}_{text} = \sum_{i=1}^{l_x} \alpha_{t,i}\mathbf{h}_i, \tag{5.7}$$

$$\alpha_{t,i} = \text{softmax}(S(\mathbf{s}_t, \mathbf{h}_i)), \tag{5.8}$$

$$S(\mathbf{s}_t, \mathbf{h}_i) = \mathbf{v}_\alpha^T \tanh(\mathbf{W}_\alpha[\mathbf{s}_t; \mathbf{h}_i] + \mathbf{b}_\alpha), \tag{5.9}$$

where $S(\mathbf{s}_t, \mathbf{h}_i)$ is a score function to measure the compatibility between the t-th word to be decoded and the i-th word from the text encoder. $\mathbf{W}_\alpha \in \mathbb{R}^{d\times 2d}, \mathbf{b}_\alpha, \mathbf{v} \in \mathbb{R}^d$ are trainable weights.
Next, we incorporate the static multi-modal vector $\mathbf{c}_{fuse}$ (produced by M^{3}H-Att and independent of the decoder state) to construct a context-rich representation $\mathbf{c}_t = [\mathbf{u}_t; \mathbf{s}_t; \mathbf{c}_{text} \oplus \mathbf{c}_{fuse}]$, where $\oplus$ denotes the addition operation. Based on it, we apply another MLP with softmax to produce a word distribution over token vocabulary V_{gen}:

$$P_{gen}(y_t) = \text{softmax}(\text{MLP}_{gen}(\mathbf{c}_t)). \tag{5.10}$$

To further allow the decoder to explicitly extract words from the source post, we apply the copy mechanism [127] by calculating a soft switch $\lambda_t \in [0, 1]$ with a sigmoid-activated MLP on $\mathbf{c}_t$. It indicates whether to generate the word from the vocabulary V_{gen} or copy it from the input sequence, where the extractive distribution is decided by the text attention weights $\alpha_{t,i}$ in Eq. (5.8).

Classification Output Aggregation. We further extend the copy mechanism to aggregate the classification's outputs to benefit keyphrase generation. First, we retrieve the top-K predictions from the classifier and convert each into the word sequence $\mathbf{w} = \langle w_1, ..., w_{l_w}\rangle$, where l_w is the sequence length of the combined

predictions. Then, we normalize their classification logits using softmax into a word-level distribution $\beta \in \mathbb{R}^{l_w}$, which represents the extractive probability from the classification output. Finally, we obtain the unified prediction via:

$$P_{unf}(y_t) = \lambda_t \cdot P_{gen}(y_t) \; + \; (1-\lambda_t) \cdot (a \cdot \sum_{i:x_i=y_t}^{l_x} \alpha_{t,i} + b \cdot \sum_{j:w_j=y_t}^{l_w} \beta_j), \tag{5.11}$$

where a, b $(a+b=1)$ are hyper-parameters to decide whether to copy from the input sequence or the classification outputs. To stabilize the aggregation of classification outputs, we warm up the classifier for several epochs first by setting a to 1 and b to 0 and then both to 0.5 for further training.

5.2.4 Joint Training Objective

We employ the standard negative log-likelihood loss and define the entire framework's training objective with the linear combination of the label classification loss and the token-level sequence generation loss for multitask learning:

$$\mathcal{L}(\theta) = -\sum_{n=1}^{N} [\underbrace{\log P_{cls}(\mathbf{y}^n)}_{\text{Classification}} + \gamma \cdot \sum_{t=1}^{l_y^n} \underbrace{\log P_{unf}(y_t^n)}_{\text{Unified}}], \tag{5.12}$$

where N is size of the training text-image pairs and γ is a hyper-parameter to balance the two losses (empirically set to 1) and θ denotes the trainable parameters shared for the whole framework. Intuitively, jointly training keyphrase classification would benefit the unified prediction by not only implicitly better parameter learning, but also explicitly providing more precise outputs to be copied by the aggregation module.

5.3 Experimental Setup

5.3.1 Data Collection

Since there are no publicly available datasets for multi-modal keyphrase annotation, we contribute a new dataset with social media posts from Twitter. Specifically, we employ the Twitter advanced search API[2] to query English tweets that contain both images and hashtags from January to June 2019. For keyphrases, we consider to use user-generated hashtags following common practice [176, 179].

Data Filtering. We clean the raw data in the following ways: (1) we only retain tweets with one color image in JPG form; (2) we remove tweets with less than 4 tokens or more than 5 hashtags to filter out noise data (e.g., #AI, #MachineLearning, #DeepLearning, #ML, #DL, #Tech, #ArtificialIntelligence); (3) rare hashtags (occurring less than 10 times) and their corresponding tweets are removed to alleviate sparsity issue; (4) we remove the duplicate tweets (e.g., retweets) and images and obtain 53,701 tweets with each containing a distinct tweet text-image pair.

Preprocessing. We employ an open-source Twitter preprocessing tool[3] [12] to tokenize the tweets, segment the hashtags, and apply common spelling corrections. To reduce the errors introduced by the automatic hashtag segmentation, we manually check them and construct a complete mapping list. Following Wang et al. [148], we retain tokens in hashtags (without

[2] https://twitter.com/search-advanced

[3] https://github.com/cbaziotis/ekphrasis

Split	#Post	Post Len	#KP /Post	|KP|	KP Len	% of occ. KP	Vocab
Train	42,959	27.26	1.33	4,261	1.85	37.14	48,019
Val	5,370	26.81	1.34	2,544	1.85	36.01	16,892
Test	5,372	27.05	1.32	2,534	1.86	37.45	17,021

Table 5.1: Data split statistics. KP: keyphrase; |KP|: the size of unique keyphrase; % of occ. KP: percentage of keyphrases occurring in the source post.

prefix) for those occurring in the middle of the posts due to their inseparable semantic roles. We further remove all the non-alphabetic tokens and replace links, mentions (@username), digits into special tokens as $\langle url \rangle$, $\langle mention \rangle$,and $\langle number \rangle$ respectively.

Finally, we obtain $53,701$ text-image tweets.. For training and evaluation, we randomly split the data into 80%, 10%, 10% corresponding to training, validation, and test set. The data split statistics of tweet texts are displayed in Table 5.1. We observe that only around 37% keyphrases appear in the source posts, making it difficult for extraction methods to perform well.

5.3.2 Dataset Analysis

Tweet Image Analysis. To further analyze the Twitter image characteristics, we sample 200 text-image tweets and analyze their distributions over varying types in Figure 5.4. We observe a rather diverse set of categories: cartoon/drawings (12%), posters (11%), sports-related images (11%), screenshots (6%), pure-text images (4%), and others (2%). We also notice that only around half of the images are natural photos (54%), which

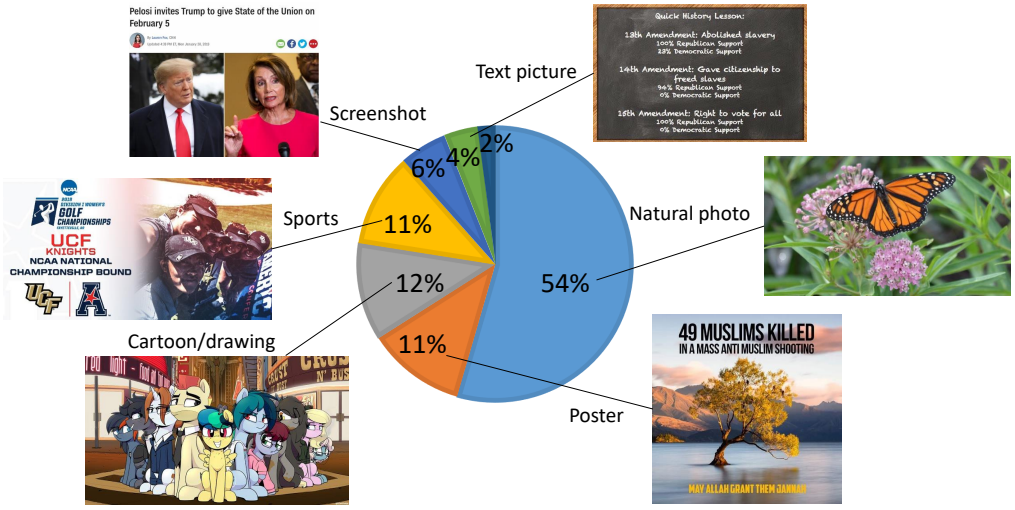


Figure 5.4: Image type distribution of 200 sampled text-image tweets in our collected dataset.

differs a lot from other image datasets such as MS-COCO. Moreover, we conduct a pilot study to categorize the text-image relations following Vempala et al.[142]. Some example tweets for four text-image relationships in our sampled set are shown in Figure 5.5. Post (a) represents text in the image and image adds to the semantics since it helps to infer that "good girl" refers to dogs, while in post (b), image represents but does not add to due to no additional information provided in the image. Post (c) does not represent text in the image but image adds to semantics as it reveals the connection between text with "Trump". As for post (d), image is just a comment for text and does not have a direct semantic association with text. We observe there are (1) 48%: image can represent text and add to more semantics of the tweet; (2) 25%: image can represent text but does not add to semantics; (3) 15%: image cannot represent text but add to semantics; (4) 12%: image cannot represent text and also does not add to semantics. Namely, 52% of them have either

Figure 5.5: Tweets of four different types of text-image relationship in our dataset. Post (a): text is represented and image adds to. Post (b): text is represented and image does not add to. Post (c): text is not represented and image adds to. Post (d): text is not represented and image does not add to.

texts or images useless to represent semantics. Such diverse categories of images and complex text-image relationship pose the challenge to attend essential information from noisy cross-media data, where our M^{3}H-Att and image wordings may help alleviate such issue.

Image Wording Analysis. Here we shed light on some interesting statistics on image wordings. We first visualize the word cloud of our image attributes in Figure 5.6. The top 5 attributes predicted from the images in our dataset are {*man, shirt, woman, sign, white*}, which shows that *most of the images on Twitter are about people.* The top 5 attributes predicted from the images in our dataset are {*man, shirt, woman, sign, white*}, which shows that *most of the images on Twitter are about people.* For OCR texts, we employ a widely used OCR engine Tesserocr[4] to extract optical characters. From all matching images, there are around 35% of them contain characters, significantly larger

[4]`https://pypi.org/project/tesserocr/`

Figure 5.6: Word cloud for the image attributes from our dataset, indicating most tweet images are about people.

than the corresponding number in COCO images (4%), indicating *social media users' preference to post images containing optical characters*. To mitigate the effects of OCR errors, we only consider tokens present in the vocabulary of tweet texts and find about 17% images left. We further analyze their statistics. Their median length is 16 tokens while 32% have words appearing in their corresponding keyphrases and 13% contain the entire keyphrases. This suggests the potential help from OCR texts in keyphrase prediction.

5.3.3 Comparisons

Evaluation Metrics. We mainly evaluate our model with popular information retrieval metrics macro-average F1@K, where K is 1 or 3 as there are 1.33 keyphrases on average per tweet (Table 5.1). To further measure the keyphrase orders (as we can generate a keyphrase ranking list with beam search), we employ mean average precision (MAP) for the top five predictions following Chen et al. [26]. The higher scores from all the metrics indicate better performance. For word matchings in evaluation,

we consider the results after processed with Porter Stemmer following Meng et al. [100].

Comparison Models. We first consider the upper-bound performance of extractive methods, denoted as EXT-ORACLE. Then, the following baselines are compared:

- **Image-only** models: we apply max/average pooling on the grid-level VGG features or object-level BUTD [5] and aggregate them for classification.
- **Text-only** models: we consider classification-based (CLS) or sequence generation-based (GEN) methods. For CLS models, we consider simple max/average pooling on the text features learned from Bi-GRU encoder and the Topic Memory Network (TMN) [174] (a SOTA short text classification model). For GEN models, we employ the seq2seq with attention [8], copy mechanism [127], and latent topics [148] (the SOTA topic-aware model for social media keyphrase generation).
- **Text-image** models: we consider the SOTA CLS model for multi-modal hashtag recommendation [175] using co-attention and its variant with image-attention [168], as well as Bilinear Attention Networks (BAN) [66] (a SOTA variant for Visual Question Answering [6]). For our models, we first adopt the basic variants with M^{3}H-Att separately applying to either CLS or GEN. Then we additionally combine image wordings and the joint training strategy (Eq. (5.12)). Our full model is obtained by further aggregating the CLS and GEN outputs (Eq. (5.11)).

5.3.4 Model Settings

We maintain a generation vocabulary V_{gen} of $45K$ tokens and the keyphrase classification vocabulary V_{cls} with $4,262$ labels. All the models are pretrained with 200-d Twitter GloVe embedding [114]. We employ two layers of Bi-GRU for the encoder and a single layer GRU for the decoder with hidden size set to 300. For visual signals, we extract either 49 grid-level VGG 512-d features or 36 object-level BUTD 2048-d features. We set up our models on the NVIDIA TITAN Xp GPU with 12G memory. In training, we set the loss coefficient $\gamma = 1$ and employ Adam optimizer [68] with a learning rate as 0.001. We decay it by 0.5 if validation loss does not drop and apply gradient clipping with the max gradient norm as 5. Early stop [21] is adopted via monitoring the change of validation loss. For the M^{3}H-Att, we employ 4 heads with 64-d subspace, where 4 layers are stacked for attention to text modality, and 1 layer for vision or attribute modality. For inference, we employ beam search with beam size set to 10 to generate a ranking list of keyphrases. For the baselines, we re-implement CLS-IMG-ATT and CLS-CO-ATT, and employ the released codes to produce results for CLS-TMN[5], GEN-TOPIC[6], and CLS-BAN[7].

5.4 Results and Analysis

5.4.1 Main Comparison Results

We first report the main comparison results in Table 5.2 and draw the following observations:

[5] https://github.com/zengjichuan/TMN
[6] https://github.com/yuewang-cuhk/TAKG
[7] https://github.com/jnhwkim/ban-vqa

	Models	F1@1	F1@3	MAP@5
	EXT-ORACLE	39.50	23.20	39.26
Image-only	CLS-VGG-MAX	14.20_{35}	12.20_{24}	17.68_{31}
	CLS-VGG-AVG	15.69_{21}	13.67_{06}	19.70_{20}
	CLS-BUTD-MAX	17.65_{32}	15.00_{21}	21.77_{29}
	CLS-BUTD-AVG	20.02_{27}	16.97_{06}	24.73_{11}
Text-only	CLS-AVG	35.96_{11}	27.59_{05}	41.84_{14}
	CLS-MAX	38.33_{47}	28.84_{09}	44.15_{34}
	CLS-TMN	40.33_{39}	30.07_{28}	46.28_{27}
	GEN-ATT	38.36_{28}	27.83_{15}	43.35_{20}
	GEN-COPY	42.10_{19}	29.91_{30}	46.94_{35}
	GEN-TOPIC	43.17_{24}	30.73_{13}	48.07_{23}
Text-Image	CLS-BAN	38.73_{18}	29.68_{23}	45.03_{15}
	CLS-IMG-ATT	41.48_{33}	31.22_{14}	47.93_{34}
	CLS-CO-ATT	42.12_{38}	31.55_{33}	48.39_{34}
	CLS-M^{3}H-ATT (ours)	44.11_{17}	31.47_{14}	49.45_{11}
	+ image wording	44.46_{12}	32.82_{24}	50.39_{15}
	+ joint-train	45.16_{09}	33.27_{10}	51.48_{11}
	GEN-M^{3}H-ATT (ours)	44.25_{05}	31.58_{13}	49.35_{10}
	+ image wording	44.56_{09}	31.77_{23}	49.95_{22}
	+ joint-train	45.69_{17}	32.78_{09}	51.37_{12}
	GEN-CLS-M^{3}H-ATT (ours)	$\mathbf{47.06}_{04}$	$\mathbf{33.11}_{01}$	$\mathbf{52.07}_{03}$

Table 5.2: Comparison results (in %) displayed with average scores from 5 random seeds. Our GEN-CLS-M^{3}H-ATT significantly outperforms all the comparison models (paired t-test $p < 0.05$). Subscripts denote the standard deviation (e.g., $47.06_{04} \Rightarrow 47.06 \pm 0.04$).

• *Textual features are more important than visual signals.* It is seen from the better performance of the text-only models compared with their counterparts relying solely on images. For image-only models, we find that object-level BUTD outperforms grid-level VGG, while for pooling methods, average pooling works better for visual signals while max pooling is more suitable for texts.[8]

• *Vision modality can provide complementary information to the text.* Most models considering cross-media signals perform better than text-only and image-only baselines. An exception is observed on CLS-CO-ATT, which indicates the limitation of traditional co-attention to well exploit multi-modality representations from social media.

• *Both M^{3}H-Att and image wordings are helpful to encode social media features.* We find that both M^{3}H-Att and image wordings contribute to the performance boost of keyphrase classification or generation or their joint training results, which showcase their ability to handle multi-modality data from social media. We will discuss more in 5.4.3.

• *Our output aggregation strategy is effective.* Seq2seq-based keyphrase generation models (especially armed with the copy mechanism to enable better extraction capability) perform better than most classification models and even upper bound results of extraction models. It is probably because of the high absent keyphrase rate and the large size of keyphrase tags (Table 5.1) exhibited in the noisy social media data. Nevertheless, GEN-CLS-M^{3}H-ATT, coupling advantages of classification and generation, obtains the best results (47.06 F1@1), drastically

[8]In experiments, we find that VGG works better than BUTD features for M^{3}H-Att in our variants. Below we show results with the better setting without otherwise specified.

