HICL: Hashtag-Driven In-Context Learning for Social Media Natural Language Understanding

Hanzhuo Tan, Chunpu Xu, Jing Li, Yuqun Zhang, *Member, IEEE*, Zeyang Fang, Zeyu Chen, and Baohua Lai

***Abstract*— Natural language understanding (NLU) is integral to various social media applications. However, the existing NLU models rely heavily on context for semantic learning, resulting in compromised performance when faced with short and noisy social media content. To address this issue, we leverage in-context learning (ICL), wherein language models learn to make inferences by conditioning on a handful of demonstrations to enrich the context and propose a novel hashtag-driven ICL (HICL) framework. Concretely, we pretrain a model #Encoder, which employs #hashtags (user-annotated topic labels) to drive BERT-based pretraining through contrastive learning. Our objective here is to enable #Encoder to gain the ability to incorporate topic-related semantic information, which allows it to retrieve topic-related posts to enrich contexts and enhance social media NLU with noisy contexts. To further integrate the retrieved context with the source text, we employ a gradient-based method to identify trigger terms useful in fusing information from both sources. For empirical studies, we collected 45 M tweets to set up an in-context NLU benchmark, and the experimental results on seven downstream tasks show that HICL substantially advances the previous state-of-the-art results. Furthermore, we conducted an extensive analysis and found that the following hold: 1) combining source input with a top-retrieved post from #Encoder is more effective than using semantically similar posts and 2) trigger words can largely benefit in merging context from the source and retrieved posts.**

***Index Terms*— In-context learning (ICL), natural language processing, pretrained language model, social media.**

I. Introduction

SOCIAL media provide rich resources of real-life and real-time content to understand our world and society. It motivates the demands and advances of various natural language processing (NLP) applications on these platforms, such as stance detection [1] and content recommendation [2]. For these applications, natural language understanding (NLU) plays an essential role in featuring the text and representing its semantics, where pretrained language models [3], [4], [5] contribute cutting-edge advances and serve as the backbone to broadly benefit downstream applications [6].

Pretrained language models gain generic NLU capabilities by navigating large-scale text and exploring the context, such as word co-occurrence patterns. Their performance, thus, heavily relies on *rich and high-quality* context, whereas that on social media is prevalently short and noisy. It results in a severe problem of *data sparsity*, meaning the context on social media exhibits an extremely sparse distribution of language features [7]. It would universally and negatively affect NLU pretraining and its downstream tasks [8], [9]. Viewing this challenge, BERTweet and Bernice are pretrained by randomly concatenating social media posts to lengthen context and alleviate sparsity [10], [11]. It is, however, suboptimal as random concatenations are unlikely to form coherent contexts.

Given these concerns, we envision that effective methods to automate *context enriching* will allow less sparse features and promisingly advance the generic NLU on social media. Our idea is inspired by recent advances in *in-context learning* (ICL) [12], [13], [14], uplifting model performance via conditioning on a few example data from training samples. However, the existing ICL studies are predominantly done on unidirectional models, such as GPT [15], and have primarily overlooked bidirectional models, such as the BERT family, despite the unique advantages of the latter on NLU tasks [3].

Motivated by the above points, our study aims to effectively retrieve external data and properly fine-tune bidirectional models to advance generic NLU on social media (henceforth, *in-context social media NLU*). To that end, we first pretrain an embedding model to help any social media post in context enriching by retrieving another relevant post; then, we insert trigger terms to fuse the enriched context for language models to refer to in semantics learning under sparsity. This way, the framework can easily be plugged into various task-specific fine-tuning frameworks as external features and broadly benefits downstream social media tasks.

In the existing approaches, ICL examples are usually constructed by retrieving the samples using metrics, such as semantic similarity [16] and mutual information [17]. However, their effectiveness is concerning due to social media's short and informally written text. A related study about

Manuscript received 16 August 2023; revised 16 January 2024; accepted 1 April 2024. Date of publication 15 April 2024; date of current version 7 April 2025. This work was supported in part by the Research Grants Council of the Hong Kong Special Administrative Region, China, under Project PolyU/25200821; in part by the Innovation and Technology Fund under Project PRP/047/22FX; in part by the National Natural Science Foundation of China Young Scientists Fund under Grant 62006203; in part by the National Natural Science Foundation of China under Grant 62372220; and in part by the China Computer Federation-Baidu Open Research Fund under Grant 2021PP15002000. *(Corresponding author: Jing Li.)*

Hanzhuo Tan is with the Department of Computing, Hong Kong Polytechnic University, Hong Kong, and also with the Department of Computer Science and Engineering, Southern University of Science and Technology, Shenzhen 518055, China (e-mail: hanzhuo.tan@connect.polyu.hk).

Chunpu Xu and Jing Li are with the Department of Computing, Hong Kong Polytechnic University, Hong Kong (e-mail: chun-pu.xu@connect.polyu.hk; jing-amelia.li@polyu.edu.hk).

Yuqun Zhang is with the Department of Computer Science and Engineering, Southern University of Science and Technology, Shenzhen 518055, China (e-mail: zhangyq@sustech.edu.cn).

Zeyang Fang and Zeyu Chen are with Baidu Inc., Beijing 100080, China (e-mail: fangzeyang@baidu.com; chenzeyu01@baidu.com).

Baohua Lai was with Baidu Inc., Beijing 100080, China. He is now with TOPFLYtech, Shenzhen, China (e-mail: laibaohua@baidu.com).

Digital Object Identifier 10.1109/TNNLS.2024.3384987

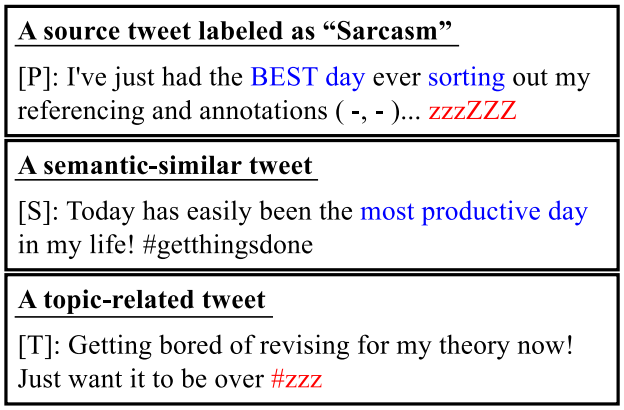


Fig. 1. Sample tweet P sarcasming overwork through "*zzzZZZ*" (top). It is followed by a SimCSE-retrieved tweet S with similar semantics (middle) and a "*#zzz*"-hashtagged tweet T cross refer other topic-related tweets (bottom). Blue words show semantic indicators, and red words show topical hints.

social media image–text understanding [18] showed retrieving context-enriching data was helpful; yet, image features contribute substantially more than text in retrieval training. It sheds light on the nontrivial challenge of learning what to retrieve given data sparsity in a text-only context, which is much more common in social media posts than those with images. To address this issue, we pretrain the retrieval model via utilizing *hashtags*, user-annotated topic labels starting with a "#" and cross referring to other topic-related posts [19]. It associates posts about the same topic, learns semantics in a richer topic-coherent context, and gains topic relevance for retrieval. Hashtags have been adopted in many task-specific scenarios (e.g., image captioning [20] and sentiment analysis [21]). In contrast, we present a novel initiative to explore its effects in ICL for a broad range of NLU tasks.

To better illustrate the potential of hashtags in context enriching, Fig. 1 shows a sample tweet P that conveys a sense of sarcasm through an emoticon "*zzzZZZ*" (indicating overwork and opposing the previous sayings). As can be seen, P's short context and implicit writings may hinder NLU models from capturing the genuine underlying meanings. We then retrieved a tweet S using a popular semantic-based retrieval model SimCSE [22], which exhibits similar semantics (heavy work), partially helps enrich context, and yet ignores the sarcastic hint from "*zzzZZZ*." Meanwhile, a related hashtag "*#zzz*" might gather other topic-related (such as T in Fig. 1) complaining about overwork, strengthen the NLU in "*zzzZZZ*," and offer more direct assistance to infer sarcasm.

Here, a straightforward approach is to feed the encoder with the concatenation of the source post with another topic-related post. Nevertheless, this method may also distract the model, causing it to pay undue attention to nonessential details instead of focusing on the main message of the source post. Therefore, we employ a gradient-based approach to identify trigger terms that facilitate the incorporation of the retrieved text's context. To the best of our knowledge, *hashtag-driven ICL* (HICL) is the first framework leveraging hashtags in large-scale pretraining for social media NLU, which enables the pretrained model to retrieve topic-related posts and enhances the ICL framework by incorporating automatically generated trigger terms for context enrichment.

Concretely, HICL works in a pretraining and fine-tuning paradigm. In pretraining, we develop *#Encoder*, a hashtag-driven pretrained model based on RoBERTa [4]. It is pretrained on 179 M hashtagged tweets via contrastive learning to pull the tweets with the same hashtags closer together in embedding space and push apart those with different hashtags. Then, in the fine-tuning, #Encoder helps retrieve topic-related data, which is later utilized for context enriching and merging with the help of trigger terms during the training of specific downstream tasks. Here, we set up HICL with a *#Database* containing 45 M tweets grouped by hashtags for #Encoder to retrieve context-enriching tweets.

To evaluate HICL's performance, we conducted experiments on seven popular Twitter benchmark datasets. The main results demonstrate that HICL enables bidirectional language models, such as BART [23], RoBERTa [4], and BERTweet [10], to achieve superior performance by incorporating the top-retrieved tweet from #Encoder. Furthermore, inserting trigger terms between the source and retrieved tweets can enhance the overall performance, indicating that these trigger terms can positively facilitate information integration between the two components.

In further discussion about HICL, we first quantify the number of trigger terms and show that even a single trigger term can positively impact downstream tasks. Then, by probing into the position of trigger terms, we find that those at the beginning or middle of sentences effectively facilitate information integration; in contrast, those at the end are less useful. Next, we quantify the scale of retrieved context and observe that augmenting with more context is beneficial to enhance social media NLU. However, the marginal benefits of adding additional text to the input diminish with the increasing number of retrieved pieces of information. Finally, case studies and analysis of the trigger terms provide insight into how HICL helps NLU.

In summary, our contributions are threefold.

We propose a novel HICL framework for generic social media NLU in data sparsity, which can retrieve topic-related posts and enrich contexts via gradient-searched trigger terms.

We develop the first hashtag-driven pretrained model, #Encoder, leveraging hashtags to learn interpost topic relevance (for retrieval) via contrastive learning over 179 M tweets.

We contribute a large corpus with 45 M tweets for retrieval, and the experiments on seven Twitter benchmarks show that HICL advances the overall results of various trendy NLU models.[1]

II. Related Work

Our HICL is built upon ICL and retrieves posts based on sentence embeddings and hashtags. In the following, we first discuss previous ICL work, followed by the discussion on sentence embedding and hashtag modeling.

[1]The HICL framework and benchmark with 45 M tweets are available at https://github.com/albertan017/HICL.

A. In-Context Learning

In the initial ICL work, researchers enhance the GPT3 model's zero-shot inference potential by concatenating numerous exemplar instances ahead of the input text [15]. It offers an interpretable interface for interacting with large language models (LLMs), making it easier to integrate human knowledge by modifying the templates and demonstrations. With the rapid scaling of LLMs size, the enormous computational expense of fine-tuning LLMs accentuates the necessity for ICL. To select suitable demonstration examples, researchers employ various metrics to retrieve samples, e.g., SentenceBert embedding similarity [16], mutual information [17], supervised retriever EPR [24], and so on. There is also an inductive class learning experiment that showcases how demonstration samples drive end-task performance [14], which indicates that demonstration samples provide the following: 1) instances from the label space demonstrating the range of possible labels; 2) examples of the distribution of the input text, illustrating the kinds of inputs the model will encounter; and 3) demonstrations of the overall format of the sequence, exhibiting the structure that the model's predictions should follow. These factors comprise the key reasons demonstration samples facilitated ICL model performance.

Although ICL has shown encouraging outcomes, previous work has predominantly concentrated on unidirectional models for natural language generation (NLG), such as GPT3 or LLaMa, leaving bidirectional models (such as the BERT family) largely unexplored. Meanwhile, bidirectional models have shown unique advantages in NLU [3]. It is because the bidirectional attention mechanisms can incorporate context from both directions when encoding a word or sentence, allowing effectiveness in capturing linguistic phenomena, such as long-distance dependencies, pronoun resolution, and negation understanding. It also reflects how human readers process language as we understand words and sentences beyond relying solely on left-to-right contexts, since it cannot fully capture the dependencies between the context words [25]. We, thus, study tailor-making ICL to fine-tune bidirectional models and thoroughly evaluate its capabilities in social media NLU.

B. Sentence Embedding

Sentence embedding is the process of mapping sentences into continuous vector representations. It captures sentences' semantic meaning and allows them to be compared mathematically using distance metrics. This vector representation enables various downstream NLP applications, such as sentence classification, semantic similarity, and sentiment analysis, and is a widely applied index in information retrieval. Early works build sentence embeddings via averaging word vectors, e.g., word2vec [26], which are word-level vector representations pretrained from word co-occurrences. Doc2Vec [27] extends the idea of word embeddings to the document level and generated document embeddings by using either distributed memory mode or distributed bag of words mode, where the former pretrains embeddings by predicting words from their context and the latter does the opposite. Despite its simplicity, Doc2Vec has been shown to produce helpful sentence representations.

Inspired by Siamese network, researchers later leverage contrastive learning to obtain sentence embeddings. InferSent [28] uses natural language inference (NLI) datasets to train a Siamese bi-LSTM to predict the relations of input sentence pairs. As the model is trained to distinguish between entailment, contradiction, and neutral relationships between sentence pairs, it forces the model to learn meaningful sentence representations. The idea of encoding sentences with the NLI dataset is further extended into transformer architecture in Universal Sentence Encoder [29]. Also, the corresponding results indicate that sentence embeddings are significantly helpful for transfer learning and can be used to obtain promising task performance with significantly less task-specific training data. More recently, scholars have incorporated the concept of contrastive learning into the pretraining paradigm. SentenceBert [30] is among the initial models to modify the pretrained BERT model [3] by utilizing a Siamese architecture to encode the semantic meaning of sentences into embeddings. SimCSE [22] presents an unsupervised method that utilizes standard dropout as noise and predicts an input sentence itself in a contrastive objective. They further include supervised contrastive learning with NLI datasets and reach state-of-the-art performance on semantic textual similarity (STS) tasks. Although the dominant techniques for generating sentence embeddings are trained on formal written text, such as the Stanford NLI dataset (SNLI) [31] and Multi-Genre NLI dataset (MNLI) [32], social media language—which is often characterized by sparsity and noise—has received relatively little attention. As a result, researchers have largely overlooked the informal writing style of social media language and instead adopted language encoders that are specifically designed for formal written text [33], [34], which may compromise the final results.

To the best of our knowledge, there are very few pretrained models for sentence embedding that are specifically tailored for social media language. While some attempts have been made to pretrain language models on social media data, such as BERTweet [10], Bernice [11], and TwHIN-BERT [35], most of them have been limited to using randomly grouped tweets, which would result in a lack of coherent context and may consequently lead to confusion in pretraining. In contrast, our #Encoder exhibits the first pretrained sentence embedding model specifically tailored for social media language in a context-rich manner. Rather than prioritizing the semantic content of social media posts, usually characterized as noisy and lacking in context, #Encoder adopts a topic-based perspective and utilizes hashtags as a means of grouping posts and driving contrastive pretraining for encoding social media posts. Built upon the #Encoder-learned embeddings, we further explore HICL, a novel framework on their use for downstream tasks under an ICL approach.

C. Hashtag Modeling

Our work is also related to prior studies using hashtags for language learning on social media platforms. Although social

media language lacks context within individual posts, it offers a vast quantity of data. Hashtags, which are user-generated topic labels, are widely available on social media platforms and serve as clusters of post topics. These hashtags are typically used as indicators for constructing language resources [1], [36] and for social media tasks [37], [38], [39]. For instance, hashtag semantics have been incorporated and benefit content recommendation [40]. Moreover, a recent study shows that adding automatically generated hashtags can enrich the context of tweets and help low-resource classification [41]. However, directly supplementing hashtags to tweets is arguably suboptimal, as it may also bring noise and mislead the model, because the appended hashtags and tweets may not be featured in the same semantic space for classification. In contrast, to allow models to attend salient parts, we propose generating trigger terms to serve as a bridge for improving the integration between retrieved content and source input. Furthermore, they restricted their scope to low-resource classification with limited labeled data, whereas here, we focus on a more general scenario of social media NLU.

In addition, some researchers work on hashtag embedding to help models gain hashtag-level understanding. In this line, Hashtag2Vec [42] learns hashtag representations by jointly modeling their co-occurrence patterns and associated textual content; SHE [43] captures semantic and sentiment information in hashtag embeddings leveraging multitask learning. Nevertheless, no prior work has exploited hashtags in gathering topic-related posts for large-scale language pretraining, which is a research gap we aim to address in this article.

Meanwhile, to understand the topics of posts without hashtags, researchers have proposed various algorithms for recommending the hashtags. Early approaches utilize machine learning techniques, such as topic modeling [44] and convolutional neural networks [45], to analyze post semantics for hashtag recommendations. Considering the rich information in social networks, other methods incorporate user data, such as communities [46] and hub nodes [47] to improve recommendations. Recently, reference [48] constructs separate latent spaces for embedding post text and associated hashtags. A multilayer perceptron mapping process then learns a translation from text semantic features to hashtag latent representations for recommendation. Inspired by the hashtag embedding, we propose leveraging hashtags as topic indicators and employing contrastive learning to pretrain an #Encoder model that can encode topic information implicitly into the high-dimensional representation space. Our objective diverges from the field of hashtag recommendation, which focuses on classifying posts for subsequent application. In contrast, our #Encoder model is tailored to directly retrieve posts related to certain topics to provide a rich context for language models to advance generic NLU on social media.

III. HICL Framework

This section introduces how we pretrain #Encoder and apply it in the HICL framework. The framework design is first overviewed in Section III-A. Then, we discuss the pretraining process for #Encoder in Section III-B and how it is further leveraged in HICL to fine-tune language models in Section III-C. Finally, we present the details to search for the trigger terms in Section III-D.

A. Framework Design Overview

As discussed above, HICL employs #Encoder for retrieving posts to enrich post-level context in task-specific fine-tuning. For this reason, we feed #Encoder with hashtag-grouped posts (posts with the same hashtag), which differs from the BERTweet, Bernice, or TwHIN-BERT schemes that take randomly concatenated tweets as input. Our intuition is that posts about the same topic (hinted by hashtags) would allow richer context for pretrained models to learn semantics. The grouping design considers that the limited words in a post may prevent the model's language learning potential from being fully exploited in pretraining.

To better interpret this point, we first review the general design of most pretrained models for NLU [3], [4]. It adopts a transformer encoder [49] fed by a word sequence $\mathbf{x} = \langle x_1, x_2, \ldots, x_L \rangle$ (L is the word number). For each word $x_i \in \mathbf{x}$ and its word embedding e_i, the model explores its representation h_i through multiple self-attention encoder layers based on x_i's occurrences with all words in $\mathbf{x}$. A self-attention layer is formulated as follows:

$$h_i = \sum_{j=1}^{L} \text{softmax}\left(\frac{Q_i K_j}{d_k}\right) V_i. \tag{1}$$

Q, K, and V are the projections of $\mathbf{x}$'s input embeddings. d_k is the scaling factor to avoid a small gradient.

In pretraining, the transformer encoders leverage self-supervised learning tasks, e.g., masked language model (MLM), to explore the word features in context for learning general NLU skills. However, because of the sparsely distributed features, NLU encoders may need help to practice these tasks given post-level context only. To mitigate sparsity, #Encoder is pretrained on grouped input with contrastive learning for a richer context in semantic learning. Consequently, HICL matches a post with a retrieved post to follow this context-rich design and enable easier fine-tuning [50].

We then present the details of contrastive learning. Formally, given a batch of post pairs $D = \{[\mathbf{x}_1, \mathbf{x}_1^+], \ldots, [\mathbf{x}_n, \mathbf{x}_n^+]\}$ ($\mathbf{x}_i$ and $\mathbf{x}_i^+$ are tagged the same hashtag in our scenario), #Encoder encodes D into latent semantic space, $H = \{[h_1, h_1^+], \ldots, [h_n, h_n^+]\}$ as their representations.