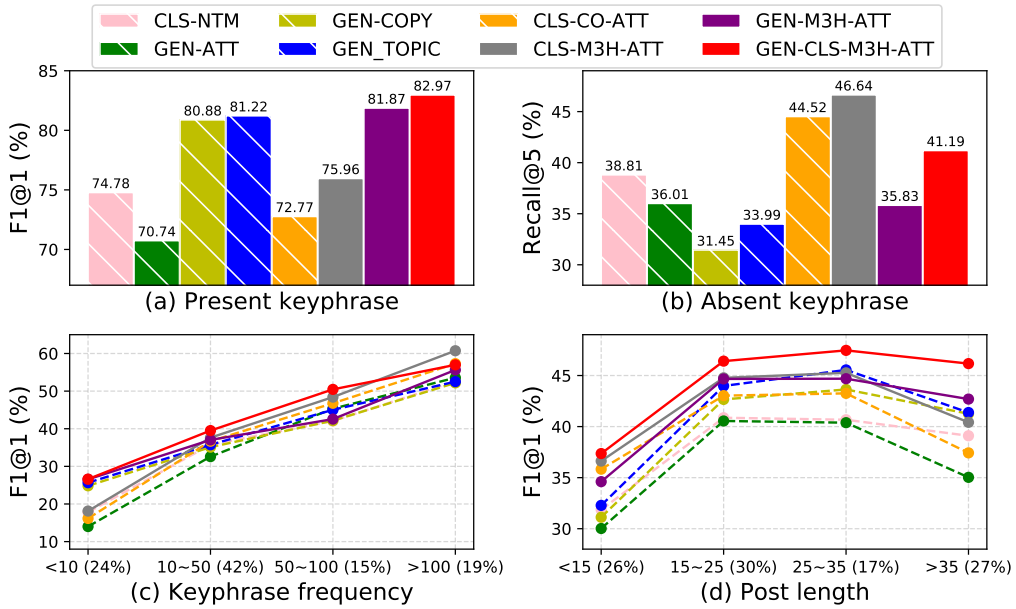


Figure 5.7: Model comparison over: (a) present keyphrases, (b) absent keyphrases, (c) varying keyphrase frequency, and (d) varying post length. Striped bars or dashed lines denote previous models while solid ones denote ours. In (a) and (b), x-axis: various models; y-axis: F1@1 for present and recall@5 for absent keyphrases. In (c) and (d), x-axis (%): data proportion; y-axis: F1@1. Best viewed in color.

outperforms the SOTA text-only model (43.17) and text-image one (42.12).

5.4.2 Quantitative Analysis

We examine how our model performs in diverse scenarios: present vs. absent keyphrases and varying keyphrase frequency and post length in Figure 5.7.

Present vs. Absent Keyphrases. We report the F1@1 for evaluating present keyphrases and recall@5 for absent keyphrases. As shown in Figure 5.7 (a-b), generation models with copy mechanism consistently outperform classification models for

present keyphrase, while the latter works better for absent keyphrases. Nonetheless, our output aggregation strategy is able to cover generation models' inferiority for absent keyphrases and exhibits better results from GEN-CLS-M^{3}H-ATT than GEN-M^{3}H-ATT (41.19 vs. 35.83 recall@5 score). Besides, visual signals are helpful to both generation and classification to yield either present or absent keyphrases, though larger boost is observed for the latter probably owing to the inadequate clues available from texts.

Keyphrase Frequency. From Figure 5.7 (c), we observe better F1@1 from all models to produce more frequent keyphrases, because common keyphrases allow better representation learning from more training instances. For extremely rare keyphrases (occur $<$ 10 times in training), generation models with copy mechanisms exhibit better capability to handle them than classification ones.

Post Length. From Figure 5.7 (d), we observe that longer post length does not guarantee better performance and the best results are obtained for posts with 15 $\sim$ 35 tokens. It might be attributed to the noisy nature of social media data — longer posts provide both richer contents and more noise. For the posts with $<$ 15 tokens, all multi-modal methods perform better than the text-only ones, as the image modality enriches the context for short texts.

5.4.3 Analysis of M^{3}H-Att and Image Wording

We proceed to quantify the effects of different settings in M^{3}H-Att and image wording.

# Layer	2 Head			4 Head			8 Head			12 Head		
	64-d	128-d	256-d	64-d	128-d	256-d	64-d	128-d	256-d	64-d	128-d	256-d
1	42.06	43.32	43.01	43.11	43.98	43.63	43.75	44.18	43.43	43.48	43.81	43.53
2	43.22	44.36	44.26	44.27	44.38	44.27	44.58	44.59	43.12	45.05	38.16	39.97
3	43.51	44.23	43.62	44.50	44.25	43.00	44.70	43.27	36.05	44.49	35.70	31.35
4	44.38	44.42	31.72	45.29	36.03	30.47	37.17	32.73	31.69	37.85	34.99	30.91

Table 5.3: Analysis of M^{3}H-Att with various stacked layer number, head number, and subspace dimension.

Models	No Image Wording			Add OCR				Add Attribute			
	Full	OCR	Attr	Full	Δ (%)	OCR	Δ (%)	Full	Δ (%)	Attr	Δ (%)
CLS-MAX	38.31	36.11	32.04	38.75	+1.1	40.67	+12.6	41.09	+7.3	37.87	+18.2
GEN-COPY	42.01	40.81	35.55	42.86	+2.0	43.58	+6.8	43.11	+2.6	38.10	+7.2
CLS-M^{3}H-ATT	44.19	42.93	36.93	44.27	+0.2	46.53	+8.4	44.38	+0.4	38.73	+4.9
GEN-M^{3}H-ATT	44.33	43.26	35.93	44.48	+0.3	46.31	+7.1	44.77	+1.0	39.90	+11.0

Table 5.4: F1@1 over three test sets with settings: no image wording, adding either OCR or attribute. Δ: the relative improvements over no image wording.

M^{3}H-Att Analysis. We investigate how various configurations ($L_{vis} \in \{1, 2, 3, 4\}$, $H \in \{2, 4, 8, 12\}$, $d_H \in \{64, 128, 256\}$) of our M^{3}H-Att affect the prediction results in Table 5.3. Here we only show the classification results (and similar trends are observed from generation). We notice that more complex models do not always present better results and even render performance deteriorate in some cases due to the overfitting issue. The best performance is attained by 4 stacked layers of 4 heads with a 64-d subspace.

Image Wording Analysis. To examine image wording effects, we compare four models in three settings: no image wording, OCR (only), and image attributes (only) in Table 5.4. The results are shown in three test sets: the entire test set (Full), the 889

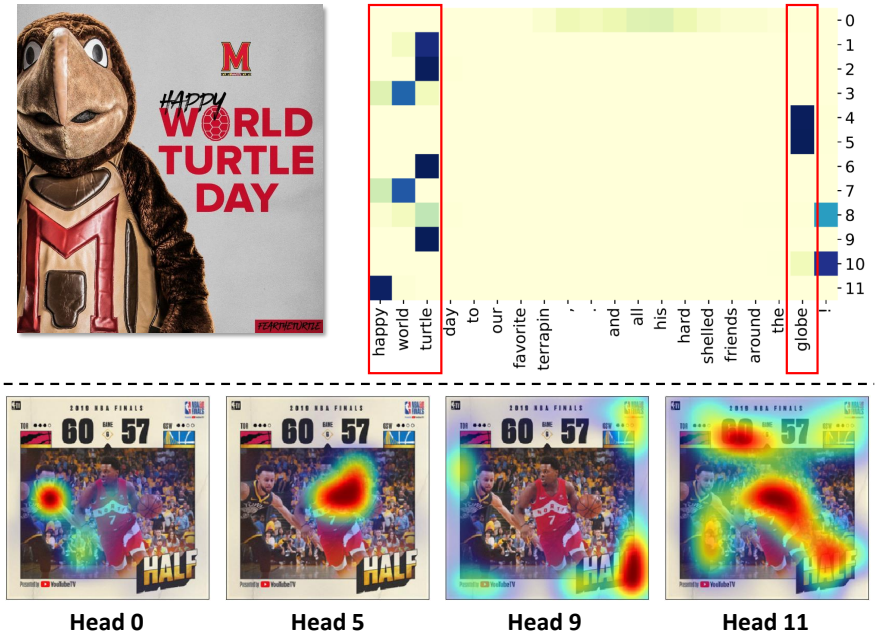


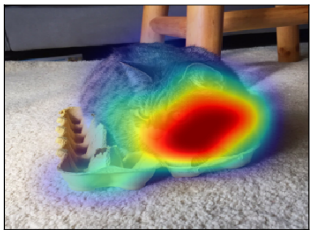
Figure 5.8: Attention weight visualization of M^{3}H-Att for two example posts with image-to-text (top) and text-to-image attention (bottom). Best viewed in color.

subset instances with OCR tokens (OCR), and the 266 ones containing keyphrases from ImageNet labels[9] (Attr) [123]. For the CLS-MAX and GEN-COPY, we add attributes by using its max-pooled features to attend the text memory, which is later used for prediction.

We observe that either OCR texts or image attributes contribute to better F1@1 on the entire test set for all chosen models, while much more performance gain can be observed on their subsets with OCR texts or ImageNet keyphrases, indicating that images with optical characters and natural styles can benefit more from image wordings.

[9]Here we assume that posts with ImageNet keyphrases have a higher probability to contain natural photos drawn from our observations.

Post (a): Contemplating the mysteries of life from inside my egg carton...☺ *#CatsOfTwitter*



(cat yellow grey bananas)
GEN-COPY: star wars
CLS-CO-ATT: **cats of twitter**
Our: **cats of twitter**

Post (b): Epic Texas *#sunset* from NNE Bastrop County TX. @TxStormChasers

(sky sun sunset field)
GEN-COPY: storm hour
CLS-CO-ATT: storm hour
Our: **sunset**

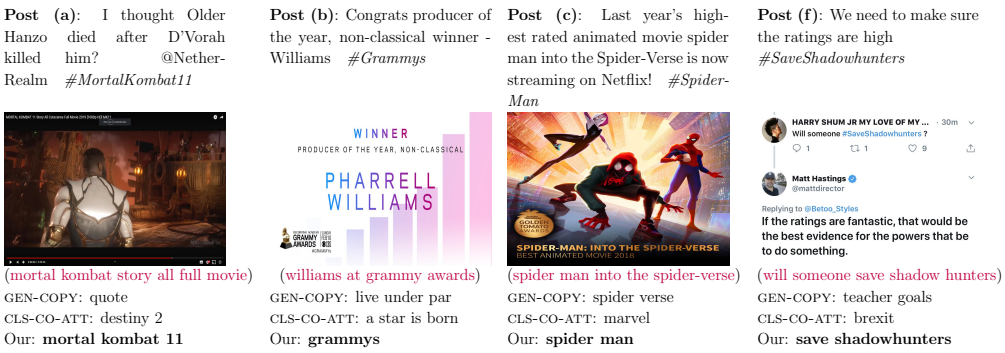
Post (c): Your plastic bag ends up somewhere, and sometimes, it goes to the ocean. *#WorldOceansDay*

(world oceans day June 8)
GEN-COPY: plastic fandom
CLS-CO-ATT: plastic
Our: **world oceans day**

Figure 5.9: Tweet image's effects for keyphrase prediction. Blue tokens are the top four attributes and purple ones are OCR tokens. Correct predictions are in **bold**.

5.4.4 Qualitative Analysis

To explore whether M^{3}H-Att is able to attend different aspects from the image, we probe into its attention weights via heatmap visualization in Figure 5.8. Here CLS-M^{3}H-ATT is employed with a single layer of 12 heads, whose image-to-text and text-to-image attention are examined. The top figure shows that all its heads attend to the text based on the visual cues, where some attend to "turtle" while others attend to "world" and "globe" with various emphasis. Interestingly, Head 11 highlights the "happy" token, which also appears in the image. For the text-to-image attentions (bottom), we find some heads tend to highlight the specific local objects, such as the two players by Head 0 and 5 and the textual regions by Head 9, while some capture a more global view of the image like Head 11. We provide more attention visualizations in Figure 5.10, where our M^{3}H-Att is able to attend various aspects from both image-to-text or text-



(mortal kombat story all full movie)
GEN-COPY: quote
CLS-CO-ATT: destiny 2
Our: **mortal kombat 11**

(williams at grammy awards)
GEN-COPY: live under par
CLS-CO-ATT: a star is born
Our: **grammys**

(spider man into the spider-verse)
GEN-COPY: spider verse
CLS-CO-ATT: marvel
Our: **spider man**

(will someone save shadow hunters)
GEN-COPY: teacher goals
CLS-CO-ATT: brexit
Our: **save shadowhunters**

Figure 5.11: More qualitative examples showing the effectiveness of encoding OCR texts. Among various models, only our model that considers OCR tokens correctly predicts the keyphrases (in bold). Purple tokens are some of OCR tokens detected by an off-the-shelf OCR engine. We observe that keyphrases directly appear in these images.

to-image directions with different heads.

We further illustrate how images (visual signals, image attributes, and OCR tokens) help cross-media keyphrase prediction by analyzing their predictions in Figure 5.9. In post (a), visual features help both CLS-CO-ATT and our model correctly predict its keyphrase, where our model precisely attends the cat's face (key region reflecting the image's semantics) . Without such context, GEN-COPY wrongly predicts *"star wars"*, which might be caused by the misleading token *"mysterious"* in the texts. Besides, the keyphrase is also revealed in the top predicted attribute. In post (b-c), only our model with image wordings makes correct predictions, where we observe that the ground-truth keyphrases directly appear in the attributes or OCR texts. More outputs from different models are provided for demonstrating the effectiveness of OCR texts (Figure 5.11) and image attributes (Figure 5.12). Among most of these cases, image wordings help our model to correctly predict keyphrases

Post (a): Good night, everyone. I hope that you have had a delightful day and a restful weekend. *#hoorayfordogs*

(dog white yellow brown plate)
GEN-COPY: friday feeling
CLS-CO-ATT: **hooray for dogs**
Our: **hooray for dogs**

Post (b): Head up, chest out! A handsome purple finch poses for a shot. *#birds #wildlife #photography*

(branch bird red top small)
GEN-COPY: gap ol
CLS-CO-ATT: **birding**
Our: **birds**; **wildlife**

Post (c): I was watching all the bees Honeybee collecting pollen on the flowers Bouquet *#CatsOfTwitter*

(cat white pink grey flowers)
GEN-COPY: photography
CLS-CO-ATT: springwatch
Our: **cats of twitter**

Post (d): For 1970, Plymouth intended to make its GTX model a street powerhouse. *#MuscleCar #ClassicCar*

(car roof park old meter)
GEN-COPY: plymouth
CLS-CO-ATT: mopar
Our: **classic car**

Figure 5.12: More qualitative examples showing the effectiveness of encoding image attributes. Our model that considers image attributes correctly predicts the keyphrases for all these cases (in bold). Blue tokens are the top five predicted attributes.

while GEN-COPY considering only texts and CLS-CO-ATT relying on both texts and images fail to so.

5.5 Summary

In this chapter, we extensively study cross-media keyphrase prediction on social media and present a unified framework to couple the advantages of generation and classification models for this task. Moreover, we propose a novel *Multi-Modality Multi-Head Attention* to capture the dense interactions between texts and images, where image wordings explicit in optical characters and implicit in image attributes are further exploited to bridge their semantic gap. Experimental results on a large-scale newly-collected Twitter corpus show that our model significantly outperforms SOTA either generation or classification models with traditional attentions. Further discussions show our ability to attend useful multi-modal features to indicate keyphrases.

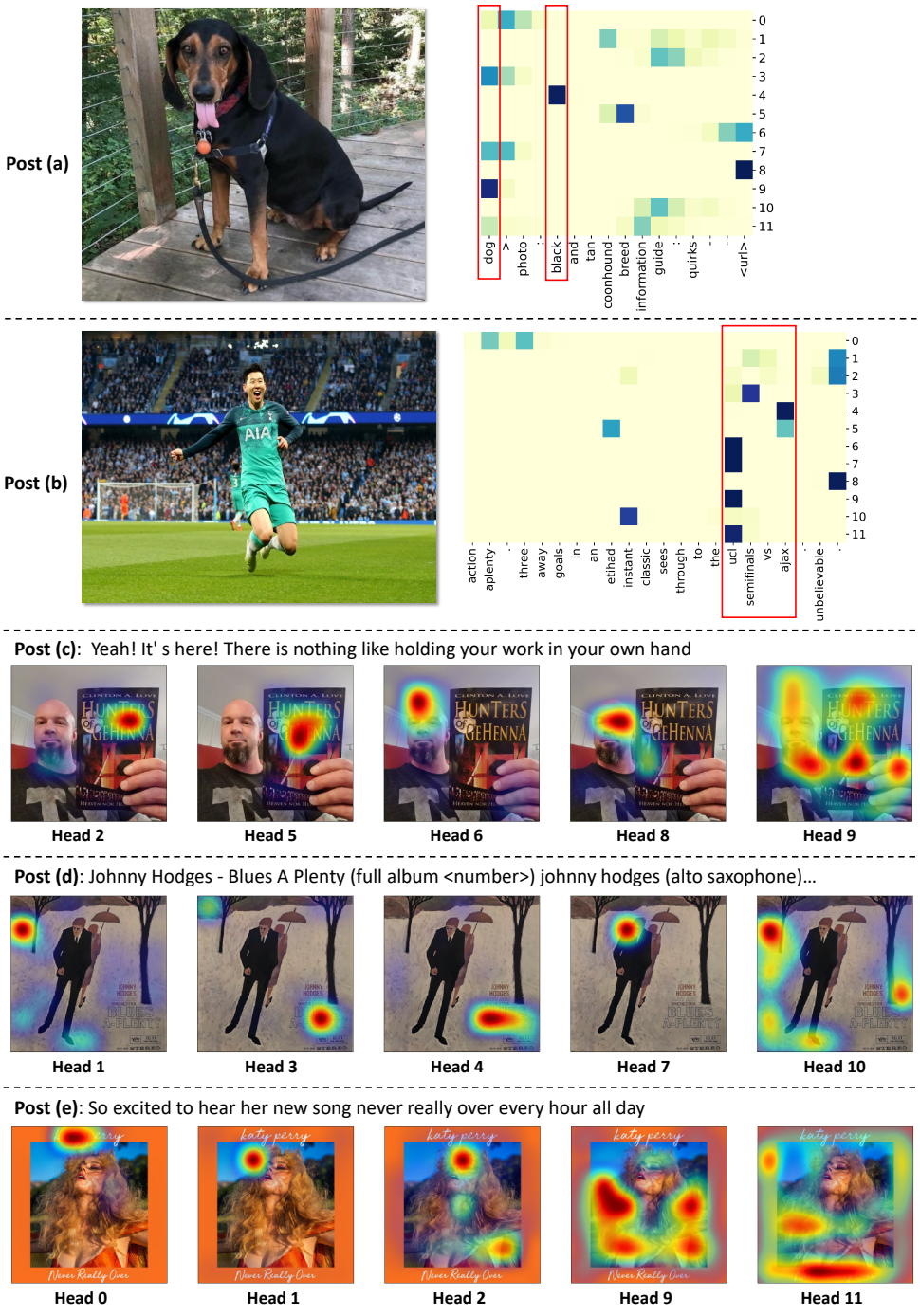


Figure 5.10: More attention weight visualization for both image-to-text attention and text-to-image attention.