In the hashtag-driven pretraining, #Encoder aims to pull representations of posts with the same hashtag, $[h_i, h_i^+]$, closer and push apart those with different hashtags, $[h_i, h_j^+]$ ($i \neq j$). Here, we follow SimCSE [22] and compute the cross-entropy objective with in-batch negatives. Also, the training loss for a batch D is defined as follows:

$$\text{loss} = -\log \frac{e^{\text{sim}(h_i, h_i^+)/\tau}}{\sum_j^N e^{\text{sim}\left(h_i, h_j^+\right)/\tau}} \tag{2}$$

where $\text{sim}(h_i, h_i^+)$ is the cosine similarity between post embeddings h_i and h_i^+, and τ is a temperature hyperparameter.

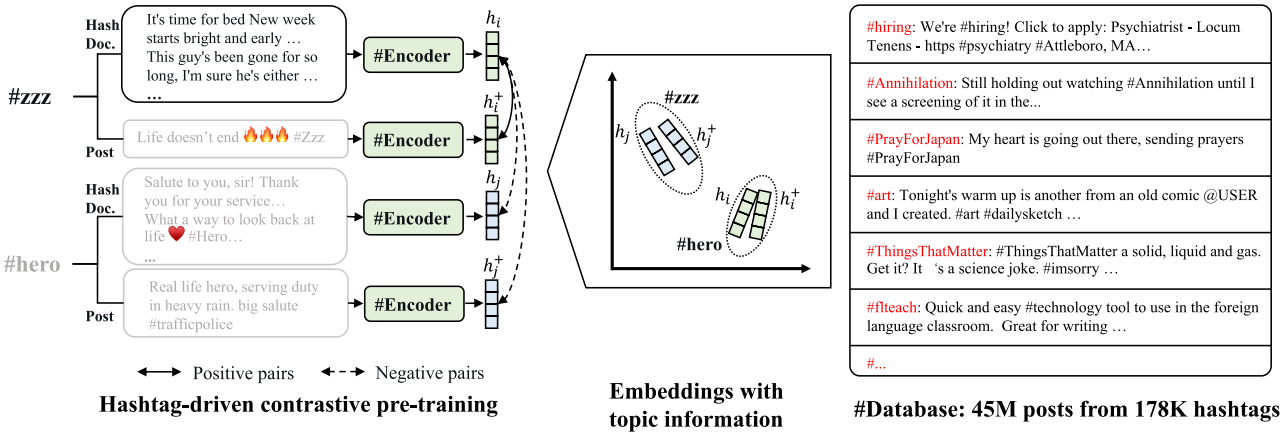


Fig. 2. Workflow to pretrain #Encoder on 179 M Twitter posts, each containing a hashtag. #Encoder is pretrained on pairwise posts, and contrastive learning which guides it to learn topic relevance by learning to identify posts with the same hashtag. The pretraining aims to pull representations of posts with the same hashtag, $[h_i, h_i^+]$, closer, and push apart those with different hashtags, $[h_i, h_j^+]$ $(i \neq j)$. Consequently, this results in embeddings that are infused with topic-specific information. We randomly noise the hashtags to avoid trivial representation.

B. #Encoder Pretraining

We then discuss how to pretrain #Encoder, and the workflow is shown in Fig. 2. It is built on the architecture of RoBERTa with a 12-layer transformer encoder [49]. We employ contrastive learning to pretrain large-scale tweets. In the following, we first discuss how to gather the pretraining data, followed by the training methods.

1) Pretraining Data: #Encoder is pretrained on 15 GB of plain text from 179 M tweets and 4 billion tokens. Following the practices to pretrain BERTweet [10], the raw data were collected from the archived Twitter stream containing 4 TB of sampled tweets from January 2013 to June 2021.[2] For data preprocessing, we ran the following steps. First, we employed FastText [51] to extract English tweets and only kept tweets with hashtags. Then, low-frequency hashtags appearing in less than 100 tweets were further filtered out to reduce sparsity. After that, we obtained a large-scale dataset containing 179 M tweets; each has at least one hashtag and, hence, corresponds to 180k hashtags in total.

To further examine how to utilize hashtags, we show the log-scaled distribution of hashtag frequency in Fig. 3. As can be seen, it is extremely imbalanced and roughly exhibits a long tail, where each hashtag appears in an average of 951.4 tweets. We observe that the majority (86%) of hashtags contain less than 1000 tweets, while several (the generic ones) appear in millions of tweets, e.g., #job occurs in 1.6 M tweets, #nowplaying 1.3 M, and #hiring 0.9 M.

To enable a more balanced training, we sampled the posts with respect to the inverse of hashtag frequency and randomly formed pairs of tweets sharing a hashtag for contrastive learning. Besides, in order to guide #Encoder to focus on nontrivial representation learning, we randomly add noise to hashtags, such as deletion and segmentation [52]. It is because hashtags are characterized by the # symbol and the nonindent format, which may mislead the model to encode trivial and useless features for tackling pretraining tasks.

[2]https://archive.org/details/twitterstream

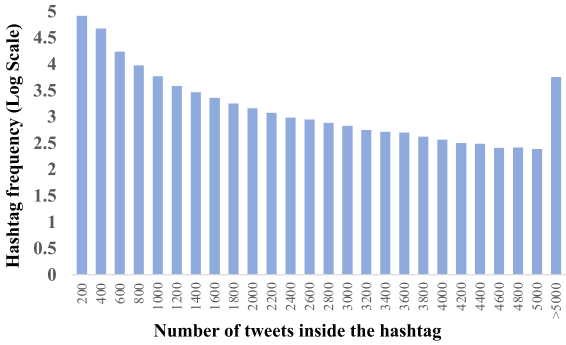


Fig. 3. This histogram shows the distribution of hashtag frequencies based on the number of tweets containing each hashtag. The x-axis bins hashtags by the number of tweets they appear in. The y-axis shows the frequency—how many hashtags occur in each bin's tweet count range, on a logarithmic scale to enhance the visibility of the data. The distribution demonstrates a long-tail effect, with most hashtags tweeted infrequently (left bins) and few achieving widespread use (right bins).

2) Pretraining Methods: To leverage hashtag-gathered context in pretraining, we exploit contrastive learning and train #Encoder to identify pairwise posts sharing the same hashtag for gaining topic relevance, as illustrated in (2).

To effectively encode the topic information, half of the input is constructed by concatenating posts with the same hashtag to the max sequence length, resulting in a single long document. The other half is present in individual posts. This way, #Encoder can explore topic information in a context-rich setting while considering the limited length of social media posts. Furthermore, we engage MLM with loss coefficient α as an auxiliary pretraining task to the aforementioned hashtag-driven contrastive learning. It is to retain the word representation capability in #Encoder pretraining.

For evaluation of sentence encoding models on downstream tasks, we refer to SimCSE [22] and find that its results are inferior when directly fine-tuned for classification. Likewise, #Encoder is pretrained on paired posts to learn topic relevance, which may better gain text-matching capability than

classification. We, hence, apply #Encoder to retrieve posts in HICL fine-tuning, which will be discussed below.

C. HICL Fine-Tuning

In fine-tuning, most NLU downstream tasks are formulated as a classification problem, which is to maximize posterior probability $P(y|\mathbf{x})$, meaning the most likely class y given a post $\mathbf{x}$. Due to data sparsity, the limited features in $\mathbf{x}$ may challenge NLU models to explicitly connect $\mathbf{x}$ to y. We, hence, introduce a latent variable $\mathbf{x}'$ (from unlimited post space on social media) to mitigate their information gap, and the theoretical formulation is as follows:

$$P(y|\mathbf{x}) = \sum_{\mathbf{x}'} P\left(y \mid \mathbf{x}, \mathbf{x}'\right) P\left(\mathbf{x}' \mid \mathbf{x}\right). \tag{3}$$

In practice, rather than sampling from an unlimited collection of posts, we use the #Encoder to retrieve the top-K most relevant posts $\mathbf{x}'$ given an input post $\mathbf{x}$, denoted as $P(\mathbf{x}'|\mathbf{x})$. Since #Encoder is designed to capture topical information, such retrieval provides posts on the same topic as $\mathbf{x}$ (indicated by hashtags). This allows richer topical context for models to connect the sparse information in post $\mathbf{x}$ with supplementary topical knowledge in post $\mathbf{x}'$ to make the classification prediction y, formulated as $P(y|\mathbf{x}, \mathbf{x}')$. We design the following processes to run HICL fine-tuning and show the workflow in Fig. 4.

First, the #Encoder is pretrained in an unsupervised manner using hashtag-driven contrastive learning (detailed in Section III-B) to encode the latent topic semantics. This allows mapping input $\mathbf{x}$ to a high-dimensional embedding h_x, which implicitly represents its latent topic z. Then, for each input $\mathbf{x}$ in the fine-tuning dataset, the pretrained #Encoder retrieves the most topic-related post $\mathbf{x}'$ from a large external corpus based on the similarity between embeddings h_x and $h_{x'}$. In this way, the retrieved $\mathbf{x}'$ acts as supplementary contextual information reflecting the latent topic indicated by the #Encoder's continuous representation for input $\mathbf{x}$. This allows providing an enriched contextual representation for task-specific NLU fine-tuning and inference, without requiring predefined discrete topics. Finally, the retrieved $\mathbf{x}'$ connects $\mathbf{x}$ and y through the concatenation of the $\mathbf{x}$ with originally limited context and expanded $\mathbf{x}'$ containing relevant latent topic information. This enriched representation mitigates data sparsity challenges and enhances NLU modeling with flexible topic encoding.

For the setup of a retrieval dataset, we consider the observations in Fig. 3, where most hashtags have a 100-scale frequency while very few million scale. To enable a reasonable search space for efficient and balanced retrieval, we randomly sampled at most 500 tweet samples from each hashtag group, resulting in 45 M unique tweets from 178 657 hashtags. The dataset then bases the #Encoder retrieval in the HICL framework (thereby, #Database).

We acknowledge the potential scalability concerns raised by a direct retrieval approach, as it requires comparing every post against a potentially vast database of encoded representations. However, at our current database scale of 45 M tweets, direct embedding retrieval is still highly efficient. As a reference, previous study [53] has shown that direct embedding similarity search over 1B vectors takes only around 0.2 s.[3] Our strategy benefits from cutting-edge algorithms tailored for rapid and voluminous retrieval tasks, ensuring that scalability remains manageable. Furthermore, the direct retrieval method bypasses the need for explicit topic classification, eliminating the risks associated with misclassification and the propagation of errors through the retrieval process. By directly comparing embeddings, we ensure that the most topical similar posts are retrieved, reflecting a comprehensive understanding of latent topics inherent in social media content. Hence, for the sake of clarity, we concentrate on the direct retrieval method in our research.

D. Trigger Terms Search Algorithm

Here, we further discuss how to fuse the retrieved and source context in fine-tuning. Although the retrieved posts are intended to provide supportive background, directly appending the two posts may be ineffective, because the retrieved posts may not share the classification labels with the source and potentially confuse the model. Accordingly, we propose inserting trigger terms optimized to combine the information from the retrieved text and source input, resulting in a coherent representation conducive to classification. Inspired by previous work [54], we employ continuous vectors as trigger terms rather than utilizing natural language trigger terms. Concretely, given post $\mathbf{x}$, retrieved post $\mathbf{x}'$, and series of trigger terms $T_1, T_2, \ldots, T_n$, we reformulate the input in the following form:

$$[T_1, \ldots, T_l], \mathbf{x}, \left[T_{(l+1)}, \ldots, T_m\right], \mathbf{x}', \left[T_{(m+1)}, \ldots, T_n\right]. \tag{4}$$

For a reformulated input $\mathbf{x}, \mathbf{x}', T$, the model's training loss is calculated as follows:

$$\underset{\theta, T}{\operatorname{argmin}}\ \mathcal{L} = -\sum \log P_\theta\left(y|\mathbf{x}, \mathbf{x}', T\right). \tag{5}$$

The algorithm is summarized in Algorithm 1. To seek effective trigger terms, we first initialize trigger terms with random continuous embeddings (Lines 2) and train the embeddings of the set of trigger terms, T, alongside other input tokens to establish a strong initialization (Lines 3–6). After that, we freeze the other model parameters (Lines 7) and solely fine-tune the embeddings of these trigger terms for optimal solutions (Lines 8–11). Note that tuning the trigger terms is computational efficient as we freeze the model parameters and only update the trigger embeddings.

We also present an ablation study on this separate training process to evaluate its contributions (see Section V).

IV. Experimental Setup

We set up the evaluation of HICL on the Twitter data, where we test our fine-tuned results on seven popular tasks to examine generic capability in social media NLU. In the experimental discussion, a Twitter post is thereby referred to as a *tweet*.

[3]https://github.com/facebookresearch/faiss/tree/main/benchs

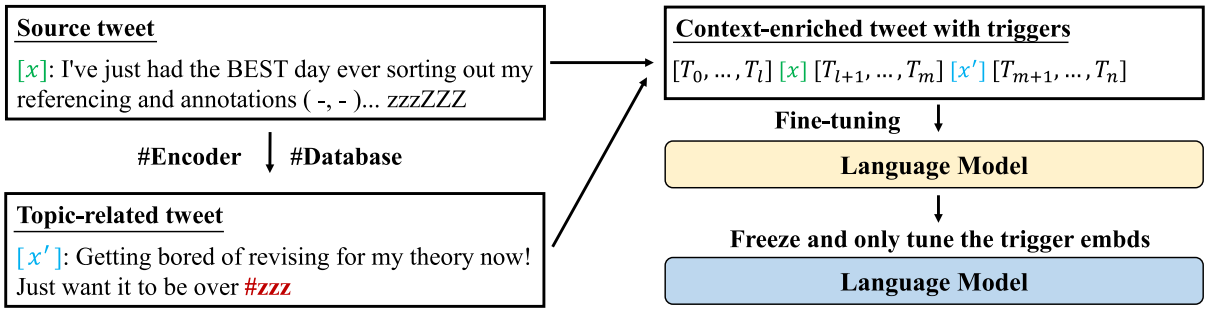


Fig. 4. Workflow of HICL fine-tuning. A tweet $\mathbf{x}$ is first encoded with #Encoder, and the output embedding h_i is then used to search the #Database to retrieve the most topic-related tweet $\mathbf{x}'$, which has an embedding h_r with the smallest distance to h_i. After that, $\mathbf{x}'$ and $\mathbf{x}$ are paired in concatenation and inserted with trigger terms for task-specific fine-tuning. Here, HICL can both work for tweets with and without hashtags.

Algorithm 1 Trigger Terms Search Algorithm

Input: Model θ, trigger term T, Dataset D
Output: Optimized Model θ and trigger term embd. T_{embd}.

Function `TriggerTermSearch`(θ, T, D)**:**
  Randomly Initialize trigger term embd. T_{embd}
  for $\mathbf{x}$ *in* $\mathcal{D}$ **do**
    $\mathcal{L} = -\log P_\theta(y|\mathbf{x}, \mathbf{x}', T)$ ▷ Compute loss in Eq. 5
    $g \leftarrow \nabla\mathcal{L}(\mathbf{x}, \mathbf{x}', T)$ ▷ Compute gradient for all the input
    update$(\theta, T_{embd}) \leftarrow g$ ▷ update all the parameters
  Freeze θ, only update trigger embd. T_{embd}
  for $\mathbf{x}$ *in* $\mathcal{D}$ **do**
    $\mathcal{L} = -\log P_\theta(y|\mathbf{x}, \mathbf{x}', T)$
    $g \leftarrow \nabla\mathcal{L}(T)$ ▷ Compute gradient only for trigger terms
    update$(T_{embd}) \leftarrow g$ ▷ Only update trigger embd.
  return θ, T_{embd}

A. #Encoder Pretraining Settings

The hashtag-driven contrastive learning is implemented with PyTorch[4] and hugging face transformers library.[5] For post representation, we use the embedding from the last hidden layer of #Encoder for the "< s >" token [22], [30]. This "< s >" token is inserted at the start of the sentence to signify the beginning [4], [10], [23]. For hyperparameters, we primarily follow BERTweet configurations and initialize the #Encoder parameters with BERTweet checkpoint for continued pretraining based on four NVIDIA RTX3090 GPUs. The pretraining is conducted by Adam optimizer with a peak learning rate set to $1e-5$, maximum sequence length to 128, and batch size to 512. We set temperature $\tau = 0.05$ in contrastive learning [as shown in (2)] and MLM loss coefficient $\alpha = 0.1$. #Encoder is pretrained for ten epochs, roughly taking 7 days.

[4]https://pytorch.org/
[5]https://github.com/huggingface/transformers

B. Benchmark Datasets

The evaluation presented in this article is based on seven widely used SemEval Twitter benchmark datasets, each related to a different popular NLU task. In the following, we briefly introduce each benchmark, and the corresponding statistics are presented in Table I.

Stance detection focuses on understanding the author's stance and is formulated as follows. Given a tweet, the model aims to predict whether the author has a favorable, neutral, or unfavorable position toward a proposition or target. Here, we employ SemEval-2016 task 6 on Detecting Stance in Tweets, which provides five target domains: abortion, atheism, climate change, feminism, and Hillary Clinton. In this study, we merge the target domains and predict the stance.

Emotion recognition is to recognize the author's emotion evoked by a tweet. We use SemEval-2018 task 1 dataset following TweetEval's practice, where the model should distinguish four emotions: anger, joy, sadness, and optimism.

Irony detection focuses on recognizing whether a tweet includes ironic intents or not, making it a binary classification task. Here, we use the data from SemEval-2018 task 3.

Offensive language identification aims to allow models to predict whether or not some offensive language is present in an input tweet, whose data are from SemEval-2019 task 6.

Hate speech detection is to predict whether a tweet is hateful against any of two target communities: immigrants and women. Our dataset comes from SemEval-2019 task 5.

Humor detection is to enable automatic detection of whether a given tweet exhibits a sense of humor, and the data are from SemEval-2021 task 7.

Sarcasm detection is a binary classification task of predicting whether a tweet shows a sense of sarcasm. The benchmark is set up based on SemEval-2022 task 6, which the tweet authors themselves labeled.

C. Evaluation Metrics

The evaluation metrics we use for each dataset are the same as those employed in the original paper that introduced the dataset. For **emotion**, **offensive**, **hate**, **humor**, and **sarcasm** benchmarks, macroaveraged F1 over all classes is employed. For **stance** benchmark, macroaveraged of F1 of "favor" and

TABLE I
BENCHMARK DATASET STATISTICS

Dataset	Train	Val	Test
Stance	2,620	294	1,249
Emotion	3,257	374	1,421
Irony	2,862	955	784
Offensive	11,916	1,324	860
Hate	9,000	1,000	2,970
Humor	8,000	1,000	1,000
Sarcasm	3,114	353	1,400

"against" classes is used. As for **irony** benchmark, we adopt F1 of ironic class as evaluation metric.

Overall, these seven tweet classification benchmarks reflect a wide range of NLU capabilities to tackle social media data and comprehensively assess our proposed HICL framework's effectiveness in understanding such data.

D. Comparison Setup

We thoroughly experiment with the proposed HICL on the backbone of widely employed bidirectional language models: BART and RoBERTa, and the state-of-the-art model for tweet NLU, BERTweet. In the following, we provide a concise overview of each model.

Bidirectional and autoregressive transformer (BART) [23] is a pretrained language model that employs the vanilla transformer architecture. It can be viewed as a combination of the bidirectional encoder, similar to BERT, and an autoregressive decoder, akin to GPT, into a single Seq2Seq model. BART is trained via a two-step process involving the corruption of text using an arbitrary noising function and the subsequent learning of a model to reconstruct the original text.

Robustly optimized BERT pretraining approach (RoBERTa) [4] is an optimized BERT pretraining model through the use of larger data scales, longer training time, dynamic masking strategies, and optimized hyperparameters.

BERTweet [10] is the first large-scale pretrained model for the NLU of English tweets. It leverages an 80-GB corpus consisting of 850 M tweets. BERTweet adopts the RoBERTa architecture and training strategy and yet concatenates tweets to achieve the maximum sequence length. In addition, the model provides a specialized tokenizer for tweets.