Chapter 6

Vision-Language Pretraining for Visual Dialog

In cross-modality learning, the core step is to fuse features from distinct modalities and derive a joint generic representation for various downstream applications. In this chapter, we take a further step to study how to effectively learn visual and linguistic representations in a more general task: visual dialog. It is one of the most challenging tasks where an agent is required to answer a series of questions grounded on an image. We explore the use of Vision and Language pretraining with Transformers for this task. The main points of this chapter are as follows. (1) We propose a unified vision-dialog Transformer with BERT (VD-BERT) for visual dialog tasks, which captures the intricate interactions between image and dialog using Transformer and achieves their effective fusion from the two modalities via simple visually grounded training. (2) Our VD-BERT supports both answer ranking and answer generation seamlessly through the same architecture. (3) Our model achieves effective vision and language fusion within a unified Transformer encoder and yields a new state of the art for visual dialog tasks.

6.1 Introduction

Visual Dialog (or VisDial) aims to build an AI agent that can answer a human's questions about visual content in a natural conversational setting [33]. Unlike the traditional single-turn Visual Question Answering (VQA) [6], the agent in VisDial requires to answer questions through multiple rounds of interactions together with visual content understanding.

The primary research direction in VisDial has been mostly focusing on developing various attention mechanisms [8] for a better fusion of vision and dialog contents. Compared to VQA that predicts an answer based only on the question about the image (Figure 6.1 (a)), VisDial needs to additionally consider the dialog history. Typically, most of previous work [110, 40, 60] uses the question as a query to attend to relevant image regions and dialog history, where their interactions are usually further exploited to obtain better visual-historical cues for predicting the answer. In other words, the attention flow in these methods is *unidirectional* – from question to the other components (Figure 6.1 (b)).

By contrast, in this work, we allow for *bidirectional* attention flow between all the entities using a unified Transformer [141] encoder, as shown in Figure 6.1 (c). In this way, all the entities simultaneously play the role of an "information seeker" (query) and an "information provider' (key-value), thereby fully unleashing the potential of attention similar to [125]. We employ the Transformer as the encoding backbone due to its powerful representation learning capability exhibited in pretrained language models like BERT [35]. Inspired by its recent success in vision-language pretraining, we further extend

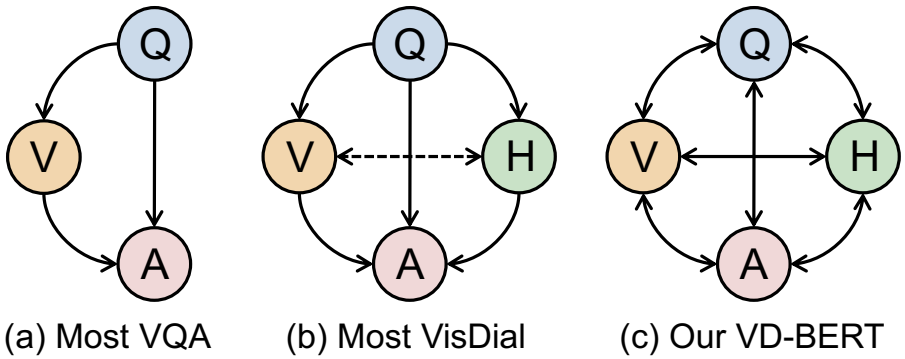


Figure 6.1: Attention flow direction illustration. V: vision, H: dialog history, Q: question, A: answer. The arrow denotes the attention flow direction and the dashed line represents an optional connection.

BERT to achieve simple yet effective fusion of vision and dialog contents in VisDial tasks.

Recently several emerging works have attempted to adapt BERT for multimodal tasks [138, 92, 140, 181]. They often use self-supervised objectives to pretrain BERT-like models on large-scale external vision-language data and then fine-tune on downstream tasks. This has led to compelling results in tasks such as VQA, image captioning, image retrieval [170], and visual reasoning [136]. However, it is still unclear how visual dialog may benefit from such vision-language pretraining due to its unique multi-turn conversational structure. Specifically, each image in the VisDial dataset is associated with up to 10 dialog turns, which contain much longer contexts than either VQA or image captioning.

In this work, we present VD-BERT, a novel unified vision-dialog Transformer framework for VisDial tasks. Specifically, we first encode the image into a series of detected objects and feed them into a Transformer encoder together with the image caption and multi-turn dialog. We initialize the encoder with BERT for

better leveraging the pretrained language representations. To effectively fuse features from the two modalities, we make use of two *visually grounded* training objectives – Masked Language Modeling (MLM) and Next Sentence Prediction (NSP). Different from the original MLM and NSP in BERT, we additionally take the visual information into account when predicting the masked tokens or the next answer.

VisDial models have been trained in one of two settings: discriminative or generative. In the discriminative setting, the model ranks a pool of answer candidates, whereas the generative setting additionally allows the model to generate the answers. Instead of employing two types of decoders like prior work, we rely on a unified Transformer architecture with two different self-attention masks [36] to seamlessly support both settings. During inference, our VD-BERT either ranks the answer candidates according to their NSP scores or generates the answer sequence by recursively applying the MLM operations. We further fine-tune our model on dense annotations that specify the relevance score for each answer candidate with a ranking optimization module.

In summary, we make the following contributions:

- To the best of our knowledge, our work serves as one of the first attempts to explore pretrained language models for visual dialog. We showcase that BERT can be effectively adapted to this task with simple visually grounded training for capturing the intricate vision-dialog interactions. Besides, our VD-BERT is the first unified model that supports both discriminative and generative training settings without explicit decoders.

- We conduct extensive experiments not only to analyze how our model performs with various training aspects (Section 6.5.2) and fine-tuning on dense annotations (Section 6.5.4), but also to interpret it via attention visualization (Section 6.5.3), shedding light on future transfer learning research for VisDial tasks.
- Without the need to pretrain on external vision-language data, our model yields new state-of-the-art results in the discriminative setting and promising results in the generative setting on the visual dialog benchmarks (Section 6.5.1).

6.2 Related Work

Visual Dialog. The Visual Dialog task has been recently proposed by Das et al. [33], where a dialog agent needs to answer a series of questions grounded by an image. It is one of the most challenging vision-language tasks that require not only to understand the image content according to texts, but also to reason through the dialog history. Previous work [93, 130, 159, 71, 59, 167, 51, 110] focuses on developing a variety of attention mechanisms to model the interactions among entities including image, question, and dialog history. For example, Kang et al. [60] proposed DAN, a dual attention module to first refer to relevant contexts in the dialog history, and then find indicative image regions. ReDAN, proposed by Gan et al. [40], further explores the interactions between image and dialog history via multi-step reasoning.

Different from them, we rely on the self-attention mechanism of the Transformer model to capture such interactions in a unified manner and derive a "holistic" contextualized representation

for all the entities. Similar to this, Schwartz et al. [125] proposed FGA, a general factor graph attention that can model interactions between any two entities but in a pairwise manner. There is recent work [109, 3] also applying the Transformer to model the interactions among many entities. However, their model neglects the important early interaction of the answer entity and cannot naturally leverage the pretrained language representations from BERT like ours.

Regarding the architecture, our model mainly differs from previous work in two facets: first, unlike most prior work that considers answer candidates only at the final similarity computation layer, our VD-BERT integrates each answer candidate at the input layer to enable its early and deep fusion with other entities, similar to [125]; second, existing models adopt an encoder-decoder framework [139] with two types of decoder for the discriminative and generative settings separately, where we instead adopt a unified Transformer encoder with two different self-attention masks [36] to seamlessly support both settings without extra decoders.

Pretraining in Vision and Language. Pretrained language models like ELMo [115], GPT [119], and BERT [35] have boosted performance greatly in a broad set of NLP tasks. In order to benefit from the pretraining, there are many recent work on extending BERT for vision and language pretraining. They typically employ the Transformer encoder as the backbone with either a two-stream architecture to encode text and image independently such as ViLBERT [92] and LXMERT [140], or a single-stream architecture to encode both text and image together, such as B2T2 [4], Unicoder-VL [79], VisualBERT [86],

VL-BERT [135], and UNITER [28]. Our VD-BERT belongs to the second group. These models yield prominent improvements in a wide spectrum of understanding-based vision-language tasks including VQA, text-image retrieval [170, 62], visual entailment [162], referring expression [63], visual reasoning [136], and commonsense reasoning [172].

More recently, Zhou et al. [181] proposed VLP which also allows generation using a unified Transformer with various self-attention masks [36]. Their model was proposed for VQA and image captioning. Our model is inspired by VLP and specifically tailored for the visual dialog task. Most closely related to this work is the concurrent work VisDial-BERT by [105], who also employ vision-language pretrained models (i.e., ViLBERT) for visual dialog. Our work has two major advantages over VisDial-BERT: first, VD-BERT supports both discriminative and generative settings while theirs is restricted to only the discriminative setting; second, we do not require to pretrain on large-scale external vision-language datasets like theirs and still yield better performance (Section 6.5.1).

6.3 The VD-BERT Model

We first formally describe the visual dialog task. Given a question Q_t grounded on an image I at t-th turn, as well as its dialog history formulated as $H_t = \{C, (Q_1, A_1), ..., (Q_{t-1}, A_{t-1})\}$ (where C denotes the image caption), the agent is asked to predict its answer A_t by ranking a list of 100 answer candidates $\{\hat{A}_t^1, \hat{A}_t^2, ..., \hat{A}_t^{100}\}$. In general, there are two types of decoder to predict the answer: a *discriminative* decoder that *ranks* the answer candidates and is trained with a cross entropy loss, or

a *generative* decoder that *synthesizes* an answer and is trained with a maximum log-likelihood loss.

Figure 6.2 shows the overview of our approach. First, we employ a unified vision-dialog Transformer to encode both the image and dialog history, where we append an answer candidate $\hat{A}_t$ in the input to model their interactions in an early fusion manner. Next, we adopt visually grounded MLM and NSP objectives to train the model for effective vision and dialog fusion using two types of self-attention masks – bidirectional and seq2seq. This allows our unified model to work in both discriminative and generative settings. Lastly, we devise a ranking optimization module to further fine-tune on the dense annotations.

6.3.1 Vision-Dialog Transformer Encoder

Vision Features. Following previous work, we employ Faster R-CNN [120] pretrained on Visual Genome [72] to extract the object-level vision features. Let $O_I = \{o_1, ..., o_k\}$ denote the vision features for an image I, where each object feature o_i is a 2048-d Region-of-Interest (RoI) feature and k is the number of the detected objects (fixed to 36 in our setting). As there is no natural orders among these objects, we adopt normalized bounding box coordinates as the spatial location. Specifically, let (x_1, y_1) and (x_2, y_2) denote the coordinates of the bottom-left and top-right corner of the object o_i, its location is encoded into a 5-d vector: $p_i = (\frac{x_1}{W}, \frac{y_1}{H}, \frac{x_2}{W}, \frac{y_2}{H}, \frac{(x_2-x_1)(y_2-y_1)}{WH})$, where W and H respectively denote the width and height of the input image, and the last element is the relative area of the object. We further extend p_i with its class id and confidence score for a richer representation.

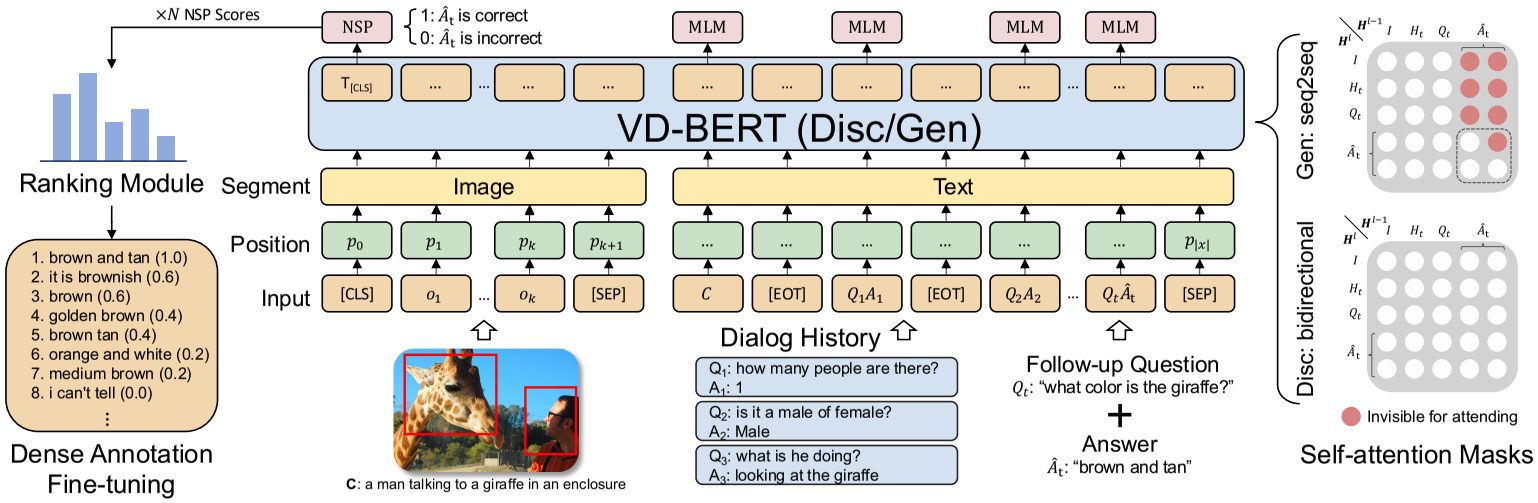


Figure 6.2: The model architecture of our unified VD-BERT. It first encodes the input image I, multi-turn dialog history H_t (including the caption C), follow-up question Q_t, and the appended answer candidate $\hat{A}_t$ into a single-stream Transformer encoder, and then train it with two *visually grounded* learning objectives: masked language modeling (MLM) and next sentence prediction (NSP). The NSP is trained to distinguish whether $\hat{A}_t$ is the correct answer or not. The unified VD-BERT supports both *discriminative* (Disc) and *generative* (Gen) settings by adopting bidirectional and sequence-to-sequence (seq2seq) self-attention masks, respectively. The NSP scores of N answer candidates are further optimized using a ranking module based on the provided dense annotations.

Language Features. We pack all the textual elements (caption and multi-turn dialog) into a long sequence. We employ WordPiece tokenizer [160] to split it into a word sequence $\mathbf{w}$, where each word is embedded with an absolute positional code following Devlin et al. [35].

Cross-Modality Encoding. To feed both image and text into the Transformer encoder, we integrate the image objects with language elements into a whole input sequence. Similar to BERT, we use special tokens like `[CLS]` to denote the beginning of the sequence, and `[SEP]` to separate the two modalities. Moreover, to inject the multi-turn dialog structure into the model, we utilize a special token `[EOT]` to denote *end of turn* [155], which informs the model when the dialog turn ends. As such, we prepare the input sequence into the format as $\mathbf{x}$ = (`[CLS]`, $o_1, ..., o_k$, `[SEP]`, C, `[EOT]`, Q_1A_1, `[EOT]`, ..., $Q_t\hat{A}_t$, `[SEP]`). To notify the model for the answer prediction, we further insert a `[PRED]` token between the $Q_t\hat{A}_t$ pair. Finally, each input token embedding is combined with its position embedding and segment embedding (0 or 1, indicating whether it is image or text) with layer normalization [7].

Transformer Backbone. We denote the embedded vision-language inputs as $\mathbf{H}^0 = [\mathbf{e}_1, ..., \mathbf{e}_{|\mathbf{x}|}]$ and then encode them into multiple levels of contextual representations $\mathbf{H}^l = [\mathbf{h}_1^l, ..., \mathbf{h}_{|\mathbf{x}|}^l]$ using L-stacked Transformer blocks, where the l-th Transformer block is denoted as $\mathbf{H}^l = \text{Transformer}(\mathbf{H}^{l-1}), l \in [1, L]$. Inside each Transformer block, the previous layer's output $\mathbf{H}^{l-1} \in \mathbb{R}^{|\mathbf{x}| \times d_h}$ is aggregated using the multi-head self-attention [141]:

$$\mathbf{Q} = \mathbf{H}^{l-1}\mathbf{W}_l^Q, \mathbf{K} = \mathbf{H}^{l-1}\mathbf{W}_l^K, \mathbf{V} = \mathbf{H}^{l-1}\mathbf{W}_l^V, \tag{6.1}$$

$$\mathbf{M}_{ij} = \begin{cases} 0, & \text{allow to attend} \\ -\infty, & \text{prevent from attending} \end{cases} \tag{6.2}$$

$$\mathbf{A}_l = \text{softmax}(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{d_k}} + \mathbf{M})\mathbf{V}, \tag{6.3}$$

where $\mathbf{W}_l^Q, \mathbf{W}_l^K, \mathbf{W}_l^V \in \mathbb{R}^{d_h \times d_k}$ are learnable weights for computing the queries, keys, and values respectively, and $\mathbf{M} \in \mathbb{R}^{|\mathbf{x}| \times |\mathbf{x}|}$ is the self-attention mask that determines whether tokens from two layers can attend each other. Then $\mathbf{A}_l$ is passed into a feedforward layer with a residual connection [52] to compute $\mathbf{H}^l$ for next layer. In the following, the self-attention mask $\mathbf{M}$ will be adjusted accordingly to support different training settings.

6.3.2 Visually Grounded Training Objectives

We use two *visually grounded* training objectives—masked language modeling (MLM) and next sentence prediction (NSP) to train our VD-BERT. Particularly, we aim to capture dense interactions among both inter-modality (i.e., image-dialog) and intra-modality (i.e., image-image, dialog-dialog).

Similar to MLM in BERT, 15% tokens in the text segment (including special tokens like `[EOT]` and `[SEP]`) are randomly masked out and replaced with a special token `[MASK]`. The model is then required to recover them based not only on the surrounding tokens $\mathbf{w}_{\setminus m}$ but also on the image I:

$$\mathcal{L}_{MLM} = -E_{(I,\mathbf{w})\sim D} \log P(w_m | \mathbf{w}_{\setminus m}, I), \tag{6.4}$$

where w_m refers to the masked token and D denotes the training set. Following Zhou et al. [181], we do not conduct masked object/region modeling in the image segment.