Our experiments consider taking these three models as the baselines to fine-tune the original datasets (namely, Base). For comparable results, HICL fine-tuning (see Section III-C) is also carried out on varying base models, which takes paired input from a given tweet and its match retrieved by #Encoder. To allow the easy use of HICL, the pretrained #Encoder is directly applied for retrieval without task-specific fine-tuning. Here, we employ Faiss Library [53] to speed up retrieval and costs around 30 ms/45 M search on an Intel Xeon Gold 6248R CPU. We empirically insert five trigger terms between the given tweet and its matched retrieved text. Following this setup, we also examine HICL variants with pretrained retrieval counterparts, enriching a tweet's context with SimCSE (namely, SimCSE). We fine-tune BERTweet for 30 epochs for each task with a warm-up learning rate of $1e-5$ and a batch size of 16. We apply early stopping if no improvement is observed on validation for over five continuous epochs. All models run for ten times, and we report their average results in Section V.

In addition, we evaluate the effectiveness of conventional ICL, which involves conditioning the model's inferences on several demonstrations from training samples (namely, ICL). We follow the methods LMBFF proposed in [55] to implement this baseline. Concretely, we first sample a single example from each class for each input to create a demonstration set and then perform prompt-tuning to enable the model to learn from the demonstrations in the training set.

V. EXPERIMENTAL RESULTS

We first discuss the main comparison results and ablations in Section V-A. Then, a quantitative analysis is presented in Section V-B to examine how trigger terms and retrieved tweets perform in varying scenarios, followed by a case study in Section V-C to interpret how HICL benefits social media NLU.

A. Main Comparison Results and Ablation Study

The fine-tuned results on the seven Twitter benchmarks (Section IV) and the averages are shown in Table II.

Our experimentation results provide support for our assertion that topic-related information, as obtained through the #Encoder retrieved tweets, is more effective in enhancing generic NLU than semantic-related information (SimCSE-retrieved tweets) or demonstrations from similar training samples (+ICL). These results suggest that enriching a tweet's context with relevant tweets is a simple yet effective approach for improving generic NLU in data sparsity. As social media tweets face several sparsity problems, enriching the topic-related context becomes even more crucial in helping the language model understand the given scenario. On the other hand, concatenating a semantic-similar tweet to the input may not be as helpful. While a semantic-similar tweet may contain similar words or phrases to the input tweet, it may not necessarily provide additional context or information that can help the model better understand the topic being discussed.

Furthermore, ICL is generally effective in improving downstream task performance. This is done by providing demonstration tweets that are derived from training samples. These demonstrations could guide the model in NLU training and have been shown to improve the model's overall performance on downstream tasks. However, the degree of improvement achieved by the basic ICL or SimCSE is limited. The possible reason is that demonstration tweets from training samples are already familiar to the model or have been incorporated into its training. Meanwhile, HICL shows larger performance gains, implying that the topic-related tweets found by #Encoder can better help the model comprehend the topic at hand and offer a relevant yet supplementary view.

To further investigate the relative contributions of varying HICL modules, we present the ablation studies in Table III, where "base" refers to the vanilla base models. For other ablations, we first examined the effectiveness of trigger terms,

TABLE II

COMPARISON RESULTS OF DIFFERENT MODELS WITH VARYING BIDIRECTIONAL BACKBONES. THE BEST RESULTS IN EACH COLUMN UNDER A BACKBONE ARE UNDERLINED. ON AVERAGE, OUR HICL FRAMEWORK SIGNIFICANTLY OUTPERFORMS OTHER COMPARISON MODELS WITH $p < 0.05$

Method	Stance	Emotion	Irony	Offensive	Hate	Humor	Sarcasm	Average
BART								
Base	67.3 ± 0.6	77.8 ± 0.5	67.3 ± 1.9	81.2 ± 0.6	49.4 ± 1.8	95.2 ± 0.3	34.6 ± 1.6	67.5 ± 1.1
+ICL	67.4 ± 1.8	77.4 ± 0.9	69.5 ± 1.1	80.8 ± 0.8	50.2 ± 1.4	94.4 ± 0.4	33.3 ± 1.7	67.6 ± 1.1
+SimCSE	66.3 ± 1.4	76.3 ± 0.4	66.4 ± 2.4	79.6 ± 1.1	51.0 ± 1.8	95.3 ± 0.3	32.7 ± 1.2	66.8 ± 1.2
+HICL	68.0 ± 1.0	78.6 ± 0.4	68.6 ± 0.8	80.9 ± 0.9	51.0 ± 1.1	94.7 ± 0.4	34.5 ± 2.5	68.1 ± 1.0
RoBERTa								
Base	69.0 ± 0.5	78.2 ± 0.5	64.3 ± 2.6	79.7 ± 0.9	47.9 ± 1.8	95.0 ± 0.6	38.0 ± 2.5	67.4 ± 1.4
+ICL	67.5 ± 1.4	77.8 ± 0.7	68.6 ± 1.8	79.5 ± 1.2	50.8 ± 1.3	94.2 ± 0.4	36.0 ± 1.9	67.8 ± 1.2
+SimCSE	68.0 ± 0.7	77.1 ± 1.0	68.8 ± 2.3	78.5 ± 1.0	48.6 ± 1.8	94.9 ± 0.3	36.8 ± 1.8	67.5 ± 1.3
+HICL	69.4 ± 1.3	78.4 ± 0.6	72.8 ± 1.8	79.9 ± 0.7	51.2 ± 1.4	94.7 ± 0.2	41.0 ± 2.1	69.6 ± 1.2
BERTweet								
Base	70.3 ± 0.9	81.2 ± 0.8	78.7 ± 1.4	80.5 ± 0.8	54.9 ± 0.9	95.9 ± 0.3	45.9 ± 2.7	72.5 ± 1.1
+ICL	69.8 ± 1.5	67.5 ± 0.9	80.3 ± 1.4	76.4 ± 1.3	58.6 ± 2.2	94.4 ± 0.6	43.3 ± 0.8	70.0 ± 1.3
+SimCSE	69.0 ± 0.8	80.5 ± 0.7	80.9 ± 1.5	80.1 ± 0.7	56.5 ± 1.6	96.2 ± 0.4	47.2 ± 1.9	72.9 ± 1.1
+HICL	69.5 ± 0.7	81.2 ± 0.6	81.5 ± 0.9	80.1 ± 0.6	56.1 ± 1.8	96.0 ± 0.3	49.0 ± 2.4	73.4 ± 1.1

TABLE III

AVERAGE RESULTS FOR DIFFERENT TRAINING METHODS

Model	Base	+HICL w/o Tri.	+HICL w/o Sep.	+HICL
BART	67.5	67.8	67.5	68.1
RoBERTa	67.4	68.9	68.8	69.6
BERTweet	72.5	73.0	73.4	73.4
Average	69.2	69.9	70.0	70.3

Note: "w/o" is abbreviation for without, "Tri." represents trigger terms, "Sep." indicates separate tuning as discussed in Algorithm 1 Lines 7-11. The use of trigger terms between the source tweet and retrieved tweets improves performance. Further optimization of the trigger embeddings leads to additional gains. This supports the hypothesis that trigger terms help integrate semantic information from the retrieved and source texts.

TABLE IV

AVERAGE RESULTS VARYING THE NUMBER OF TRIGGERS

Model	Base	w/o Tri.	T. #1	T. #3	T. #5	T. #7	T. #9
BART	67.5	67.8	67.9	68.0	68.0	68.2	68.0
RoBERTa	67.4	68.9	69.0	68.9	69.6	68.7	68.9
BERTweet	72.5	73.0	73.2	73.5	73.4	73.3	73.4
Average	69.2	69.9	70.0	70.1	70.3	70.1	70.1

Note: "T. #N" indicates N trigger terms are inserted between the retrieved and source tweet. Adding more trigger terms shows minimal impact on the average performance.

with "+HICL w/o Tri." denoting simply concatenating the retrieved tweet with the source input. Second, recall that in Section III-D, we described that during training, we simultaneously train the embeddings of trigger terms and other tokens for initialization, followed by further fine-tuning the trigger embeddings separately. An alternative approach would be to train the trigger embeddings jointly with the other token embeddings without additional separate tuning. We present comparative results to validate the importance of this separate tuning—"+HICL w/o Sep." indicates training without further separate tuning. "+HICL" denotes the full model with separate fine-tuning to optimize the trigger embeddings.

The averaged results on seven benchmarks are detailed in Table III. It demonstrates that inserting trigger terms between the source and retrieved tweets can enhance the final performance. Moreover, additional optimization of the trigger embeddings exhibits further downstream performance gains. These results support our hypothesis that trigger terms facilitate the merging of semantic information carried by the retrieved text and the source input. Thus, our study underscores the potential utility of trigger embeddings for generally improving the automatic NLU capability on social media language.

B. Quantitative Analysis

In the previous section, we have shown the benefits of leveraging #Encoder-retrieved tweets through our trigger term search algorithm. In the following, we quantify how the trigger term usage and retrieved tweets help social media NLU learning. The analyses of the sensitivity of trigger terms regarding their quantity and position are presented first. Then, we examine the impact of the number of retrieved tweets on performance of the downstream tasks.

1) Varying the Number of Trigger Terms: Here, we investigate how language models handle trigger terms with varying numbers and show the all-task average results in Table IV, where "T. #N" indicates that N trigger terms are inserted between the retrieved and source tweet. We observe that although trigger terms are helpful, adding more trigger terms shows minimal impact on the average performance. Notably, even a single trigger can positively affect the downstream task, reinforcing our argument that trigger terms are critical for facilitating the integration of information between the source and matched retrieved tweets.

TABLE V
AVERAGE RESULTS VARYING THE PLACING POSITIONS OF TRIGGERS

Model	Base	No Trigger	Front	Middle	End	ALL
BART	67.5	67.8	67.7	68.1	67.7	68.0
RoBERTa	67.4	68.9	69.1	69.6	68.6	69.0
BERTweet	72.5	73.0	73.7	73.4	73.0	73.4
Average	69.2	69.9	70.2	70.3	69.8	70.1

Note: "Front," "Middle," "End," and "All" denote the trigger placements, either either at the beginning, in the middle, at the end, or at all of the aforementioned positions. Trigger terms placed at the front or middle of tweets effectively facilitate information integration.