As for NSP, instead of modeling the relationship between two sentences (as in BERT) or the matching of an image-text pair (as in other vision-language pretraining models like ViLBERT), VD-BERT aims to predict whether the appended answer candidate $\hat{A}_t$ is correct or not based on the joint understanding of the image and dialog history:

$$\mathcal{L}_{NSP} = -E_{(I,\mathbf{w})\sim D} \log P(y|S(I,\mathbf{w})), \tag{6.5}$$

where $y \in \{0,1\}$ indicates whether $\hat{A}_t$ is correct, and $S(\cdot)$ is a binary classifier to predict the probability based on the `[CLS]` representation $\mathrm{T}_{\texttt{[CLS]}}$ at the final layer. Below we introduce the discriminative and generative settings of VD-BERT.

Discriminative Setting. For training in the discriminative setting, we transform the task of selecting an answer into a point-wise binary classification problem. Concretely, we sample an answer $\hat{A}_t$ from the candidate pool and append it to the input sequence, and ask the NSP head to distinguish whether the sampled answer is correct or not. We employ the *bidirectional* self-attention mask to allow all the tokens to attend to each other by setting the mask matrix $\mathbf{M}$ in Eq. (6.2) to all 0s. To avoid imbalanced class distribution, we keep the ratio of positive and negative instances to 1:1 in each epoch. To encourage the model to penalize more on negative instances, we randomly resample a negative example from the pool of 99 negatives w.r.t. every positive one at different epochs. During inference, we rank the answer candidates according to the positive class score of their NSP heads.

Generative Setting. In order to autoregressively generate an answer, we also train VD-BERT with the *sequence-to-sequence*

(seq2seq) self-attention mask [36]. For this, we divide the input sequence to each Transformer block into two subsequences, *context* and *answer*:

$$\mathbf{x} \triangleq (I, \mathbf{w}) = (\underbrace{I, H_t, Q_t,}_{\text{context}} \hat{A}_t). \tag{6.6}$$

We allow tokens in the context to be fully visible for attending by setting the left part of $\mathbf{M}$ to all 0s. For the answer sequence, we mask out (by setting $-\infty$ in $\mathbf{M}$) the "future" tokens to get autoregressive attentions (see the red dots in Figure 6.2).
During inference, we rely on the same unified Transformer encoder with sequential MLM operations without an explicit decoder. Specifically, we recursively append a `[MASK]` token to the end of the sequence to trigger a one-step prediction and then replace it with the predicted token for the next token prediction. The decoding process is based on greedy sampling and terminated when a `[SEP]` is emitted, and the resulting log-likelihood scores will be used for ranking the answer candidates.

6.3.3 Fine-tuning with Rank Optimization

As some answer candidates may be semantically similar (e.g. "brown and tan" vs "brown" in Figure 6.2), VisDial v1.0 additionally provides dense annotations that specify real-valued relevance scores for the 100 answer candidates, $[s_1, ..., s_{100}]$ with $s_i \in [0, 1]$. To fine-tune on this, we combine the NSP scores from the model for all answer candidates together into a vector $[p_1, ..., p_{100}]$.
As dense annotation fine-tuning is typically a Learning to Rank (LTR) problem, we can make use of some ranking optimization methods. After comparing various methods in Table 6.3c, we

adopt ListNet [20] with the top-1 approximation as the ranking module for VD-BERT:

$$\mathcal{L}_{ListNet} = -\sum_{i=1}^{N} f(s_i) \log(f(p_i)), \tag{6.7}$$

$$f(x_i) = \frac{\exp(x_i)}{\sum_{j=1}^{N} \exp(x_j)}, \quad i = 1, ..., N. \tag{6.8}$$

For training efficiency, we sub-sample the candidate list and use only $N = 30$ answers (out of 100) for each instance. To better leverage the contrastive signals from the dense annotations, the sub-sampling method first picks randomly the candidates with non-zero relevance scores, and then it picks the ones from zero scores (about 12% of candidates are non-zero on average).

6.4 Experimental Setup

6.4.1 Datasets

We evaluate our model on the VisDial v0.9 and v1.0 datasets[1] [33]. Specifically, v0.9 contains a training set of 82,783 images and a validation set of 40,504 images. The v1.0 dataset combines the training and validation sets of v0.9 into one training set and adds another 2,064 images for validation and 8,000 images for testing (hosted blindly in the task organizers' server). Each image is associated with one caption and 10 question-answer pairs. For each question, it is paired with a list of 100 answer candidates, one of which is regarded as the correct answer.

Apart from these sparse annotations, extra dense annotations for the answer candidates are provided for the v1.0 validation

[1] Available at `https://visualdialog.org/data`

split and a part of v1.0 train split (2,000 images) to make the evaluation more reasonable. The dense annotation specifies a relevance score for each answer candidate based on the fact that some candidates with similar semantics to the ground truth answer can also be considered as correct or partially correct, e.g., "brown and tan" and "brown" in Figure 6.2.

6.4.2 Evaluation Metric

Following Das et al. [33], we evaluate our model using the ranking metrics like Recall@K (K $\in \{1, 5, 10\}$), Mean Reciprocal Rank (MRR), and Mean Rank, where only one answer is considered as correct. Since the 2018 VisDial challenge (after the acquisition of dense annotations), NDCG metric that considers the relevance degree of each answer candidate, has been adopted as the main metric; the winner of the challenge is picked based solely on this metric.

6.4.3 Model Settings

We use $\text{BERT}_{\text{BASE}}$ as the backbone, which consists of 12 Transformer blocks, each with 12 attention heads and a hidden state dimensions of 768. We keep the max input sequence length (including 36 visual objects) to 250. We use Adam [68] with an initial learning rate of $3e-5$ and a batch size of 32 to train our model. A linear learning rate decay schedule with a warmup of 0.1 is employed. We first train VD-BERT for 30 epochs on a cluster of 4 V100 GPUs with 16G memory using MLM and NSP losses (with equal coefficients). Here we only utilize one previous dialog turn for training efficiency. For instances where the appended answer candidate is incorrect, we do not conduct

MLM on the answer sequence to reduce the noise introduced by the negative samples. After that, we train for another 10 epochs with full dialog history using either NSP in the discriminative setting or MLM on the answer sequence in the generative setting. For dense annotation fine-tuning in the discriminative setting, we train with the ListNet loss for 5 epochs.

6.5 Results and Analysis

In this section, we first compare our VD-BERT with state-of-the-art baselines on VisDial datasets. Then we conduct extensive ablation studies to examine various aspects of our model. Further, we interpret how VD-BERT attains the effective fusion of vision and dialog via visualizing attention weights, followed by an in-depth analysis of fine-tuning on dense annotations.

6.5.1 Main Results

We report main quantitative comparison results on both VisDial v1.0 and v0.9 datasets below.

Comparison. We consider state-of-the-art published baselines, including NMN [56], CorefNMN [71], GNN [180], FGA [125], DVAN [49], RvA [110], DualVD [59], HACAN [167], Synergistic [51], DAN [60], ReDAN [40], CAG [50], Square [65], MCA [3], MReal-BDAI and P1_P2 [117]. We further report results from the leaderboard[2] for a more up-to-date comparison, where some can be found in the arXiv, such as MVAN [112], SGLNs [61], VisDial-BERT [105], and Tohoku-CV [109].

[2] https://evalai.cloudcv.org/web/challenges/challenge-page/161/leaderboard/483\#leaderboardrank-1

	Model	NDCG↑	MRR↑	R@1↑	R@5↑	R@10↑	Mean ↓
Published Results	NMN	58.10	58.80	44.15	76.88	86.88	4.81
	CorefNMN	54.70	61.50	47.55	78.10	88.80	4.40
	GNN	52.82	61.37	47.33	77.98	87.83	4.57
	FGA	52.10	63.70	49.58	80.97	88.55	4.51
	DVAN	54.70	62.58	48.90	79.35	89.03	4.36
	RvA	55.59	63.03	49.03	80.40	89.83	4.18
	DualVD	56.32	63.23	49.25	80.23	89.70	4.11
	HACAN	57.17	64.22	50.88	80.63	89.45	4.20
	Synergistic	57.32	62.20	47.90	80.43	89.95	4.17
	Synergistic†	57.88	63.42	49.30	80.77	<u>90.68</u>	3.97
	DAN	57.59	63.20	49.63	79.75	89.35	4.30
	DAN†	59.36	<u>64.92</u>	51.28	<u>81.60</u>	**90.88**	<u>3.92</u>
	ReDAN†	64.47	53.73	42.45	64.68	75.68	6.64
	CAG	56.64	63.49	49.85	80.63	90.15	4.11
	Square†	60.16	61.26	47.15	78.73	88.48	4.46
	MCA*	72.47	37.68	20.67	56.67	72.12	8.89
	MReal-BDAI†*	74.02	52.62	40.03	68.85	79.15	6.76
	P1_P2†*	<u>74.91</u>	49.13	36.68	62.98	78.55	7.03
Leaderboard Results	LF	45.31	55.42	40.95	72.45	82.83	5.95
	HRE	45.46	54.16	39.93	70.45	81.50	6.41
	MN	47.50	55.49	40.98	72.30	83.30	5.92
	MN-Att	49.58	56.90	42.42	74.00	84.35	5.59
	LF-Att	49.76	57.07	42.08	74.82	85.05	5.41
	MS ConvAI	55.35	63.27	49.53	80.40	89.60	4.15
	UET-VNU†	57.40	59.50	45.50	76.33	85.82	5.34
	MVAN	59.37	64.84	<u>51.45</u>	81.12	90.65	3.97
	SGLNs†	61.27	59.97	45.68	77.12	87.10	4.85
	VisDial-BERT*	74.47	50.74	37.95	64.13	80.00	6.28
	Tohoku-CV†*	74.88	52.14	38.93	66.60	80.65	6.53
Ours	VD-BERT	59.96	**65.44**	**51.63**	**82.23**	<u>90.68</u>	**3.90**
	VD-BERT*	74.54	46.72	33.15	61.58	77.15	7.18
	VD-BERT†*	**75.35**	51.17	38.90	62.82	77.98	6.69

Table 6.1: Summary of results on the test-std split of VisDial v1.0 dataset. The results are reported by the test server. "†" denotes ensemble model and "∗" indicates fine-tuning on dense annotations. The "↑" denotes higher value for better performance and "↓" is the opposite. The best and second-best results in each column are in bold and underlined respectively.

Results on VisDial v1.0 test-std. We report the comparison results on VisDial v1.0 test-std split in Table 6.1 and make the following observations.

- *New state of the art for both single-model and ensemble settings.* Our single-model VD-BERT significantly outperforms all of its single-model counterparts across various metrics, even including some ensemble variants such as Synergistic, DAN (except R@10), and ReDAN (except NDCG). With further fine-tuning on dense annotations, the NDCG score increases quite sharply, from 59.96 to 74.54 with nearly 15% absolute improvement, setting a new state of the art in the single-model setting. This indicates that dense annotation fine-tuning plays a crucial role in boosting the NDCG scores. Moreover, our designed ensemble version yields new state-of-the-art results (**75.35** NDCG), outperforming the 2019 Visual Dialog challenge winner MReal-BDAI [116] (74.02 NDCG) by over 1.3 absolute points.

- *Inconsistency between NDCG and other metrics.* While dense annotation fine-tuning yields huge improvements on NDCG, we also notice that it has a severe countereffect on other metrics, e.g., reducing the MRR score from 65.44 to 46.72 for VD-BERT. Such a phenomenon has also been observed in other recent models, such as MReal-BDAI, VisDial-BERT, Tohoku-CV Lab, and P1_P2, whose NDCG scores surpass others without dense annotation fine-tuning by at least around 10% absolute points while other metrics drop dramatically. We provide a detailed analysis of this phenomenon in Section 6.5.4.

- *Our VD-BERT is simpler and more effective than VisDial-*

BERT. VisDial-BERT is a concurrent work to ours that also exploits vision-language pretrained models for visual dialog. It only reports the single-model performance of 74.47 NDCG. Compare to that, our VD-BERT achieves slightly better results (74.54 NDCG), however, note that we did not pretrain on large-scale external vision-language datasets like Conceptual Captions [131] and VQA [6] as VisDial-BERT does. Besides, while VisDial-BERT does not observe improvements by ensembling, we endeavor to design an effective ensemble strategy to further increase the NDCG score to 75.35 for VD-BERT (see Table 6.3d).

Results on VisDial v0.9 val. We further show both discriminative and generative results on v0.9 val split in Table 6.2. For comparison, we choose LF, HRE, HREA, MN [33], HCIAE [93], CoAtt [159], RvA, and DVAN as they contain results in both settings on the v0.9 val split. Our model continues to yield much better results in the discriminative setting (e.g., 70.04 MRR compared to DVAN's 66.67) and comparable results with the state of the art in the generative setting (e.g., 55.95 MRR score vs. DVAN's 55.94). This validates the effectiveness of our VD-BERT in both settings using a unified Transformer architecture. By contrast, VisDial-BERT can only support the discriminative setting.

Model	Discriminative Setting					Generative Setting				
	MRR↑	R@1↑	R@5↑	R@10↑	Mean ↓	MRR↑	R@1↑	R@5↑	R@10↑	Mean ↓
LF [33]	58.07	43.82	74.68	84.07	5.78	51.99	41.83	61.78	67.59	17.07
HRE [33]	58.46	44.67	74.50	84.22	5.72	52.37	42.29	62.18	67.92	17.07
HREA [33]	58.68	44.82	74.81	84.36	5.66	52.42	42.28	62.33	68.17	16.79
MN [33]	59.65	45.55	76.22	85.37	5.46	52.59	42.29	62.85	68.88	17.06
HCIAE [93]	62.22	48.48	78.75	87.59	4.81	54.67	44.35	65.28	71.55	14.23
CoAtt [159]	63.98	50.29	80.71	88.81	4.47	55.78	46.10	**65.69**	71.74	14.43
RvA [110]	66.34	52.71	82.97	90.73	**3.93**	55.43	45.37	65.27	**72.97**	**10.71**
DVAN [49]	66.67	53.62	82.85	90.72	**3.93**	55.94	46.58	65.50	71.25	14.79
VD-BERT	**70.04**	**57.79**	**85.34**	**92.68**	4.04	**55.95**	**46.83**	65.43	72.05	13.18

Table 6.2: Discriminative and generative results of various models on the val split of VisDial v0.9 dataset.

6.5.2 Ablation Study

We examine the effects of varying training settings and contexts, ranking optimizations, and ensemble strategies on VD-BERT. For this, we use VisDial v1.0 dataset in the discriminative setting.

(a) Training Settings. Table 6.3 (a) demonstrates how different training settings influence the results. We observe that initializing the model with weights from BERT indeed benefits the visual dialog task a lot, increasing the NDCG score by about 7% absolute over the model trained from scratch. Surprisingly, the model initialized with the weights from VLP that was pretrained on Conceptual Captions [131], does not work better than the one initialized from BERT. It might be due to the domain discrepancy between image captions and multi-turn dialogs, as well as the slightly different experiment settings (e.g., we extract 36 objects from image compared to their 100 objects). Another possible reason might be that the VisDial data with more than one million image-dialog turn pairs (as each image is associated with 10 dialog turns) can provide adequate contexts to adapt BERT for effective vision and dialog fusion. We also find that the visually grounded MLM is crucial for transferring BERT into the multimodal setting, indicated by a large performance drop when using only NSP.

(b) Training Contexts. We study the impact of varying the dialog context used for training (Table 6.3 (b)). With longer dialog history ("Full history"), our model indeed yields better results in most of the ranking metrics, while the one without using any dialog history obtains the highest NDCG score. This

	Model	NDCG↑	MRR↑	R@1↑	R@5↑	R@10↑	Mean ↓
(a)	From scratch	56.20	62.25	48.16	79.57	89.01	4.31
	Init from VLP	61.79	66.67	53.23	83.60	91.97	3.66
	Init from BERT	**63.22**	**67.44**	**54.02**	**83.96**	**92.33**	**3.53**
	↪ only NSP	55.89	63.15	48.98	80.45	89.72	4.15
(b)	No history	**64.70**	62.93	48.70	80.42	89.73	4.30
	One previous turn	63.47	65.30	51.66	82.30	90.97	3.86
	Full history	63.22	**67.44**	**54.02**	**83.96**	**92.33**	**3.53**
	↪ only text	54.32	62.79	48.48	80.12	89.33	4.27
(c)	CE	74.47	44.94	32.23	60.10	76.70	7.57
	ListNet	**74.54**	46.72	33.15	61.58	77.15	**7.18**
	ListMLE	72.96	36.81	20.70	54.60	73.28	8.90
	ApproxNDCG	72.45	**49.88**	**37.88**	**62.90**	**77.40**	7.26
(d)	EPOCH	74.84	47.40	34.30	61.58	77.78	7.12
	LENGTH	75.07	47.33	33.88	62.20	**78.50**	7.01
	RANK	75.13	50.00	38.28	60.93	77.28	6.90
	DIVERSE	**75.35**	**51.17**	**38.90**	**62.82**	77.98	**6.69**

Table 6.3: Extensive ablation studies: (a) various training settings and (b) training contexts on v1.0 val; (c) Dense annotation fine-tuning with varying ranking methods and (d) various ensemble strategies on v1.0 test-std.

indicates that dense relevance scores might be annotated with less consideration of dialog history. If we remove the visual cues from the "Full history" model, we see a drop in all metrics, especially, on NDCG. However, this version still obtains comparable results to the "No history" variant, revealing that textual information dominates the VisDial task.

(c) Ranking Optimization. In Table 6.3 (c), we compare Cross Entropy (CE) training with several other listwise ranking methods: ListNet [20], ListMLE [161], and approxNDCG [118].[3] Among these methods, ListNet yields the best NDCG and Mean

[3] `https://github.com/allegro/allRank`

Rank, while the approxNDCG achieves the best MRR and Recall on VisDial v1.0 test-std.

(d) Ensemble Strategy. We also explore ways to achieve the best ensemble performance with various model selection criteria in Table 6.3 (d). We consider three criteria, EPOCH, LENGTH, and RANK that respectively refer to predictions from different epochs of a single model, from different models trained with varying context lengths and with different ranking methods in Table 6.3 (b-c). We use four predictions from each criterion and combine their diverse predictions (DIVERSE) by summing up their normalized ranking scores. We observe that EPOCH contributes the least to the ensemble performance while RANK models are more helpful than LENGTH models. The diverse set of them leads to the best ensemble performance.

6.5.3 Attention Visualization of VD-BERT

We proceed to probe into the attention weights of our VD-BERT, aiming to analyze whether or not and how it achieves the effective vision and dialog fusion via the *visually grounded* training. We visualize their heatmaps for a validation example in Figure 6.3 and progressively dissect them below.

We first investigate whether the attention heads in our VD-BERT can be used for entity grounding. We visualize the attention weights on the top 10 detected objects in the image from its caption in Figure 6.3 (a). We observe that many heads at different layers can correctly ground some entities like `person` and `motorcycle` in the image, and even reveal some high-level semantic correlations such as `person`↔`motorcycle` (at L5H5 and L8H2) and `motorcycle`↔`street` (at L1H11). On the other

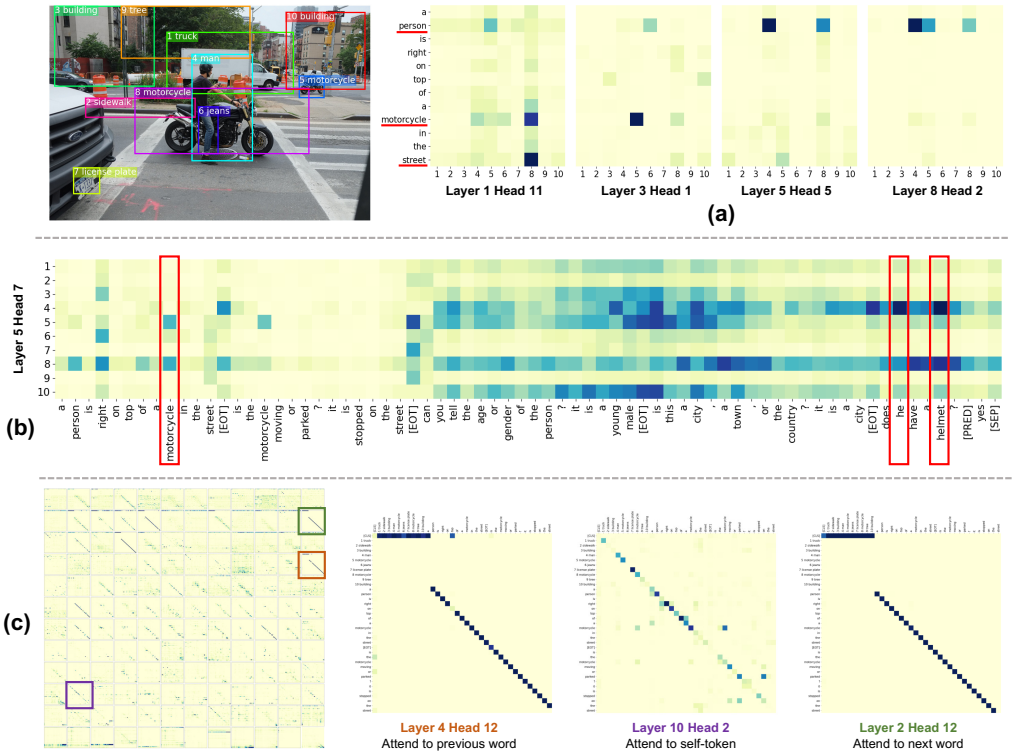


Figure 6.3: Attention weight visualization in our VD-BERT: (a) some selected heads at various layers capturing the image-caption alignment via grounding entities; (b) an attention heatmap showing the fusion of image and multi-turn dialog; (c) heatmaps of all 144 heads for both image and a single-turn dialog with some attention patterns.

hand, heads at higher layers tend to have a sharper focus on specific visual objects like the man and the motorcycles.

Next, we examine how VD-BERT captures the interactions between image and dialog history. In contrast to other vision-language tasks, visual dialog has a more complex multi-turn structure, thereby posing a hurdle for effective fusion. As shown in Figure 6.3 (b), VD-BERT can ground entities and discover some object relations, e.g., `helmet` is precisely related to the man and the motorcycle in the image (see the rightmost red box). More interestingly, it can even resolve visual pronoun

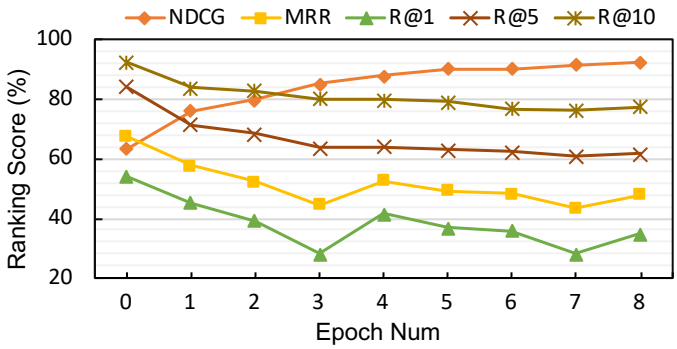


Figure 6.4: Various ranking scores across epochs of fine-tuning on dense annotations using ListNet.

coreference of `he` in the question to the man in the image (see the middle red box).

Finally, we analyze the self-attention weights for all layers and all heads for both image and dialog segments in Figure 6.3 (c). Instead of attempting to interpret all the 144 heads (12 layers and each layer has 12 heads), we analyze them in a holistic way. Compared to the words in the dialog, visual objects overall receive much less attention in most cases. This also explains the reason why relying solely on texts can still yield reasonably good results (Table 6.3 (b)). We also show three other apparent attention patterns: attentions that a token puts to its previous token, to itself, and to the next token. We see that the patterns for image and text are disparate (where image objects can hardly learn to attend previous/next tokens) as objects in image lack explicit orders like tokens in a text. We provide more attention visualization examples in Figure 6.6.

6.5.4 Fine-tuning on Dense Annotations

In this section, we focus on the effect of dense annotation fine-tuning. We first show how various metrics change for fine-

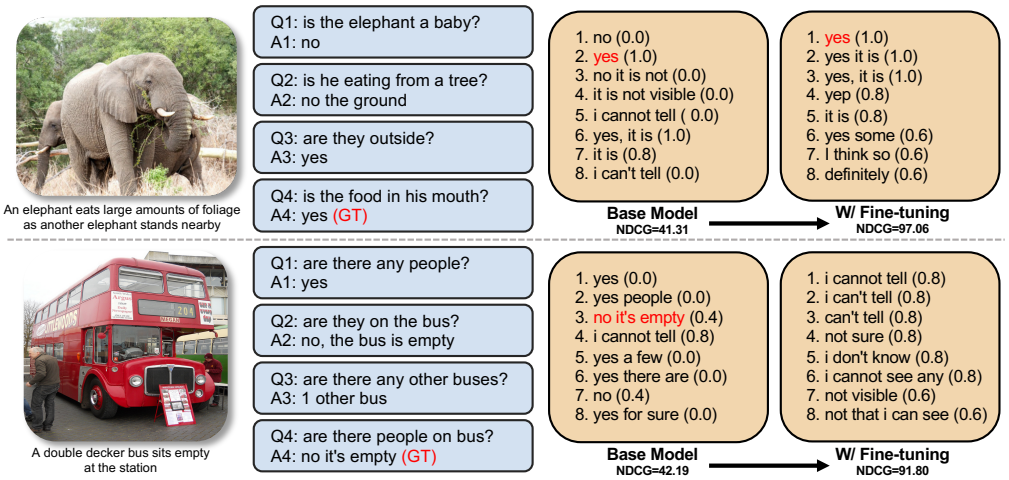


Figure 6.5: Two examples where relevant answer candidates are elevated into higher ranks after fine-tuning on dense annotations. GT: ground truth.

tuning in Figure 6.4. For this experiment, we randomly sample 200 instances from VisDial v1.0 val as the test data and use the rest for fine-tuning with the ListNet ranking method. We observe that NDCG keeps increasing with more epochs of fine-tuning, while other metrics such as Recall@K and MRR) drop. In the following, we explore the reason for this disparity between NDCG and other ranking metrics in depth.

Case Study. We provide two examples to qualitatively demonstrate how dense annotation fine-tuning results in better NDCG scores in Figure 6.5. For the example at the top, fine-tuning helps our model to assign higher ranks to the answers that share similar semantics with the ground truth answer and should also be regarded as correct ("yes, it is" and "yep" vs. "yes"). In the example at the bottom, we spot a mismatch between the sparse and dense annotations: the ground truth answer "no, it's empty" is only given 0.4 relevance score, while uncertain answers like "i don't know" are considered to be more

Models	All	Relevance Score				Question Type			
		1.0 (31%)	0.6∼0.8 (35%)	0.2∼0.4 (25%)	0.0 (9%)	Yes/no (76%)	Number (3%)	Color (11%)	Others (10%)
DAN	58.28	63.29	61.02	53.29	43.86	59.86	41.03	57.55	51.89
Ours	63.55	70.25	65.18	58.40	48.07	65.45	48.98	58.51	58.75
Ours (w/ ft)	89.62	95.38	89.76	84.63	82.84	91.05	74.41	84.00	89.12

Table 6.4: NDCG scores in VisDial v1.0 val split broken down into 4 groups based on either the relevance score or the question type. The % value in the parentheses denotes the corresponding data proportion.

relevant. In this case, fine-tuning instead makes our model fail to predict the correct answer despite the increase of NDCG score. We continue to quantitatively analyze how such annotation mismatches influence the NDCG results.

Relevance Score and Question Type Analysis. For further analysis, we classify the $2,064$ instances in VisDial v1.0 val set based on the ground-truth's relevance score and question type (Table 6.4). We consider 4 bins $\{0.0, 0.2 \sim 0.4, 0.6 \sim 0.8, 1.0\}$ for the relevance score and 4 question types: *Yes/no*, *Number*, *Color*, and *Others*. We then analyze the NDCG scores assigned by DAN [60] and our VD-BERT with and without dense annotation fine-tuning. We choose DAN as it achieves good NDCG scores (Table 6.1) and provides the source code to reproduce their predictions.

By examining the distribution of the relevance scores, we find that only 31% of them are aligned well with the sparse annotations and 9% are totally misaligned. As the degree of such mismatch increases (relevance score changes $1.0 \rightarrow 0.0$), both DAN and our model witness a plunge in NDCG ($63.29 \rightarrow 43.86$ and $70.25 \rightarrow 48.07$), while dense annotation fine-tuning significantly boosts NDCG scores for all groups, especially for

the most misaligned one (48.07 $\rightarrow$ 82.84 for our model). These results validate that the misalignment of the sparse and dense annotations is the key reason for the inconsistency between NDCG and other metrics.

In terms of question type, we observe that *Yes/no* is the major type (76%) and also the easiest one, while *Number* is the most challenging and least frequent one (3%). Our model outperforms DAN by over 10% in most of the question types except the *Color* type. Fine-tuning on dense annotations gives our model huge improvements across all the question types, especially for *Others* with over 30% absolute gain. We provide more qualitative comparison results in Figure 6.7.

6.6 Summary

In this chapter, we have presented VD-BERT, a unified vision-dialog Transformer model that exploits the pretrained BERT language models for visual dialog. VD-BERT is capable of modeling all the interactions between an image and a multi-turn dialog within a single-stream Transformer encoder and enables the effective fusion of features from both modalities via simple visually grounded training. Besides, it can either rank or generate answers seamlessly. Without pretraining on external vision-language datasets, our model establishes new state-of-the-art performance in the discriminative setting and shows promising results in the generative setting on the visual dialog benchmarks. We further conduct thorough experiments to analyze and interpret our model, providing insights for future transfer learning research on visual dialog tasks and even other cross-media understanding tasks.

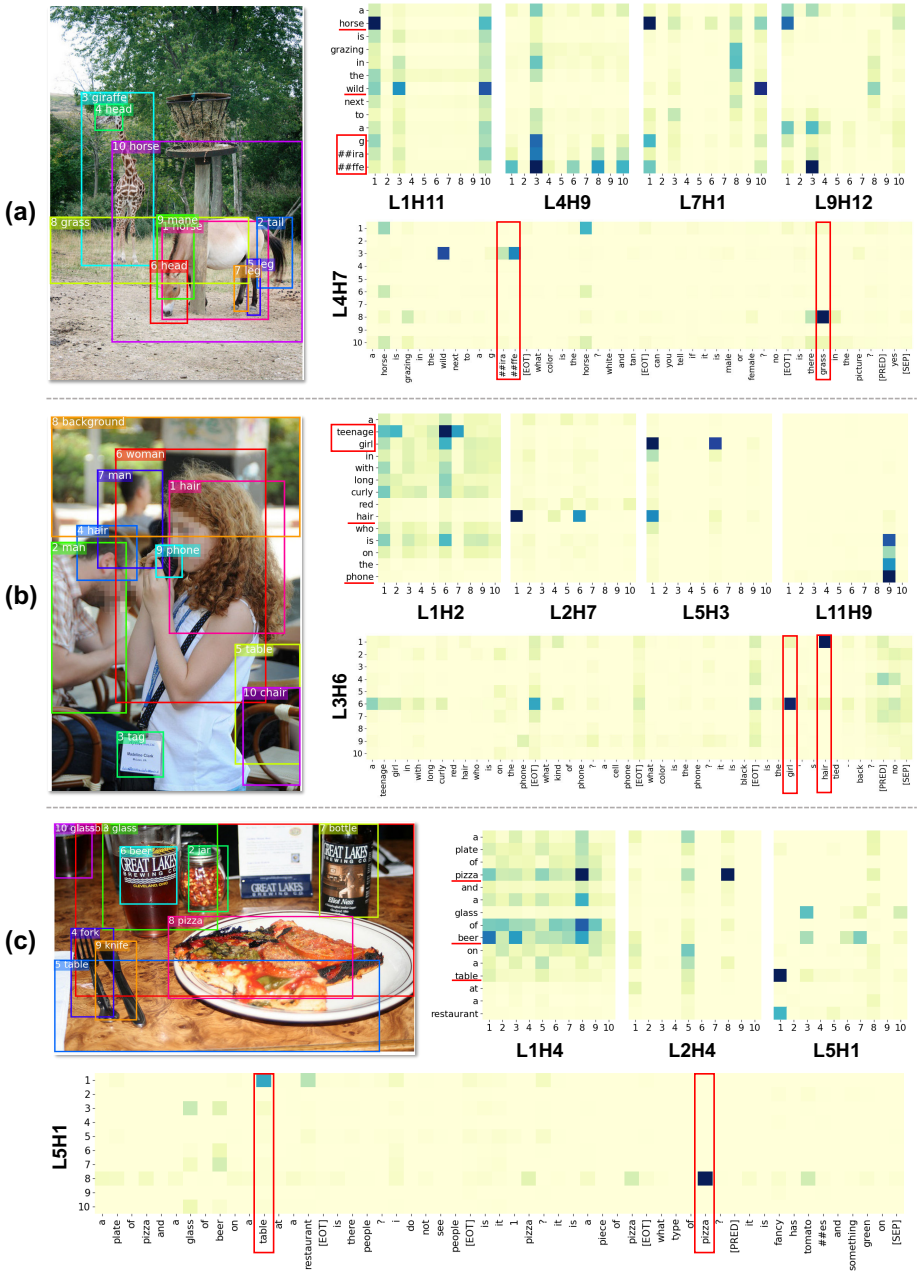


Figure 6.6: More attention visualization examples. LxHy: Layer x Head y ($1 \leq x, y \leq 12$). Our VD-BERT pretrained on the visual dialog data achieves effective fusion of vision and dialog contents, where some of its attention heads can precisely ground some entities between image and caption/multi-turn dialog: (a) `horse`, `wild`, and `giraffe`; (b) `teenage girl`, `hair`, and `phone`; (c) `pizza`, `beer`, and `table`.