2) Varying the Position of Trigger Terms: In the previous experiments, the trigger term was empirically inserted between the source and retrieved tweets. We are then interested in how the varying placement positions affect in the NLU learning outcome. The overall average results across all the tasks are illustrated in Table V. We utilize the terminology "front," "middle," "end," and "all" to denote different trigger placements. Expressly, "front" signifies the insertion of trigger terms before all the tweets, "middle" refers to the placement of trigger terms between the source and retrieved tweets, and "end" represents the concatenation of trigger terms at the end. "All" denotes the including of trigger terms in all of the positions above. "No trigger" indicates that source and retrieved tweets are concatenated directly without triggers.

Table V shows that trigger terms placed at the front or middle of tweets effectively facilitate information integration. In contrast, trigger terms placed at the end are generally unhelpful. It is intuitive why trigger terms in the middle produce the best results—their position provides explicit cues for connecting the source and retrieving information, acting as a "bridge" between the two. Trigger terms at the front also help, as they prime the language model to make a connection. However, when placing the trigger terms at the end, the hint of such a "connection" may be weaker. A possible explanation for this phenomenon is that placing trigger terms at the end may interfere with the natural sentence structure and disrupt the model's understanding of the input. The models are trained on data where relevant information is usually close to each other, which can bias the models to favor attending more strongly to adjacent or near-adjacent parts of the input. Therefore, placing the trigger terms at the end could cause the model to focus on resolving the unexpected input structure rather than integrating the source and retrieving information.

3) Varying Number of Retrieved Tweets: We have analyzed the effect of trigger terms in fusing source and retrieved tweets. Then, we center on the retrieved tweets and examine how the number of retrieved tweets affects the performance. Fig. 5 shows the all-task average results, where "#Enc.+N" indicates top-N retrieved tweets are selected to concretize the context. We exclude the sarcasm dataset for averaging due to its different trends and will discuss it later.

The findings presented in Fig. 5 suggest that augmenting the model's input with more contextual information generally enhances its NLU capabilities. However, for several reasons below, the marginal benefits of adding more text to the input gradually diminish with a continuous increase in the retrieved tweet number for use. 1) *Redundancy:* Concatenating multiple texts that revolve around the same topic may lead to redundancy in the input. This redundancy could limit the marginal utility of including the richer context in the input, since the model may not obtain additional insights from repeatedly processing similar contents. 2) *Noise:* Adding more tweets to the input may introduce noise, as only part of the information may be task-relevant. This noise can hinder the model in identifying and concentrating on the most crucial information, thereby impeding performance gains. 3) *Model Capacity:* The capacity of a language model, which is determined by its architecture (e.g., number of layers, hidden units, and self-attention heads), may constrain its performance; even when more information is provided to the model by concatenating additional texts, the model may need the capacity to utilize this information to enhance its performance effectively.

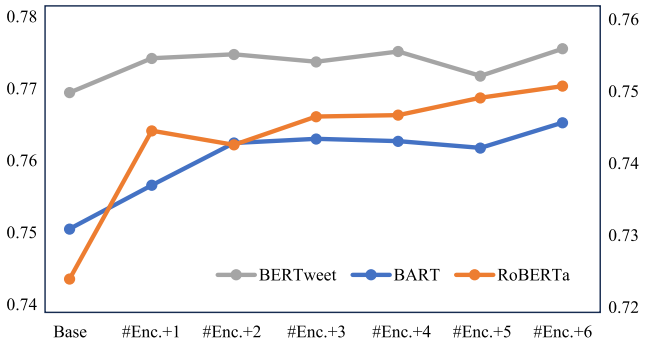


Fig. 5. Average results on varying the number of retrieved tweets, excluding the sarcasm dataset because of its different trends (to be discussed separately).

TABLE VI
SLOPE OF LINEAR LEAST SQUARES REGRESSION WITH THE NUMBER OF RETRIEVED TWEETS AS THE INDEPENDENT VARIABLE AND TASK PERFORMANCE AS THE DEPENDENT VARIABLE: COEFFICIENTS SCALED UP BY 1000 FOR CLARITY

Method	Sta.	Emo.	Iroy	Off.	Hate	Hum.	Sar.	Avg
BART	0.67	2.66	3.49	0.32	0.08	0.01	-6.79	0.04
RoBERTa	1.66	2.72	1.63	-1.47	3.26	0.92	-7.37	0.19
BERTweet	1.35	-0.02	-1.65	0.25	-0.03	-0.05	-2.02	-0.31

To probe into the impact of the retrieved tweet number on individual tasks, we analyzed the slope of linear least squares while varying the number of retrieved tweets concerning task performance. The results are presented in Table VI. Aside from the sarcasm task, BART and RoBERTa typically exhibit performance gains, as the number of concatenated tweets in the input increases for various tasks. In contrast, the BERTweet model does not enjoy such benefits due to its pretraining on randomly concatenated tweets, which lack coherence and hinder the model's ability to comprehend more extended context. It is consistent with Fig. 5, where BERTweet presents flattened trends using more than one retrieved tweet, whereas BART and RoBERTa show a more apparent increasing trend.

Notably, the sarcasm dataset negatively relates to a longer retrieved context with all backbones. It can be attributed to the significant class imbalance, as only 24% of the training data are labeled as sarcasm. This imbalance creates difficulties for

TABLE VII
THREE CASES FROM EMOTION, STANCE, AND SARCASM DATASETS. THE COLUMNS FROM LEFT TO RIGHT SHOW TASK, SOURCE TWEET (FOR RETRIEVAL), SEMANTIC-SIMILAR TWEET (RETRIEVED BY SIMCSE), AND TOPIC-RELATED TWEET (RETRIEVED BY #ENCODER)

Task	Source tweet	Semantic-similar tweet	Topic-related tweet
Emotion: Joy	@USER 3. home alone 4. fast and furious	home alone.. #FreakingOut	@USER The Amazing Spiderman #MovieTrivia
Stance: Favor	#Mission : #Climate @USER home > Run your dishwasher only if it's full. (by @USER #Tip #ActOnClimate #SemST	Only do laundry when you have a full load. The same holds true with your dishwasher. Only run when full. #moneysavingtips #energysavers	Learn how we're making homes and #buildings more energy efficient than ever. HTTPURL #ActOnClimate HTTPURL
Sarcasm	I love a Monday morning so glad the weekends over!	Loves a Monday morning #whaaaaat	How is it half 8 already?? #hatemondays #weekendplease

the model in making accurate predictions, particularly under noisy conditions when concatenating more retrieved tweets.

C. Qualitative Analysis

We have quantitatively shown how HICL benefits from using trigger terms and retrieved tweets. Below, we qualitatively analyze some outputs of HICL to provide more insight into how it manages NLU learning on social media.

1) Trigger Terms: We analyzed the Euclidean distance between the embeddings of trigger terms and all other tokens in the model vocabulary. Our findings indicate that trigger terms exhibit relatively smaller Euclidean distance and, thus, closer embedding similarity to the [mask], [pad], and [unk] tokens with respect to all other tokens in the vocabulary.

These special tokens, [mask], [pad], and [unk], have diffuse and indistinct semantic properties, as they function primarily as placeholders rather than conveyors of specific semantic content. Analogously, we posit that trigger terms improving model performance are likely to have similarly indistinct and diffuse semantic representations, as they act as placeholders or "signal" tokens, conveying information about the structural or intentional properties of the input rather than embedding precise semantic content. The semantic indeterminacy of these trigger terms may allow for a more flexible interpretation of the surrounding context, and their use as placeholding signals would further provide the model with useful structural information to improve downstream predictions. These results suggest why trigger terms are helpful in HICL design through a qualitative lens.

2) Retrieved Tweets: For the usefulness of #Encoder-retrieved tweets, we present some cases in Table VII to illustrate how it benefits social media NLU. For instance, the "home alone" in the first-row tweet is a movie's name, which may mislead the emotion detection model in predicting a negative emotion. #Encoder can connect it with other movie tweets through hashtag "#MovieTrivia" to help NLU models cognize movie names to avoid errors in task tackling. Without such capability, SimCSE retrieves a tweet with similar words and offered limited help to make sense of movie names.

By qualitatively analyzing many cases, we find SimCSE tends to find tweets with similar words and sometimes cannot provide much extra information. On the contrary, #Encoder can retrieve topic-related tweets, which may offer a complementary perspective to gain topic-level knowledge for better NLU.

VI. CONCLUSION

We have proposed an HICL framework with a pretrained #Encoder based on hashtags to retrieve topic-related social media posts, which are combined with the source input for context enriching via gradient-optimized trigger terms for task-specific fine-tuning. #Encoder is pretrained on 179 M hashtagged tweets using contrastive learning, enabling it to associate tweets with matching hashtags and differentiate those with divergent topics. We implemented HICL with a *#Database* of 45 M hashtag-grouped tweets, allowing #Encoder to acquire and integrate context with triggers in task-specific fine-tuning.

We conducted experiments on seven widely used Twitter benchmark datasets to evaluate #Encoder and HICL's effectiveness. Our results indicate that HICL significantly enhances the performance of bidirectional language models, such as BART, RoBERTa, and BERTweet, by incorporating the top-retrieved tweets from #Encoder. In addition, we found that incorporating trigger terms between the source and retrieved tweets can improve overall performance, suggesting that trigger terms facilitate effective information integration.

Through a quantitative analysis of trigger terms, we have demonstrated that even a single trigger can positively influence downstream tasks. Further investigation revealed that trigger terms at the beginning or middle of sentences contribute to effective information integration, whereas those positioned at the end of sentences are generally less beneficial. Moreover, supplementing the model with additional context improves language comprehension abilities, although the marginal benefits decrease as more information is retrieved.

Despite the promising results of the HICL framework, it presents several limitations requiring future research.

First, our pretraining corpus relies on abundant user-annotated hashtags, which lack quality assurance. In addition, hashtag frequency exhibits a long-tail distribution, leading to class imbalance challenges. Investigating automatic methods to create a high-quality pretraining corpus could be valuable.