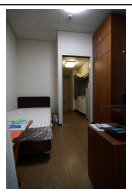
Image + Caption	GT Dialog	DAN	VD-BERT	VD-BERT (w/ft)
 a brown building stands at the corner of a street on a sunny day	Q1: is there any people in the photo? A1: yes far away Q2: are there any streets visible in the photo? A2: yes Q3: are there cars parked along the streets? A3: no Q4: what kind of buildings appear in the photo? A4: older architectural building like a library Q5: is there any signs visible in the photo? A5: there is a no entry sign Q6: are there traffic lights visible? A6: no Q7: can you see the sidewalk? A7: yes Q8: are the sidewalks clean? A8: immaculate Q9: are there any trees planted near the building? A9: there is a tree next to the building Q10: is the tree short or tall? A10: tall	Q1: is there any people in the photo? A1: yes Q2: are there any streets visible in the photo? A2: no on the street Q3: are there cars parked along the streets? A3: no Q4: what kind of buildings appear in the photo? A4: yes shops Q5: is there any signs visible in the photo? A5: yes Q6: are there traffic lights visible? A6: 1 light pole Q7: can you see the sidewalk? A7: yes Q8: are the sidewalks clean? A8: yes very clean Q9: are there any trees planted near the building? A9: yes Q10: is the tree short or tall? A10: tall	Q1: is there any people in the photo? A1: yes Q2: are there any streets visible in the photo? A2: yes Q3: are there cars parked along the streets? A3: no Q4: what kind of buildings appear in the photo? A4: older architectural building like a library Q5: is there any signs visible in the photo? A5: yes Q6: are there traffic lights visible? A6: no Q7: can you see the sidewalk? A7: yes Q8: are the sidewalks clean? A8: yes Q9: are there any trees planted near the building? A9: there is a tree next to the building Q10: is the tree short or tall? A10: tall	Q1: is there any people in the photo? A1: yes Q2: are there any streets visible in the photo? A2: yes Q3: are there cars parked along the streets? A3: no Q4: what kind of buildings appear in the photo? A4: not sure Q5: is there any signs visible in the photo? A5: yes Q6: are there traffic lights visible? A6: no Q7: can you see the sidewalk? A7: yes Q8: are the sidewalks clean? A8: yes, they are clean Q9: are there any trees planted near the building? A9: no there isn't Q10: is the tree short or tall? A10: tall
 man pointing at tv display in a store	Q1: is the picture in color? A1: yes Q2: how many people are in the picture? A2: i see 2 people Q3: how many tv's are in the picture? A3: 18 screens on the display but there are some in boxes as well Q4: is 1 of the people a worker in the store? A4: no it doesn't look like it Q5: is there an image on the tv that he is pointing at? A5: yes it is an cartoon Q6: is the man standing or sitting? A6: standing Q7: what is the brand of tv? A7: i am unable to tell Q8: can you tell what store it is? A8: no Q9: are there any other products in view? A9: i just see tv's and boxes Q10: what color is the floor? A10: blue	Q1: is the picture in color? A1: yes Q2: how many people are in the picture? A2: 2 Q3: how many tv's are in the picture? A3: 2 Q4: is 1 of the people a worker in the store? A4: yes, but they are taking a break Q5: is there an image on the tv that he is pointing at? A5: yes it is an cartoon Q6: is the man standing or sitting? A6: standing Q7: what is the brand of tv? A7: it's in japanese Q8: can you tell what store it is? A8: yes it is a list of rules Q9: are there any other products in view? A9: there is papers all over the desk so i would say yes Q10: what color is the floor? A10: blue	Q1: is the picture in color? A1: yes Q2: how many people are in the picture? A2: 2 Q3: how many tv's are in the picture? A3: 18 screens on the display but there are some in boxes as well Q4: is 1 of the people a worker in the store? A4: there are no visible workers Q5: is there an image on the tv that he is pointing at? A5: yes it is an cartoon Q6: is the man standing or sitting? A6: standing Q7: what is the brand of tv? A7: it's in japanese Q8: can you tell what store it is? A8: no Q9: are there any other products in view? A9: i just see tv's and boxes Q10: what color is the floor? A10: blue	Q1: is the picture in color? A1: yes it is Q2: how many people are in the picture? A2: 1 Q3: how many tv's are in the picture? A3: 18 screens on the display but there are some in boxes as well Q4: is 1 of the people a worker in the store? A4: there are no visible workers Q5: is there an image on the tv that he is pointing at? A5: yes Q6: is the man standing or sitting? A6: standing Q7: what is the brand of tv? A7: i can't tell Q8: can you tell what store it is? A8: no Q9: are there any other products in view? A9: yes Q10: what color is the floor? A10: blue
 a woman is smiling next to a decorated microwave	Q1: is there anyone in the room with the woman? A1: no Q2: is she young? A2: no Q3: what color is her hair? A3: black Q4: is it a big microwave? A4: yes Q5: is it sunny? A5: no Q6: is there any other appliances in the room? A6: no Q7: what is the woman wearing? A7: a shirt Q8: is her hair long? A8: no Q9: what color is the microwave? A9: white Q10: is she wearing glasses? A10: no	Q1: is there anyone in the room with the woman? A1: no, she is alone Q2: is she young? A2: no about 18 Q3: what color is her hair? A3: brown Q4: is it a big microwave? A4: normal size Q5: is it sunny? A5: no Q6: is there any other appliances in the room? A6: no Q7: what is the woman wearing? A7: brown jacket Q8: is her hair long? A8: yes Q9: what color is the microwave? A9: silver Q10: is she wearing glasses? A10: no	Q1: is there anyone in the room with the woman? A1: no Q2: is she young? A2: yes Q3: what color is her hair? A3: brown Q4: is it a big microwave? A4: yes Q5: is it sunny? A5: no Q6: is there any other appliances in the room? A6: no Q7: what is the woman wearing? A7: black jeans and a t shirt Q8: is her hair long? A8: no Q9: what color is the microwave? A9: white Q10: is she wearing glasses? A10: no	Q1: is there anyone in the room with the woman? A1: no Q2: is she young? A2: yes Q3: what color is her hair? A3: brown Q4: is it a big microwave? A4: yes Q5: is it sunny? A5: i can't tell Q6: is there any other appliances in the room? A6: not that i can see Q7: what is the woman wearing? A7: a black jacket Q8: is her hair long? A8: no Q9: what color is the microwave? A9: white Q10: is she wearing glasses? A10: no
 an elephant sprays the person on its back	Q1: is the elephant large? A1: yes, but doesn't look full grown Q2: how many people are on the elephant? A2: i can only see 1 Q3: are there more elephants in the picture? A3: no Q4: what is the person wearing? A4: i think it's shorts, hard to tell with the spray from the elephant Q5: where is the elephant getting the water from? A5: it's either a large puddle or maybe a river bank, not sure Q6: is the water clean or dirty? A6: it looks pretty muddy, so i would say dirty Q7: is there grass anywhere near the water or mainly mud? A7: mainly mud Q8: can you see the sky? A8: no Q9: are there any trees in the photo? A9: yes Q10: can you tell what type of trees? A10: no	Q1: is the elephant large? A1: yes Q2: how many people are on the elephant? A2: four, 1 in each photo Q3: are there more elephants in the picture? A3: no, just the 2 birds Q4: what is the person wearing? A4: a green button up Q5: where is the elephant getting the water from? A5: it's either a large puddle or maybe a river bank, not sure Q6: is the water clean or dirty? A6: it looks pretty muddy, so i would say dirty Q7: is there grass anywhere near the water or mainly mud? A7: there's a bit of grass, and a green reflection on the water Q8: can you see the sky? A8: a very little part of if through the trees Q9: are there any trees in the photo? A9: yes, but there's hardly any leaves Q10: can you tell what type of trees? A10: i would think late spring or summer cause all the trees are nice and green	Q1: is the elephant large? A1: yes Q2: how many people are on the elephant? A2: 2 Q3: are there more elephants in the picture? A3: no Q4: what is the person wearing? A4: a waterproof swimsuit thing Q5: where is the elephant getting the water from? A5: it's either a large puddle or maybe a river bank, not sure Q6: is the water clean or dirty? A6: it looks pretty muddy, so i would say dirty Q7: is there grass anywhere near the water or mainly mud? A7: mainly mud Q8: can you see the sky? A8: no Q9: are there any trees in the photo? A9: yes Q10: can you tell what type of trees? A10: no	Q1: is the elephant large? A1: yes Q2: how many people are on the elephant? A2: 1 Q3: are there more elephants in the picture? A3: no Q4: what is the person wearing? A4: he is wearing a jacket Q5: where is the elephant getting the water from? A5: cannot tell from the picture Q6: is the water clean or dirty? A6: it looks pretty muddy, so i would say dirty Q7: is there grass anywhere near the water or mainly mud? A7: no, there is 0 Q8: can you see the sky? A8: no Q9: are there any trees in the photo? A9: yes Q10: can you tell what type of trees? A10: no i can't
 a man wearing a hat holds a small white dog	Q1: age of man? A1: maybe 20s Q2: type of dog? A2: a white fluffy dog Q3: is it small? A3: yes Q4: can you see the head? A4: i can see the dogs head and the man's head Q5: is the mouth open on the dog? A5: no Q6: what else is the man wearing? A6: eyeglasses Q7: what color hat is it? A7: tan hat Q8: what color is his shirt? A8: hard to tell, possibly a grey vest Q9: what color is his pants? A9: they are not shown Q10: is the man outside? A10: no, he is inside	Q1: age of man? A1: 40s Q2: type of dog? A2: a white fluffy dog Q3: is it small? A3: average size Q4: can you see the head? A4: yes Q5: is the mouth open on the dog? A5: yes Q6: what else is the man wearing? A6: a jacket and a hat Q7: what color hat is it? A7: brown Q8: what color is his shirt? A8: black Q9: what color is his pants? A9: black Q10: is the man outside? A10: no, he is inside	Q1: age of man? A1: 40s Q2: type of dog? A2: a white fluffy dog Q3: is it small? A3: yes Q4: can you see the head? A4: i can see the dogs head and the man's head Q5: is the mouth open on the dog? A5: no Q6: what else is the man wearing? A6: a jacket and a hat Q7: what color hat is it? A7: tan hat Q8: what color is his shirt? A8: white Q9: what color is his pants? A9: they are not shown Q10: is the man outside? A10: no, he is inside	Q1: age of man? A1: 40s Q2: type of dog? A2: i can't tell Q3: is it small? A3: yes Q4: can you see the head? A4: yes Q5: is the mouth open on the dog? A5: nope Q6: what else is the man wearing? A6: eyeglasses Q7: what color hat is it? A7: it's brown Q8: what color is his shirt? A8: his shirt is white Q9: what color is his pants? A9: i can't see it Q10: is the man outside? A10: no
 a small bedroom with a hall with laundry in the background	Q1: do you see any people? A1: no Q2: is there a bed in the bedroom? A2: yes Q3: is it a big pile of laundry? A3: no Q4: what color is the bedspread? A4: white and light gray Q5: is it dirty laundry or clean? A5: clean Q6: is the hall carpeted or hard floor? A6: hard floor Q7: is the laundry hanging or piled or folded? A7: hanging Q8: is it a light or dark wood floor? A8: dark Q9: is the bedroom a child's room, guest room, or adult bedroom? A9: adult Q10: is there any other furniture in the bedroom? A10: yes	Q1: do you see any people? A1: no Q2: is there a bed in the bedroom? A2: no Q3: is it a big pile of laundry? A3: yes Q4: what color is the bedspread? A4: white and tan blankets Q5: is it dirty laundry or clean? A5: it's clean Q6: is the hall carpeted or hard floor? A6: wood Q7: is the laundry hanging or piled or folded? A7: hanging Q8: is it a light or dark wood floor? A8: light Q9: is the bedroom a child's room, guest room, or adult bedroom? A9: hard to say Q10: is there any other furniture in the bedroom? A10: a table and another bed	Q1: do you see any people? A1: no Q2: is there a bed in the bedroom? A2: yes Q3: is it a big pile of laundry? A3: no Q4: what color is the bedspread? A4: white Q5: is it dirty laundry or clean? A5: clean Q6: is the hall carpeted or hard floor? A6: hard floor Q7: is the laundry hanging or piled or folded? A7: hanging Q8: is it a light or dark wood floor? A8: light Q9: is the bedroom a child's room, guest room, or adult bedroom? A9: adult Q10: is there any other furniture in the bedroom? A10: a table and another bed	Q1: do you see any people? A1: no Q2: is there a bed in the bedroom? A2: yes Q3: is it a big pile of laundry? A3: yes Q4: what color is the bedspread? A4: white and light gray Q5: is it dirty laundry or clean? A5: clean Q6: is the hall carpeted or hard floor? A6: hard floor Q7: is the laundry hanging or piled or folded? A7: hanging Q8: is it a light or dark wood floor? A8: light Q9: is the bedroom a child's room, guest room, or adult bedroom? A9: adult Q10: is there any other furniture in the bedroom? A10: yes

Figure 6.7: More qualitative examples in VisDial v1.0 val split for three model variants: DAN [60], VD-BERT, and VD-BERT with dense annotation fine-tuning. The second column is for ground truth (GT) dialog.

Chapter 7

Conclusion and Future Work

In this chapter, we first summarize the contributions of this thesis and present potential future research directions.

7.1 Conclusion

The prominent use of social media platforms results in millions of user-generated messages produced every day. This thesis aims to automatically summarize the main content into a set of succinct keyphrases for a text-only or cross-media post to help users efficiently capture the core ideas from the massive amount of social media data. We propose to encode *implicit* contexts like latent topics and *explicit* contexts like user conversations and accompanying images to enrich features and design various neural network-based models for them. We conduct extensive experiments to demonstrate the effectiveness of our proposed approaches. In particular, we make the following contributions. In Chapter 3, we propose a topic-aware neural keyphrase generation approach for social media posts. Our approach consists of two components: the neural topic model to infer latent topics, and another one is the seq2seq model to generate keyphrase

sequences. The two components are integrated with carefully designed connections and can be jointly trained in an end-to-end manner. Experimental results on three newly constructed datasets from Twitter, Weibo, and StackExchange show that our model outperforms previous methods in keyphrase prediction, meanwhile generating more coherent topics.

In Chapter 4, we propose a sequence generation framework to predict keyphrases for microblogs. Our approach is able to generate rare and even new keyphrases compared to previous methods, which rely on extraction-based or generation-based models and cannot produce keyphrases out of the source post or the predefined candidate list. Moreover, we explicitly exploit user conversations initiated by the target post to enrich contexts and propose a bi-attention network to better model the interactions between them. Extensive experiments on two datasets from Twitter and Weibo validate the superiority of our model over state-of-the-art methods.

In Chapter 5, we propose a unified framework with multi-modality multi-head attention (M^3H-Att) and image wordings for cross-media keyphrase prediction. Considering the unique data nature in cross-media posts where images are diverse in terms of types and have a complicated relationship with texts, we propose to leverage image wordings distilled from the image and M^3H-Att to better capture the flexible text-image interactions. Moreover, we design a novel unified framework via extending the copy mechanism to adaptively aggregate classification outputs, aiming to couple the advantages of keyphrase classification and keyphrase generation. Extensive experiments on a large-scale text-image tweet dataset demonstrate our model's effectiveness in predicting more precise keyphrases and

being able to attend indicative information from various aspects in both modalities with our multi-head attention.
Finally, in Chapter 6, we explore how to better leverage the visual cues in a more challenging visual dialog task and propose VD-BERT to achieve the effective vision and dialog fusion. Via simple visually grounded training, our VD-BERT captures the intricate interactions between image and dialog within a single-stream Transformer encoder. Moreover, our model supports both answer ranking and answer generation seamlessly through the same architecture. Our model yields a new state of the art in discriminative settings and promising results in generative settings for visual dialog tasks.

7.2 Future Work

In this thesis, we propose a number of neural approaches to better predict keyphrases for social media posts, which can be applied or extended to solve other applications with similar settings. Besides, although the task of keyphrase generation for social media understandings is receiving growing attention in the recent decade, it is still a developing area with some critical issues that are not sufficiently addressed. We summarize the potential extensions of our approaches and future work about keyphrase generation for better social media understanding.

- **Extending the proposed approaches for other similar applications.** First, our topic-aware keyphrase generation model is a generic framework of incorporating latent topics for sequence generation, which can be easily extended to other text generation tasks where topic information could be useful, such as the text summarization,

question generation, and storytelling [145] tasks. Second, our idea of leveraging user conversations to enrich contexts could inspire other methods for tasks where user reviews or comments are available, such as text summarization for online forums and news websites. Similarly, [41] also exploits user comments for helping microblog summarization. Third, the ideas of encoding image wordings from social media images and employing multi-head attention to capture the complex text-image interactions can be borrowed and improve a lot of existing cross-media applications, such as multimedia event extraction [87], sarcasm detection [19], and text-image relation classification [142]. Last but not least, our VD-BERT can potentially benefit other vision-grounded language tasks, e.g., the video dialog tasks [75].

- **Exploiting vision-language pretraining for better cross-media understanding.** Pretraining models like BERT with self-supervised objectives have demonstrated their powerful representation learning capability and established state of the arts for numerous applications. In a more challenging visual dialog task, we have also shown that vision-language pretraining could help achieve the effective vision and dialog fusion and dramatically improve the performance. The straightforward future work would be to harness the power of vision-language pretraining for understanding cross-media posts. To the best of our knowledge, despite substantial progress made in vision-language pretraining recently, there is no prior work on extending it for cross-media understanding. With the crucial insights drawn from our cross-media keyphrase prediction work at hand, we are potentially able to design better cross-media

pretraining models by taking the unique characteristics of social media data into consideration.

- **Unsupervised learning for keyphrase prediction.** All the neural approaches proposed by this thesis require large-scale annotated training data. Although we walk around this challenge by employing the user annotated tags as the target keyphrases, their amounts might still be insufficient, e.g., only less than 15% tweets contain at least one hashtag [146, 64]. Besides, it would be costly to recruit human annotators to accomplish tagging tasks. One possible way is to devise unsupervised or semi-supervised learning algorithms to ease the need of labeled data. Unsupervised learning has demonstrated its effectiveness in many NLP tasks. For example, in neural machine translation, [73, 74] propose to rely only on monolingual data and employ back-translation techniques to align both sides, which achieves promising results. Another example is the unsupervised keyphrase extraction for scientific articles [13], where they map the document and extracted keyphrases into a shared high-dimensional embedding space, and then select the top related candidates by comparing their sentence embedding distances. Inspired by their success, it would be an interesting future work to combine both types of unsupervised learning techniques for social media keyphrase prediction.

Chapter 8

Publications during Ph.D. Study

1. **Yue Wang**, Shafiq Joty, Michael R. Lyu, Irwin King, Caiming Xiong, and Steven C.H. Hoi. *VD-BERT: A Unified Vision and Dialog Transformer with BERT*. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), Long Paper, 2020.

2. **Yue Wang**, Jing Li, Michael Lyu and Irwin King. *Cross-Media Keyphrase Prediction: A Unified Framework with Multi-Modality Multi-Head Attention and Image Wordings*. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), Long Paper, 2020.

3. **Yue Wang**, Jing Li, Hou Pong Chan, Irwin King, Michael R. Lyu, Shuming Shi. *Topic-Aware Neural Keyphrase Generation for Social Media Language*. In Proceedings of the 57th Conference of the Association for Computational Linguistics (ACL), Long Paper, 2019.

4. **Yue Wang**, Jing Li, Irwin King, Michael R. Lyu, Shuming Shi. *Microblog Hashtag Generation via Encoding Conversation Contexts*. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT), Long Paper, 2019.

5. Jian Li, **Yue Wang**, Michael R. Lyu, Irwin King. *Code Completion with Neural Attention and Pointer Networks*. In Proceedings of the 27th International Joint Conference on Artificial Intelligence (IJCAI), Long Paper, 2018.

□ End of chapter.

Bibliography

[1] Know your limit: The ideal length of every social media post. https://sproutsocial.com/insights/social-media-character-counter/.

[2] M. Abavisani, L. Wu, S. Hu, J. R. Tetreault, and A. Jaimes. Multimodal categorization of crisis events in social media. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, pages 14667–14677. IEEE, 2020.

[3] S. Agarwal, T. Bui, J. Lee, I. Konstas, and V. Rieser. History for visual dialog: Do we really need it? In D. Jurafsky, J. Chai, N. Schluter, and J. R. Tetreault, editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 8182–8197. Association for Computational Linguistics, 2020.

[4] C. Alberti, J. Ling, M. Collins, and D. Reitter. Fusion of detected objects in text for visual question answering. In K. Inui, J. Jiang, V. Ng, and X. Wan, editors, *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint*

Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019, pages 2131–2140. Association for Computational Linguistics, 2019.

[5] P. Anderson, X. He, C. Buehler, D. Teney, M. Johnson, S. Gould, and L. Zhang. Bottom-up and top-down attention for image captioning and visual question answering. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pages 6077–6086. IEEE Computer Society, 2018.

[6] S. Antol, A. Agrawal, J. Lu, M. Mitchell, D. Batra, C. L. Zitnick, and D. Parikh. VQA: visual question answering. In *2015 IEEE International Conference on Computer Vision, ICCV 2015, Santiago, Chile, December 7-13, 2015*, pages 2425–2433, 2015.

[7] L. J. Ba, J. R. Kiros, and G. E. Hinton. Layer normalization. *CoRR*, abs/1607.06450, 2016.

[8] D. Bahdanau, K. Cho, and Y. Bengio. Neural machine translation by jointly learning to align and translate. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.