Second, our retrieval method utilizes a large #Database with 45 M tweets and requires 30 ms for retrieval on an Intel Xeon Gold 6248R CPU. Corpus distillation techniques, such

as clustering and indexing, could improve retrieval efficiency while maintaining acceptable performance levels.

Third, the HICL framework and #Encoder do not enforce semantic consistency during retrieval. Although our experiments have validated the effectiveness of the proposed framework, extra efforts in selecting the optimal context through reranking algorithms can allow more performance gain and provide a better solution to the data-sparsity challenge.

REFERENCES

[1] K. Glandt, S. Khanal, Y. Li, D. Caragea, and C. Caragea, "Stance detection in COVID-19 tweets," in *Proc. 59th Annu. Meeting Assoc. Comput. Linguistics, 11th Int. Joint Conf. Natural Lang. Process. (ACL/IJCNLP)*. Stroudsburg, PA, USA: Association for Computational Linguistics, Aug. 2021, pp. 1596–1611. [Online]. Available: https://aclanthology.org/2021.acl-long.127

[2] X. Zeng, J. Li, L. Wang, Z. Mao, and K.-F. Wong, "Dynamic online conversation recommendation," in *Proc. 58th Annu. Meeting Assoc. Comput. Linguistics*, Jul. 2020, pp. 3331–3341. [Online]. Available: https://aclanthology.org/2020.acl-main.305

[3] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proc. Conf. North Amer. Chapter Association Comput. Linguistics, Human Language Technol.* Minneapolis, MN, USA: Association for Computational Linguistics, Jun. 2019, pp. 4171–4186. [Online]. Available: https://aclanthology.org/N19-1423

[4] Y. Liu et al., "RoBERTa: A robustly optimized BERT pretraining approach," 2019, *arXiv:1907.11692*.

[5] C. Raffel et al., "Exploring the limits of transfer learning with a unified text-to-text transformer," *J. Mach. Learn. Res.*, vol. 21, no. 1, pp. 1–67, Jan. 2020.

[6] F. Barbieri, J. Camacho-Collados, L. E. Anke, and L. Neves, "TweetEval: Unified benchmark and comparative evaluation for tweet classification," in *Proc. Findings Assoc. Comput. Linguistics*, Nov. 2020, pp. 1644–1650. [Online]. Available: https://aclanthology.org/2020.findings-emnlp.148

[7] J. Zeng, J. Li, Y. Song, C. Gao, M. R. Lyu, and I. King, "Topic memory networks for short text classification," in *Proc. Conf. Empirical Methods Natural Lang. Process.* Brussels, Belgium: Association for Computational Linguistics, Oct. 2018, pp. 3120–3131. [Online]. Available: https://aclanthology.org/D18-1351

[8] J. Chen, Y. Hu, J. Liu, Y. Xiao, and H. Jiang, "Deep short text classification with knowledge powered attention," in *Proc. AAAI Conf. Artif. Intell.*, Jul. 2019, vol. 33, no. 1, pp. 6252–6259, doi: 10.1609/aaai.v33i01.33016252.

[9] B. Lyu, L. Chen, S. Zhu, and K. Yu, "LET: Linguistic knowledge enhanced graph transformer for Chinese short text matching," in *Proc. AAAI Conf. Artif. Intell.*, 2021, vol. 35, no. 15, pp. 13498–13506.

[10] D. Q. Nguyen, T. Vu, and A. T. Nguyen, "BERTweet: A pre-trained language model for English tweets," in *Proc. Conf. Empirical Methods Natural Lang. Processing: Syst. Demonstrations*, 2020, pp. 9–14. [Online]. Available: https://aclanthology.org/2020.emnlp-demos.2

[11] A. DeLucia, S. Wu, A. Mueller, C. Aguirre, P. Resnik, and M. Dredze, "Bernice: A multilingual pre-trained encoder for Twitter," in *Proc. Conf. Empirical Methods Natural Lang. Process.* Abu Dhabi, United Arab Emirates: Association for Computational Linguistics, 2022, pp. 6191–6205.

[12] S. Min, M. Lewis, L. Zettlemoyer, and H. Hajishirzi, "MetaICL: Learning to learn in context," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Human Lang. Technol.*, 2022, pp. 2791–2809. [Online]. Available: https://aclanthology.org/2022.naacl-main.201

[13] S. Min, M. Lewis, H. Hajishirzi, and L. Zettlemoyer, "Noisy channel language model prompting for few-shot text classification," in *Proc. 60th Annu. Meeting Assoc. Comput. Linguistics*, S. Muresan, P. Nakov, and A. Villavicencio, Eds. Dublin, Ireland: Association for Computational Linguistics, 2022, pp. 5316–5330. [Online]. Available: https://aclanthology.org/2022.acl-long.365

[14] S. Min et al., "Rethinking the role of demonstrations: What makes in-context learning work?" in *Proc. Conf. Empirical Methods Natural Lang. Process.* Abu Dhabi, United Arab Emirates: Association for Computational Linguistics, 2022, pp. 11048–11064. [Online]. Available: https://aclanthology.org/2022.emnlp-main.759

[15] T. B. Brown et al., "Language models are few-shot learners," in *Proc. NIPS*, vol. 33, H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, Eds. Curran Associates, 2020, pp. 1877–1901. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf

[16] J. Liu, D. Shen, Y. Zhang, B. Dolan, L. Carin, and W. Chen, "What makes good in-context examples for GPT-3?" in *Proc. 3rd Workshop Knowl. Extraction Integr. Deep Learn. Architectures*. Dublin, Ireland: Association for Computational Linguistics, May 2022, pp. 100–114. [Online]. Available: https://aclanthology.org/2022.deelio-1.10

[17] T. Sorensen et al., "An information-theoretic approach to prompt engineering without ground truth labels," in *Proc. 60th Annu. Meeting Assoc. Comput. Linguistics*. Dublin, Ireland: Association for Computational Linguistics, May 2022, pp. 819–862. [Online]. Available: https://aclanthology.org/2022.acl-long.60

[18] C. Xu and J. Li, "Borrowing human senses: Comment-aware self-training for social media multimodal classification," in *Proc. Conf. Empirical Methods Natural Lang. Process.* Abu Dhabi, United Arab Emirates: Association for Computational Linguistics, Dec. 2022, pp. 5644–5656. [Online]. Available: https://aclanthology.org/2022.emnlp-main.381

[19] Y. Zhang, Y. Zhang, C. Xu, J. Li, Z. Jiang, and B. Peng, "#HowYouTagTweets: Learning user hashtagging preferences via personalized topic attention," in *Proc. Conf. Empirical Methods Natural Lang. Process.* Punta Cana, Dominican Republic: Association for Computational Linguistics, Nov. 2021, pp. 7811–7820. [Online]. Available: https://aclanthology.org/2021.emnlp-main.616

[20] D. Lu, S. Whitehead, L. Huang, H. Ji, and S.-F. Chang, "Entity-aware image caption generation," in *Proc. Conf. Empirical Methods Natural Lang. Process.* Brussels, Belgium: Association for Computational Linguistics, 2018, pp. 4013–4023. [Online]. Available: https://aclanthology.org/D18-1435

[21] L. Gyanendro Singh, A. Mitra, and S. Ranbir Singh, "Sentiment analysis of tweets using heterogeneous multi-layer network representation and embedding," in *Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP)*, 2020, pp. 8932–8946. [Online]. Available: https://aclanthology.org/2020.emnlp-main.718

[22] T. Gao, X. Yao, and D. Chen, "SimCSE: Simple contrastive learning of sentence embeddings," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2021, pp. 6894–6910. [Online]. Available: https://aclanthology.org/2021.emnlp-main.552

[23] M. Lewis et al., "BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension," in *Proc. 58th Annu. Meeting Assoc. Comput. Linguistics*, D. Jurafsky, J. Chai, N. Schluter, and J. Tetreault, Eds. Jul. 2020, pp. 7871–7880. [Online]. Available: https://aclanthology.org/2020.acl-main.703

[24] O. Rubin, J. Herzig, and J. Berant, "Learning to retrieve prompts for in-context learning," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Human Lang. Technol.*, Jul. 2022, pp. 2655–2671. [Online]. Available: https://aclanthology.org/2022.naacl-main.191

[25] Z. Du et al., "GLM: General language model pretraining with autoregressive blank infilling," in *Proc. 60th Annu. Meeting Assoc. Comput. Linguistics*. Dublin, Ireland: Association for Computational Linguistics, May 2022, pp. 320–335. [Online]. Available: https://aclanthology.org/2022.acl-long.26

[26] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Adv. Neural Inf. Process. Syst.*, vol. 26, C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Weinberger, Eds. Lake Tahoe, NV, USA: Curran Associates, 2013, pp. 1–9. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2013/file/9aa42b31882ec039965f3c4923ce901b-Paper.pdf

[27] Q. Le and T. Mikolov, "Distributed representations of sentences and documents," in *Proc. 31st Int. Conf. Mach. Learn.*, vol. 32, no. 2, E. P. Xing and T. Jebara, Eds. Beijing, China, Jun. 2014, pp. 1188–1196. [Online]. Available: https://proceedings.mlr.press/v32/le14.html

[28] A. Conneau, D. Kiela, H. Schwenk, L. Barrault, and A. Bordes, "Supervised learning of universal sentence representations from natural language inference data," in *Proc. Conf. Empirical Methods Natural Lang. Process.* Copenhagen, Denmark: Association for Computational Linguistics, Sep. 2017, pp. 670–680. [Online]. Available: https://aclanthology.org/D17-1070

[29] D. Cer et al., "Universal sentence encoder," 2018, *arXiv:1803.11175*.