[9] H. Bai, Z. Chen, M. R. Lyu, I. King, and Z. Xu. Neural relational topic models for scientific article analysis. In *Proceedings of ACM International Conference on Information and Knowledge Management*, 2018.

[10] P. Bansal, S. Jain, and V. Varma. Towards semantic retrieval of hashtags in microblogs. In *World Wide Web Conference*, 2015.

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

[12] C. Baziotis, N. Pelekis, and C. Doulkeridis. Datastories at semeval-2017 task 4: Deep LSTM with attention for message-level and topic-based sentiment analysis. In *Proceedings of the 10th International Workshop on Semantic Evaluation, SemEval@NAACL-HLT 2016, San Diego, CA, USA, June 16-17, 2016*, pages 747–754, 2017.

[13] K. Bennani-Smires, C. Musat, A. Hossmann, M. Baeriswyl, and M. Jaggi. Simple unsupervised keyphrase extraction using sentence embeddings. In A. Korhonen and I. Titov, editors, *Proceedings of the 22nd Conference on Computational Natural Language Learning, CoNLL 2018, Brussels, Belgium, October 31 - November 1, 2018*, pages 221–229. Association for Computational Linguistics, 2018.

[14] D. M. Blei, A. Kucukelbir, and J. D. McAuliffe. Variational inference: A review for statisticians. *CoRR*, abs/1601.00670, 2016.

[15] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *Journal of Machine Learning Research*, 2003.

[16] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov. Enriching word vectors with subword information. *Trans. Assoc. Comput. Linguistics*, 5:135–146, 2017.

[17] J. Bollen, H. Mao, and X. Zeng. Twitter mood predicts the stock market. *J. Comput. Sci.*, 2(1):1–8, 2011.

[18] Buffer. What 1 million tweets taught us about how people tweet successfully. `https://blog.bufferapp.com/twitter-data-1-million-tweets`.

[19] Y. Cai, H. Cai, and X. Wan. Multi-modal sarcasm detection in twitter with hierarchical fusion model. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 2506–2515, 2019.

[20] Z. Cao, T. Qin, T. Liu, M. Tsai, and H. Li. Learning to rank: from pairwise approach to listwise approach. In Z. Ghahramani, editor, *Machine Learning, Proceedings of the Twenty-Fourth International Conference (ICML 2007), Corvallis, Oregon, USA, June 20-24, 2007*, volume 227 of *ACM International Conference Proceeding Series*, pages 129–136. ACM, 2007.

[21] R. Caruana, S. Lawrence, and C. L. Giles. Overfitting in neural nets: Backpropagation, conjugate gradient, and early stopping. In *Advances in Neural Information Processing Systems 13, Papers from Neural Information Processing Systems (NIPS) 2000, Denver, CO, USA*, pages 402–408, 2000.

[22] H. P. Chan, W. Chen, L. Wang, and I. King. Neural keyphrase generation via reinforcement learning with adaptive rewards. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 2163–2174, 2019.

[23] Y. Chang, X. Wang, Q. Mei, and Y. Liu. Towards twitter context summarization with user influence models. In *International Conference on Web Search and Data Mining*, 2013.

[24] J. Chen, X. Zhang, Y. Wu, Z. Yan, and Z. Li. Keyphrase generation with correlation constraints. In E. Riloff, D. Chiang, J. Hockenmaier, and J. Tsujii, editors, *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 4057–4066. Association for Computational Linguistics, 2018.

[25] T. Chen, X. He, and M. Kan. Context-aware image tweet modelling and recommendation. In A. Hanjalic, C. Snoek, M. Worring, D. C. A. Bulterman, B. Huet, A. Kelliher, Y. Kompatsiaris, and J. Li, editors, *Proceedings of the 2016 ACM Conference on Multimedia Conference, MM 2016, Amsterdam, The Netherlands, October 15-19, 2016*, pages 1018–1027. ACM, 2016.

[26] W. Chen, H. P. Chan, P. Li, L. Bing, and I. King. An integrated approach for keyphrase generation via exploring the power of retrieval and extraction. In J. Burstein, C. Doran, and T. Solorio, editors, *Proceedings of the 2019 Conference*

of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers), pages 2846–2856. Association for Computational Linguistics, 2019.

[27] W. Chen, Y. Gao, J. Zhang, I. King, and M. R. Lyu. Title-guided encoding for keyphrase generation. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019*, pages 6268–6275, 2019.

[28] Y. Chen, L. Li, L. Yu, A. E. Kholy, F. Ahmed, Z. Gan, Y. Cheng, and J. Liu. UNITER: learning universal image-text representations. *CoRR*, abs/1909.11740, 2019.

[29] D. Chinnappa, S. Murugan, and E. Blanco. Extracting possessions from social media: Images complement language. In K. Inui, J. Jiang, V. Ng, and X. Wan, editors, *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pages 663–672. Association for Computational Linguistics, 2019.

[30] K. Cho, B. van Merrienboer, Ç. Gülçehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio. Learning phrase representations using RNN encoder-decoder for statistical

machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 1724–1734, 2014.

[31] Y. Chuang, C. Liu, and H. Lee. Speechbert: Cross-modal pre-trained language model for end-to-end spoken question answering. *CoRR*, abs/1910.11559, 2019.

[32] Colah. Understanding lstm networks. `https://colah.github.io/posts/2015-08-Understanding-LSTMs/`.

[33] A. Das, S. Kottur, K. Gupta, A. Singh, D. Yadav, J. M. F. Moura, D. Parikh, and D. Batra. Visual dialog. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pages 1080–1089, 2017.

[34] D. Davidov, O. Tsur, and A. Rappoport. Enhanced sentiment learning using twitter hashtags and smileys. In *International Conference on Computational Linguistics*, 2010.

[35] J. Devlin, M. Chang, K. Lee, and K. Toutanova. 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, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186, 2019.

[36] L. Dong, N. Yang, W. Wang, F. Wei, X. Liu, Y. Wang, J. Gao, M. Zhou, and H. Hon. Unified language model pre-training for natural language understanding and generation. In H. M. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. B. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, 8-14 December 2019, Vancouver, BC, Canada*, pages 13042–13054, 2019.

[37] M. Efron. Hashtag retrieval in a microblogging environment. In *Conference on Research and Development in Information Retrieval*, 2010.

[38] G. Farnadi, J. Tang, M. D. Cock, and M. Moens. User profiling through deep multimodal fusion. In Y. Chang, C. Zhai, Y. Liu, and Y. Maarek, editors, *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining, WSDM 2018, Marina Del Rey, CA, USA, February 5-9, 2018*, pages 171–179. ACM, 2018.

[39] E. Frank, G. W. Paynter, I. H. Witten, C. Gutwin, and C. G. Nevill-Manning. Domain-specific keyphrase extraction. In T. Dean, editor, *Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence, IJCAI 99, Stockholm, Sweden, July 31 - August 6, 1999. 2 Volumes, 1450 pages*, pages 668–673. Morgan Kaufmann, 1999.

[40] Z. Gan, Y. Cheng, A. E. Kholy, L. Li, J. Liu, and J. Gao. Multi-step reasoning via recurrent dual attention for visual dialog. In *Proceedings of the 57th Conference of*

the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers, pages 6463–6474, 2019.

[41] S. Gao, X. Chen, P. Li, Z. Ren, L. Bing, D. Zhao, and R. Yan. Abstractive text summarization by incorporating reader comments. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019*, pages 6399–6406. AAAI Press, 2019.

[42] Y. Gao, P. Li, I. King, and M. R. Lyu. Interconnected question generation with coreference alignment and conversation flow modeling. In A. Korhonen, D. R. Traum, and L. Màrquez, editors, *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 4853–4862. Association for Computational Linguistics, 2019.

[43] S. D. Gollapalli and C. Caragea. Extracting keyphrases from research papers using citation networks. In C. E. Brodley and P. Stone, editors, *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence, July 27 -31, 2014, Québec City, Québec, Canada*, pages 1629–1635. AAAI Press, 2014.

[44] S. D. Gollapalli, X. Li, and P. Yang. Incorporating expert knowledge into keyphrase extraction. In *Proceedings of the*

Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA., pages 3180–3187, 2017.

[45] Y. Gong and Q. Zhang. Hashtag recommendation using attention-based convolutional neural network. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, IJCAI 2016, New York, NY, USA, 9-15 July 2016*, pages 2782–2788, 2016.

[46] Y. Gong, Q. Zhang, and X. Huang. Hashtag recommendation using dirichlet process mixture models incorporating types of hashtags. In *Empirical Methods in Natural Language Processing*, 2015.

[47] I. J. Goodfellow, Y. Bengio, and A. C. Courville. *Deep Learning*. Adaptive computation and machine learning. MIT Press, 2016.

[48] J. Gu, Z. Lu, H. Li, and V. O. K. Li. Incorporating copying mechanism in sequence-to-sequence learning. In *Association for Computational Linguistics*, 2016.

[49] D. Guo, H. Wang, and M. Wang. Dual visual attention network for visual dialog. In S. Kraus, editor, *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, August 10-16, 2019*, pages 4989–4995. ijcai.org, 2019.

[50] D. Guo, H. Wang, H. Zhang, Z. Zha, and M. Wang. Iterative context-aware graph inference for visual dialog. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, pages 10052–10061. IEEE, 2020.

[51] D. Guo, C. Xu, and D. Tao. Image-question-answer synergistic network for visual dialog. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pages 10434–10443, 2019.

[52] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 770–778. IEEE Computer Society, 2016.

[53] K. He, X. Zhang, S. Ren, and J. Sun. Identity mappings in deep residual networks. In *Computer Vision - ECCV 2016 - 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part IV*, pages 630–645, 2016.

[54] P. Heymann, D. Ramage, and H. Garcia-Molina. Social tag prediction. In *Conference on Research and Development in Information Retrieval*, 2008.

[55] L. Hong, A. Ahmed, S. Gurumurthy, A. J. Smola, and K. Tsioutsiouliklis. Discovering geographical topics in the twitter stream. In *World Wide Web Conference*, 2012.

[56] R. Hu, J. Andreas, M. Rohrbach, T. Darrell, and K. Saenko. Learning to reason: End-to-end module networks for visual question answering. In *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*, pages 804–813. IEEE Computer Society, 2017.

[57] H. Huang, Q. Zhang, Y. Gong, and X. Huang. Hashtag recommendation using end-to-end memory networks with hierarchical attention. In *COLING 2016, 26th International Conference on Computational Linguistics, Proceedings of the Conference: Technical Papers, December 11-16, 2016, Osaka, Japan*, pages 943–952, 2016.

[58] A. Hulth. Improved automatic keyword extraction given more linguistic knowledge. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing, EMNLP 2003, Sapporo, Japan, July 11-12, 2003*, 2003.

[59] X. Jiang, J. Yu, Z. Qin, Y. Zhuang, X. Zhang, Y. Hu, and Q. Wu. Dualvd: An adaptive dual encoding model for deep visual understanding in visual dialogue. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 11125–11132. AAAI Press, 2020.

[60] G. Kang, J. Lim, and B. Zhang. Dual attention networks for visual reference resolution in visual dialog. In K. Inui, J. Jiang, V. Ng, and X. Wan, editors, *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pages 2024–2033. Association for Computational Linguistics, 2019.

[61] G. Kang, J. Park, H. Lee, B. Zhang, and J. Kim. Dialgraph: Sparse graph learning networks for visual dialog. *CoRR*, abs/2004.06698, 2020.

[62] A. Karpathy and F. Li. Deep visual-semantic alignments for generating image descriptions. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7-12, 2015*, pages 3128–3137. IEEE Computer Society, 2015.

[63] S. Kazemzadeh, V. Ordonez, M. Matten, and T. L. Berg. Referitgame: Referring to objects in photographs of natural scenes. In A. Moschitti, B. Pang, and W. Daelemans, editors, *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 787–798. ACL, 2014.

[64] E. Khabiri, J. Caverlee, and K. Y. Kamath. Predicting semantic annotations on the real-time web. In *ACM Conference on Hypertext and Social Media*, 2012.

[65] H. Kim, H. Tan, and M. Bansal. Modality-balanced models for visual dialogue. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 8091–8098. AAAI Press, 2020.

[66] J. Kim, J. Jun, and B. Zhang. Bilinear attention networks. In S. Bengio, H. M. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, 3-8 December 2018, Montréal, Canada*, pages 1571–1581, 2018.

[67] Y. Kim, Y. Jernite, D. A. Sontag, and A. M. Rush. Character-aware neural language models. In D. Schuurmans and M. P. Wellman, editors, *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, February 12-17, 2016, Phoenix, Arizona, USA*, pages 2741–2749. AAAI Press, 2016.

[68] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.

[69] D. P. Kingma and M. Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[70] G. Klein, Y. Kim, Y. Deng, J. Senellart, and A. M. Rush. Opennmt: Open-source toolkit for neural machine translation. In *Association for Computational Linguistics*, 2017.

[71] S. Kottur, J. M. F. Moura, D. Parikh, D. Batra, and M. Rohrbach. Visual coreference resolution in visual dialog using neural module networks. In *Computer Vision - ECCV 2018 - 15th European Conference, Munich, Ger-*

many, September 8-14, 2018, Proceedings, Part XV, pages 160–178, 2018.

[72] R. Krishna, Y. Zhu, O. Groth, J. Johnson, K. Hata, J. Kravitz, S. Chen, Y. Kalantidis, L. Li, D. A. Shamma, M. S. Bernstein, and L. Fei-Fei. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *Int. J. Comput. Vis.*, 123(1):32–73, 2017.

[73] G. Lample, A. Conneau, L. Denoyer, and M. Ranzato. Unsupervised machine translation using monolingual corpora only. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018.

[74] G. Lample, M. Ott, A. Conneau, L. Denoyer, and M. Ranzato. Phrase-based & neural unsupervised machine translation. In E. Riloff, D. Chiang, J. Hockenmaier, and J. Tsujii, editors, *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 5039–5049. Association for Computational Linguistics, 2018.

[75] H. Le, D. Sahoo, N. F. Chen, and S. C. H. Hoi. Multimodal transformer networks for end-to-end video-grounded dialogue systems. In A. Korhonen, D. R. Traum, and L. Màrquez, editors, *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 5612–5623. Association for Computational Linguistics, 2019.

[76] T. T. N. Le, M. L. Nguyen, and A. Shimazu. Unsupervised keyphrase extraction: Introducing new kinds of words to keyphrases. In *AI 2016: Advances in Artificial Intelligence - 29th Australasian Joint Conference, Hobart, TAS, Australia, December 5-8, 2016, Proceedings*, pages 665–671, 2016.

[77] Y. LeCun, B. E. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. E. Hubbard, and L. D. Jackel. Backpropagation applied to handwritten zip code recognition. *Neural Computation*, 1(4):541–551, 1989.

[78] K. Lee, D. Palsetia, R. Narayanan, M. M. A. Patwary, A. Agrawal, and A. Choudhary. Twitter trending topic classification. In *2011 IEEE 11th International Conference on Data Mining Workshops*, pages 251–258. IEEE, 2011.

[79] G. Li, N. Duan, Y. Fang, M. Gong, and D. Jiang. Unicoder-vl: A universal encoder for vision and language by cross-modal pre-training. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 11336–11344. AAAI Press, 2020.

[80] J. Li, M. Galley, C. Brockett, J. Gao, and B. Dolan. A diversity-promoting objective function for neural conversation models. In K. Knight, A. Nenkova, and O. Rambow, editors, *NAACL HLT 2016, The 2016 Conference of the North American Chapter of the Association for Compu-*

tational Linguistics: Human Language Technologies, San Diego California, USA, June 12-17, 2016, pages 110–119. The Association for Computational Linguistics, 2016.

[81] J. Li, W. Gao, Z. Wei, B. Peng, and K. Wong. Using content-level structures for summarizing microblog repost trees. In *Empirical Methods in Natural Language Processing*, 2015.

[82] J. Li, Y. Gao, L. Bing, I. King, and M. R. Lyu. Improving question generation with to the point context. In K. Inui, J. Jiang, V. Ng, and X. Wan, editors, *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pages 3214–3224. Association for Computational Linguistics, 2019.

[83] J. Li, M. Liao, W. Gao, Y. He, and K. Wong. Topic extraction from microblog posts using conversation structures. In *Association for Computational Linguistics*, 2016.

[84] J. Li, Y. Song, Z. Wei, and K. Wong. A joint model of conversational discourse and latent topics on microblogs. *Journal of Computational Linguistics*, 2018.

[85] J. Li, Y. Song, Z. Wei, and K.-F. Wong. A joint model of conversational discourse and latent topics on microblogs. *Computational Linguistics*, 2018.

[86] L. H. Li, M. Yatskar, D. Yin, C. Hsieh, and K. Chang. Visualbert: A simple and performant baseline for vision and language. *CoRR*, abs/1908.03557, 2019.

[87] M. Li, A. Zareian, Q. Zeng, S. Whitehead, D. Lu, H. Ji, and S. Chang. Cross-media structured common space for multimedia event extraction. In D. Jurafsky, J. Chai, N. Schluter, and J. R. Tetreault, editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 2557–2568. Association for Computational Linguistics, 2020.

[88] C.-Y. Lin. Rouge: A package for automatic evaluation of summaries. In *Text Summarization Branches Out: Proceedings of the Association for Computational Linguistics-04 Workshop*, 2004.

[89] T. Lin, M. Maire, S. J. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft COCO: common objects in context. In *Computer Vision - ECCV 2014 - 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V*, pages 740–755, 2014.

[90] Z. Liu, X. Chen, Y. Zheng, and M. Sun. Automatic keyphrase extraction by bridging vocabulary gap. In *Proceedings of the Fifteenth Conference on Computational Natural Language Learning, CoNLL 2011, Portland, Oregon, USA, June 23-24, 2011*, pages 135–144, 2011.

[91] P. Lopez and L. Romary. HUMB: automatic key term extraction from scientific articles in GROBID. In K. Erk and C. Strapparava, editors, *Proceedings of the 5th International Workshop on Semantic Evaluation, SemEval@ACL 2010, Uppsala University, Uppsala, Sweden, July 15-*

16, 2010, pages 248–251. The Association for Computer Linguistics, 2010.

[92] J. Lu, D. Batra, D. Parikh, and S. Lee. Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. In H. M. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. B. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, 8-14 December 2019, Vancouver, BC, Canada*, pages 13–23, 2019.

[93] J. Lu, A. Kannan, J. Yang, D. Parikh, and D. Batra. Best of both worlds: Transferring knowledge from discriminative learning to a generative visual dialog model. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, 4-9 December 2017, Long Beach, CA, USA*, pages 314–324, 2017.

[94] J. Lu, J. Yang, D. Batra, and D. Parikh. Hierarchical question-image co-attention for visual question answering. In *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*, pages 289–297, 2016.

[95] Y. Luan, M. Ostendorf, and H. Hajishirzi. Scientific information extraction with semi-supervised neural tagging. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017, Copen-*

hagen, Denmark, September 9-11, 2017, pages 2641–2651, 2017.

[96] T. Luong, H. Pham, and C. D. Manning. Effective approaches to attention-based neural machine translation. In L. Màrquez, C. Callison-Burch, J. Su, D. Pighin, and Y. Marton, editors, *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP 2015, Lisbon, Portugal, September 17-21, 2015*, pages 1412–1421. The Association for Computational Linguistics, 2015.

[97] C. D. Manning, P. Raghavan, and H. Schütze. *Introduction to information retrieval*. Cambridge University Press, 2008.

[98] O. Medelyan, E. Frank, and I. H. Witten. Human-competitive tagging using automatic keyphrase extraction. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing, EMNLP 2009, 6-7 August 2009, Singapore, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 1318–1327, 2009.

[99] O. Medelyan, I. H. Witten, and D. Milne. Topic indexing with wikipedia. In *Proceedings of the AAAI WikiAI workshop*, volume 1, pages 19–24, 2008.

[100] R. Meng, S. Zhao, S. Han, D. He, P. Brusilovsky, and Y. Chi. Deep keyphrase generation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers*, pages 582–592, 2017.

[101] X. Meng, F. Wei, X. Liu, M. Zhou, S. Li, and H. Wang. Entity-centric topic-oriented opinion summarization in twitter. In *Proceedings of ACM International Conference on Knowledge Discovery and Data Mining*, 2012.

[102] Y. Miao, E. Grefenstette, and P. Blunsom. Discovering discrete latent topics with neural variational inference. In *Proceedings of International Conference on Machine Learning*, 2017.

[103] R. Mihalcea and P. Tarau. Textrank: Bringing order into text. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing , EMNLP 2004, A meeting of SIGDAT, a Special Interest Group of the ACL, held in conjunction with ACL 2004, 25-26 July 2004, Barcelona, Spain*, pages 404–411, 2004.

[104] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In C. J. C. Burges, L. Bottou, Z. Ghahramani, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States*, pages 3111–3119, 2013.

[105] V. Murahari, D. Batra, D. Parikh, and A. Das. Large-scale pretraining for visual dialog: A simple state-of-the-art baseline. *CoRR*, abs/1912.02379, 2019.

[106] R. Nallapati, B. Zhou, C. N. dos Santos, Ç. Gülçehre, and B. Xiang. Abstractive text summarization using

sequence-to-sequence rnns and beyond. In Y. Goldberg and S. Riezler, editors, *Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning, CoNLL 2016, Berlin, Germany, August 11-12, 2016*, pages 280–290. ACL, 2016.

[107] T. D. Nguyen and M. Kan. Keyphrase extraction in scientific publications. In *Asian Digital Libraries. Looking Back 10 Years and Forging New Frontiers, 10th International Conference on Asian Digital Libraries, ICADL 2007, Hanoi, Vietnam, December 10-13, 2007, Proceedings*, pages 317–326, 2007.

[108] T. H. Nguyen and K. Shirai. Topic modeling based sentiment analysis on social media for stock market prediction. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, ACL 2015, July 26-31, 2015, Beijing, China, Volume 1: Long Papers*, pages 1354–1364. The Association for Computer Linguistics, 2015.

[109] V. Nguyen, M. Suganuma, and T. Okatani. Efficient attention mechanism for handling all the interactions between many inputs with application to visual dialog. *CoRR*, abs/1911.11390, 2019.

[110] Y. Niu, H. Zhang, M. Zhang, J. Zhang, Z. Lu, and J. Wen. Recursive visual attention in visual dialog. In *IEEE Conference on Computer Vision and Pattern Recognition,*

CVPR 2019, Long Beach, CA, USA, June 16-20, 2019, pages 6679–6688, 2019.

[111] C. C. Park, B. Kim, and G. Kim. Towards personalized image captioning via multimodal memory networks. *IEEE Trans. Pattern Anal. Mach. Intell.*, 41(4):999–1012, 2019.

[112] S. Park, T. Whang, Y. Yoon, and H. Lim. Multi-view attention networks for visual dialog. *CoRR*, abs/2004.14025, 2020.

[113] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer. Automatic differentiation in pytorch. In *Proceedings of Neural Information Processing Systems*, 2017.

[114] J. Pennington, R. Socher, and C. D. Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 1532–1543, 2014.

[115] M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer. Deep contextualized word representations. In M. A. Walker, H. Ji, and A. Stent, editors, *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2018, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 1 (Long Papers)*, pages 2227–2237. Association for Computational Linguistics, 2018.

[116] J. Qi, Y. Niu, J. Huang, and H. Zhang. Two causal principles for improving visual dialog. *CoRR*, abs/1911.10496, 2019.

[117] J. Qi, Y. Niu, J. Huang, and H. Zhang. Two causal principles for improving visual dialog. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, pages 10857–10866. IEEE, 2020.

[118] T. Qin, T. Liu, and H. Li. A general approximation framework for direct optimization of information retrieval measures. *Inf. Retr.*, 13(4):375–397, 2010.

[119] A. Radford, K. Narasimhan, T. Salimans, and I. Sutskever. Improving language understanding by generative pre-training. 2018.

[120] S. Ren, K. He, R. B. Girshick, and J. Sun. Faster R-CNN: towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, pages 91–99, 2015.

[121] M. Röder, A. Both, and A. Hinneburg. Exploring the space of topic coherence measures. In *Proceedings of ACM International Conference on Web Search and Data Mining*, 2015.

[122] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Learning representations by back-propagating errors. *nature*, 323(6088):533–536, 1986.

[123] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. S. Bernstein, A. C. Berg, and F. Li. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015.

[124] D. Ruths and J. Pfeffer. Social media for large studies of behavior. *Journal of Science*, 2014.

[125] I. Schwartz, S. Yu, T. Hazan, and A. G. Schwing. Factor graph attention. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pages 2039–2048, 2019.

[126] S. Sedhai and A. Sun. Hashtag recommendation for hyperlinked tweets. In *Conference on Research and Development in Information Retrieval*, 2014.

[127] A. See, P. J. Liu, and C. D. Manning. Get to the point: Summarization with pointer-generator networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers*, pages 1073–1083, 2017.

[128] R. Sennrich, B. Haddow, and A. Birch. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers*. The Association for Computer Linguistics, 2016.

[129] M. J. Seo, A. Kembhavi, A. Farhadi, and H. Hajishirzi. Bidirectional attention flow for machine comprehension.

In *International Conference on Learning Representations*, 2016.

[130] P. H. Seo, A. M. Lehrmann, B. Han, and L. Sigal. Visual reference resolution using attention memory for visual dialog. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, 4-9 December 2017, Long Beach, CA, USA*, pages 3719–3729, 2017.

[131] P. Sharma, N. Ding, S. Goodman, and R. Soricut. Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 1: Long Papers*, pages 2556–2565, 2018.

[132] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.

[133] R. Smith. An overview of the tesseract OCR engine. In *9th International Conference on Document Analysis and Recognition (ICDAR 2007), 23-26 September, Curitiba, Paraná, Brazil*, pages 629–633, 2007.

[134] A. Srivastava and C. Sutton. Autoencoding variational inference for topic models. *arXiv preprint arXiv:1703.01488*, 2017.

[135] W. Su, X. Zhu, Y. Cao, B. Li, L. Lu, F. Wei, and J. Dai. VL-BERT: pre-training of generic visual-linguistic representations. *CoRR*, abs/1908.08530, 2019.

[136] A. Suhr, S. Zhou, A. Zhang, I. Zhang, H. Bai, and Y. Artzi. A corpus for reasoning about natural language grounded in photographs. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 6418–6428, 2019.

[137] S. Sukhbaatar, A. Szlam, J. Weston, and R. Fergus. End-to-end memory networks. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, pages 2440–2448, 2015.

[138] C. Sun, A. Myers, C. Vondrick, K. Murphy, and C. Schmid. Videobert: A joint model for video and language representation learning. In *2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019*, pages 7463–7472. IEEE, 2019.

[139] I. Sutskever, O. Vinyals, and Q. V. Le. Sequence to sequence learning with neural networks. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27: Annual Conference on Neural In-*

formation Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada, pages 3104–3112, 2014.

[140] H. Tan and M. Bansal. LXMERT: learning cross-modality encoder representations from transformers. In K. Inui, J. Jiang, V. Ng, and X. Wan, editors, *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pages 5099–5110. Association for Computational Linguistics, 2019.

[141] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, 4-9 December 2017, Long Beach, CA, USA*, pages 5998–6008, 2017.

[142] A. Vempala and D. Preotiuc-Pietro. Categorizing and inferring the relationship between the text and image of twitter posts. In A. Korhonen, D. R. Traum, and L. Màrquez, editors, *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 2830–2840. Association for Computational Linguistics, 2019.

[143] A. Wang, A. Singh, J. Michael, F. Hill, O. Levy, and S. R. Bowman. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *7th*

International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019. OpenReview.net, 2019.

[144] M. Wang, B. Zhao, and Y. Huang. PTR: phrase-based topical ranking for automatic keyphrase extraction in scientific publications. In *Neural Information Processing - 23rd International Conference, ICONIP 2016, Kyoto, Japan, October 16-21, 2016, Proceedings, Part IV*, pages 120–128, 2016.

[145] R. Wang, Z. Wei, P. Li, H. Shan, J. Zhang, Q. Zhang, and X. Huang. Keep it consistent: Topic-aware storytelling from an image stream via iterative multi-agent communication. *CoRR*, abs/1911.04192, 2019.

[146] X. Wang, F. Wei, X. Liu, M. Zhou, and M. Zhang. Topic sentiment analysis in twitter: a graph-based hashtag sentiment classification approach. In *Conference on Information and Knowledge Management*, 2011.

[147] Y. Wang, S. R. Joty, M. R. Lyu, I. King, C. Xiong, and S. C. H. Hoi. VD-BERT: A unified vision and dialog transformer with BERT. In *Proceedings of Empirical Methods in Natural Language Processing*, 2020.

[148] Y. Wang, J. Li, H. P. Chan, I. King, M. R. Lyu, and S. Shi. Topic-aware neural keyphrase generation for social media language. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 2516–2526, 2019.

[149] Y. Wang, J. Li, I. King, and M. R. Lyu. Cross-media keyphrase prediction: A unified framework with multi-modality multi-head attention and image wordings. In *Proceedings of Empirical Methods in Natural Language Processing*, 2020.

[150] Y. Wang, J. Li, I. King, M. R. Lyu, and S. Shi. Microblog hashtag generation via encoding conversation contexts. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 1624–1633, 2019.

[151] J. Weng and B.-S. Lee. Event detection in twitter. In *Proceedings of AAAI conference on weblogs and social media*, 2011.

[152] J. Weng, E. Lim, J. Jiang, and Q. He. Twitterrank: finding topic-sensitive influential twitterers. In B. D. Davison, T. Suel, N. Craswell, and B. Liu, editors, *Proceedings of the Third International Conference on Web Search and Web Data Mining, WSDM 2010, New York, NY, USA, February 4-6, 2010*, pages 261–270. ACM, 2010.

[153] J. Weston, S. Chopra, and K. Adams. #tagspace: Semantic embeddings from hashtags. In *Association for Computational Linguistics*, 2014.

[154] J. Weston, S. Chopra, and A. Bordes. Memory networks. In Y. Bengio and Y. LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San*

Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings, 2015.

[155] T. Whang, D. Lee, C. Lee, K. Yang, D. Oh, and H. Lim. Domain adaptive training BERT for response selection. *CoRR*, abs/1908.04812, 2019.

[156] I. H. Witten, G. W. Paynter, E. Frank, C. Gutwin, and C. G. Nevill-Manning. KEA: practical automatic keyphrase extraction. In *Proceedings of the Fourth ACM conference on Digital Libraries, August 11-14, 1999, Berkeley, CA, USA*, pages 254–255, 1999.

[157] H. Wu, H. Wang, and Z. Liu. Boosting statistical word alignment using labeled and unlabeled data. In *ACL 2006, 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference, Sydney, Australia, 17-21 July 2006*, 2006.

[158] Q. Wu, C. Shen, L. Liu, A. R. Dick, and A. van den Hengel. What value do explicit high level concepts have in vision to language problems? In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 203–212, 2016.

[159] Q. Wu, P. Wang, C. Shen, I. D. Reid, and A. van den Hengel. Are you talking to me? reasoned visual dialog generation through adversarial learning. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pages 6106–6115, 2018.

[160] Y. Wu, M. Schuster, Z. Chen, Q. V. Le, M. Norouzi, W. Macherey, M. Krikun, Y. Cao, Q. Gao, K. Macherey, J. Klingner, A. Shah, M. Johnson, X. Liu, L. Kaiser, S. Gouws, Y. Kato, T. Kudo, H. Kazawa, K. Stevens, G. Kurian, N. Patil, W. Wang, C. Young, J. Smith, J. Riesa, A. Rudnick, O. Vinyals, G. Corrado, M. Hughes, and J. Dean. Google's neural machine translation system: Bridging the gap between human and machine translation. *CoRR*, abs/1609.08144, 2016.

[161] F. Xia, T. Liu, J. Wang, W. Zhang, and H. Li. Listwise approach to learning to rank: theory and algorithm. In W. W. Cohen, A. McCallum, and S. T. Roweis, editors, *Machine Learning, Proceedings of the Twenty-Fifth International Conference (ICML 2008), Helsinki, Finland, June 5-9, 2008*, volume 307 of *ACM International Conference Proceeding Series*, pages 1192–1199. ACM, 2008.

[162] N. Xie, F. Lai, D. Doran, and A. Kadav. Visual entailment: A novel task for fine-grained image understanding. *CoRR*, abs/1901.06706, 2019.

[163] H. Xu and K. Saenko. Ask, attend and answer: Exploring question-guided spatial attention for visual question answering. In B. Leibe, J. Matas, N. Sebe, and M. Welling, editors, *Computer Vision - ECCV 2016 - 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part VII*, volume 9911 of *Lecture Notes in Computer Science*, pages 451–466. Springer, 2016.

[164] Z. Xu, B. Liu, B. Wang, C. Sun, X. Wang, Z. Wang, and C. Qi. Neural response generation via GAN with an approximate embedding layer. In M. Palmer, R. Hwa, and S. Riedel, editors, *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017, Copenhagen, Denmark, September 9-11, 2017*, pages 617–626. Association for Computational Linguistics, 2017.

[165] X. Yan, J. Guo, Y. Lan, and X. Cheng. A biterm topic model for short texts. In D. Schwabe, V. A. F. Almeida, H. Glaser, R. Baeza-Yates, and S. B. Moon, editors, *22nd International World Wide Web Conference, WWW '13, Rio de Janeiro, Brazil, May 13-17, 2013*, pages 1445–1456. International World Wide Web Conferences Steering Committee / ACM, 2013.

[166] X. Yan, J. Guo, Y. Lan, and X. Cheng. A biterm topic model for short texts. In *World Wide Web Conference*, 2013.

[167] T. Yang, Z. Zha, and H. Zhang. Making history matter: History-advantage sequence training for visual dialog. In *2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019*, pages 2561–2569. IEEE, 2019.

[168] Z. Yang, X. He, J. Gao, L. Deng, and A. J. Smola. Stacked attention networks for image question answering. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 21–29, 2016.

[169] H. Ye and L. Wang. Semi-supervised learning for neural keyphrase generation. In *Empirical Methods in Natural Language Processing*, 2018.

[170] P. Young, A. Lai, M. Hodosh, and J. Hockenmaier. From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions. *TACL*, 2:67–78, 2014.

[171] X. Yuan, T. Wang, R. Meng, K. Thaker, D. He, and A. Trischler. Generating diverse numbers of diverse keyphrases. *CoRR*, abs/1810.05241, 2018.

[172] R. Zellers, Y. Bisk, A. Farhadi, and Y. Choi. From recognition to cognition: Visual commonsense reasoning. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pages 6720–6731. Computer Vision Foundation / IEEE, 2019.

[173] J. Zeng, J. Li, Y. He, C. Gao, M. R. Lyu, and I. King. What you say and how you say it: Joint modeling of topics and discourse in microblog conversations. *Transactions of Association for Computational Linguistics*, 2019.

[174] J. Zeng, J. Li, Y. Song, C. Gao, M. R. Lyu, and I. King. Topic memory networks for short text classification. In E. Riloff, D. Chiang, J. Hockenmaier, and J. Tsujii, editors, *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 3120–3131. Association for Computational Linguistics, 2018.

[175] Q. Zhang, J. Wang, H. Huang, X. Huang, and Y. Gong. Hashtag recommendation for multimodal microblog using co-attention network. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI 2017, Melbourne, Australia, August 19-25, 2017*, pages 3420–3426, 2017.

[176] Q. Zhang, Y. Wang, Y. Gong, and X. Huang. Keyphrase extraction using deep recurrent neural networks on twitter. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, Austin, Texas, USA, November 1-4, 2016*, pages 836–845, 2016.

[177] R. Zhang, W. Li, D. Gao, and O. You. Automatic twitter topic summarization with speech acts. *IEEE Trans. Audio, Speech & Language Processing*, 2013.

[178] S. Zhang, Y. Yao, F. Xu, H. Tong, X. Yan, and J. Lu. Hashtag recommendation for photo sharing services. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019*, pages 5805–5812, 2019.

[179] Y. Zhang, J. Li, Y. Song, and C. Zhang. Encoding conversation context for neural keyphrase extraction from microblog posts. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for*

Computational Linguistics: Human Language Technologies, NAACL-HLT 2018, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 1 (Long Papers), pages 1676–1686, 2018.

[180] Z. Zheng, W. Wang, S. Qi, and S. Zhu. Reasoning visual dialogs with structural and partial observations. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pages 6669–6678, 2019.

[181] L. Zhou, H. Palangi, L. Zhang, H. Hu, J. J. Corso, and J. Gao. Unified vision-language pre-training for image captioning and VQA. *CoRR*, abs/1909.11059, 2019.