[30] N. Reimers and I. Gurevych, "Sentence-BERT: Sentence embeddings using Siamese BERT-networks," in *Proc. Conf. Empirical Methods Natural Lang. Process. 9th Int. Joint Conf. Natural Lang. Process. (EMNLP-IJCNLP)*. Hong Kong: Association for Computational Linguistics, Nov. 2019, pp. 3982–3992. [Online]. Available: https://aclanthology.org/D19-1410

[31] S. R. Bowman, G. Angeli, C. Potts, and C. D. Manning, "A large annotated corpus for learning natural language inference," in *Proc. Conf. Empirical Methods Natural Lang. Process.* Lisbon, Portugal: Association for Computational Linguistics, 2015, pp. 632–642. [Online]. Available: https://aclanthology.org/D15-1075

[32] A. Williams, N. Nangia, and S. Bowman, "A broad-coverage challenge corpus for sentence understanding through inference," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Human Lang. Technol.* New Orleans, LA, USA: Association for Computational Linguistics, 2018, pp. 1112–1122. [Online]. Available: https://aclanthology.org/N18-1101

[33] N. Wenzlitschke and P. Sulzle, "Using BERT to retrieve relevant and argumentative sentence pairs," in *Proc. Conf. Labs Evaluation Forum*, vol. 3180, G. Faggioli, N. Ferro, A. Hanbury, and M. Potthast, Eds. Bologna, Italy, 2022, pp. 3149–3163. [Online]. Available: https://ceur-ws.org/Vol-3180/paper-264.pdf

[34] N. Tahaei, H. Verma, P. Bagherzadeh, F. Farahnak, N. Sheikh, and S. Bergler, "Identifying author profiles containing irony or spreading stereotypes with SBERT and emojis," in *Proc. Conf. Labs Eval. Forum*, vol. 3180, G. Faggioli, N. Ferro, A. Hanbury, and M. Potthast, Eds. Bologna, Italy. 2022, pp. 2675–2681. [Online]. Available: https://ceur-ws.org/Vol-3180/paper-222.pdf

[35] X. Zhang et al., "TWHIN-BERT: A socially-enriched pre-trained language model for multilingual tweet representations at Twitter," in *Proc. 29th ACM SIGKDD Conf. Knowl. Discovery Data Mining*. New York, NY, USA: Association for Computing Machinery, 2023, pp. 5597–5607. [Online]. Available: https://doi.org/10.1145/3580305.3599921

[36] C. Van Hee, E. Lefever, and V. Hoste, "SemEval-2018 task 3: Irony detection in English tweets," in *Proc. 12th Int. Workshop Semantic Eval.* New Orleans, LA, USA: Association for Computational Linguistics, 2018, pp. 39–50. [Online]. Available: https://aclanthology.org/S18-1005

[37] W. Wang et al., "Topic-guided variational auto-encoder for text generation," in *Proc. Conf. North Amer. Chapter Association Computational Linguistics, Human Language Technol.* Minneapolis, MN, USA: Association for Computational Linguistics, Jun. 2019, pp. 166–177. [Online]. Available: https://aclanthology.org/N19-1015

[38] K. Ding, J. Li, and Y. Zhang, "Hashtags, emotions, and comments: A large-scale dataset to understand fine-grained social emotions to online topics," in *Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP)*, Nov. 2020, pp. 1376–1382. [Online]. Available: https://aclanthology.org/2020.emnlp-main.106

[39] V. Gupta and R. Hewett, "Real-time tweet analytics using hybrid hashtags on Twitter big data streams," *Information*, vol. 11, no. 7, p. 341, Jun. 2020.

[40] J. Weston, S. Chopra, and K. Adams, "#TagSpace: Semantic embeddings from hashtags," in *Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP)*. Doha, Qatar: Association for Computational Linguistics, Oct. 2014, pp. 1822–1827. [Online]. Available: https://aclanthology.org/D14-1194

[41] S. Diao, S. S. Keh, L. Pan, Z. Tian, Y. Song, and T. Zhang, "Hashtag-guided low-resource tweet classification," in *Proc. ACM Web Conf.* New York, NY, USA: Association for Computing Machinery, 2023, pp. 1415–1426, doi: 10.1145/3543507.3583194.

[42] J. Liu, Z. He, and Y. Huang, "Hashtag2Vec: Learning hashtag representation with relational hierarchical embedding model," in *Proc. 27th Int. Joint Conf. Artif. Intell.*, Jul. 2018, pp. 3456–3462, doi: 10.24963/ijcai.2018/480.

[43] L. G. Singh, A. Anil, and S. R. Singh, "SHE: Sentiment hashtag embedding through multitask learning," *IEEE Trans. Computat. Social Syst.*, vol. 7, no. 2, pp. 417–424, Apr. 2020.

[44] F. Godin, V. Slavkovikj, W. De Neve, B. Schrauwen, and R. Van de Walle, "Using topic models for Twitter hashtag recommendation," in *Proc. 22nd Int. Conf. World Wide Web*. New York, NY, USA: Association for Computing Machinery, May 2013, pp. 593–596, doi: 10.1145/2487788.2488002.

[45] Y. Gong and Q. Zhang, "Hashtag recommendation using attention-based convolutional neural network," in *Proc. 24th Int. Joint Conf. Artif. Intell.*, 2016, pp. 2782–2788.

[46] Y. Djenouri, A. Belhadi, G. Srivastava, and J. C. Lin, "Toward a cognitive-inspired hashtag recommendation for Twitter data analysis," *IEEE Trans. Computat. Social Syst.*, vol. 9, no. 6, pp. 1748–1757, Dec. 2022.

[47] A. Javari, Z. He, Z. Huang, R. Jeetu, and K. C.-C. Chang, "Weakly supervised attention for hashtag recommendation using graph data," in *Proc. Web Conf.* New York, NY, USA: Association for Computing Machinery, 2020, pp. 1038–1048, doi: 10.1145/3366423.3380182.

[48] R. Cantini, F. Marozzo, G. Bruno, and P. Trunfio, "Learning sentence-to-hashtags semantic mapping for hashtag recommendation on microblogs," *ACM Trans. Knowl. Discovery Data*, vol. 16, no. 2, pp. 1–26, Sep. 2021, doi: 10.1145/3466876.

[49] A. Vaswani et al., "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, I. Guyon et al. Eds. Long Beach, CA, USA: Curran Associates, 2017, pp. 1–11. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf

[50] S. Gururangan et al., "Don't stop pretraining: Adapt language models to domains and tasks," in *Proc. 58th Annu. Meeting Assoc. Comput. Linguistics*, 2020, pp. 8342–8360. [Online]. Available: https://aclanthology.org/2020.acl-main.740

[51] A. Joulin, E. Grave, P. Bojanowski, and T. Mikolov, "Bag of tricks for efficient text classification," in *Proc. 15th Conf. Eur. Chapter Assoc. Comput. Linguistics*. Valencia, Spain: Association for Computational Linguistics, Apr. 2017, pp. 427–431. [Online]. Available: https://aclanthology.org/E17-2068

[52] R. C. Rodrigues et al., "Zero-shot hashtag segmentation for multilingual sentiment analysis," 2021, *arXiv:2112.03213*.

[53] J. Johnson, M. Douze, and H. Jégou, "Billion-scale similarity search with GPUs," *IEEE Trans. Big Data*, vol. 7, no. 3, pp. 535–547, Jul. 2021.

[54] Z. Zhong, D. Friedman, and D. Chen, "Factual probing is [MASK]: Learning vs. learning to recall," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Human Lang. Technol.*, 2021, pp. 5017–5033. [Online]. Available: https://aclanthology.org/2021.naacl-main.398

[55] T. Gao, A. Fisch, and D. Chen, "Making pre-trained language models better few-shot learners," in *Proc. 59th Annu. Meeting Assoc. Comput. Linguistics, 11th Int. Joint Conf. Natural Lang. Process.*, Aug. 2021, pp. 3816–3830. [Online]. Available: https://aclanthology.org/2021.acl-long.295

Hanzhuo Tan received the B.E. degree in electrical engineering and automation from Shandong University, Jinan, China, in 2015, and the M.Sc. degree in electrical engineering from the University of Southern California, Los Angeles, CA, USA, in 2017. He is currently pursuing the Ph.D. degree with the Department of Computing, The Hong Kong Polytechnic University, Hong Kong.

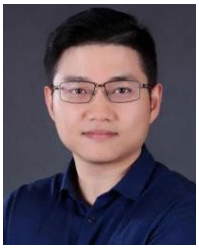
His current research interests include language models, natural language processing, and software engineering.

Chunpu Xu received the M.S. degree from the School of Artificial Intelligence and Automation, Huazhong University of Science and Technology, Wuhan, China, in 2020. He is currently pursuing the Ph.D. degree with the School of Computing, The Hong Kong Polytechnic University, Hong Kong.

His current research interests include but not limited to multimodal learning and LLM evaluation.

Jing Li received the B.S. degree in intelligence science and technology from Peking University, Beijing, China, in 2013, and the Ph.D. degree from the Department of Systems Engineering and Engineering Management, The Chinese University of Hong Kong, Hong Kong, in 2017.

She is currently an Assistant Professor at the Department of Computing, The Hong Kong Polytechnic University. Previously till 2019, she worked at Tencent AI Lab, Shenzhen, China, as a Senior Researcher. She has broad research interests in natural language processing (NLP), computational social science (CSS), and machine learning (ML). In particular, she works on novel algorithms for language representation learning, social media language understanding, conversation and social interaction modeling, and robust NLP and multimodal applications in noisy real-world applications.

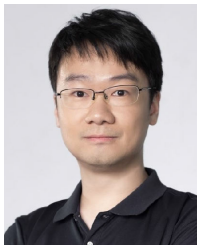


Yuqun Zhang (Member, IEEE) received the B.S. degree from Tianjin University, Tianjin, China, in 2008, the M.S. degree from the University of Rochester, Rochester, NY, USA, in 2010, and the Ph.D. degree from the University of Texas at Austin, Austin, TX, USA, in 2016.

He is an Assistant Professor at Southern University of Science and Technology, Shenzhen, China. His current research interests include software analysis, security, and testing.

Dr. Zhang has won one ACM Distinguished Paper Award in ISSTA 2019.

Zeyang Fang is currently a Senior Research and Development Engineer at Baidu's Deep Learning Technology Platform, Baidu Paddle, Beijing, China. He is currently responsible for large language model training and associated projects for the PaddleNLP large model collection built on Baidu's PaddlePaddle platform. His main research focuses on recommendation systems, graph learning, and deep learning frameworks.



Zeyu Chen is currently a Principal Architect of deep learning platform at Baidu Inc., Beijing, China, leading the research and development efforts in building open-source deep learning libraries and toolkits for NLP, speech, and miscellaneous applications all on top of PaddlePaddle. Orchestrating with crews, he serves as a Tech Lead covering three lines of products, including PaddleNLP, PaddleSpeech, and PaddleHub. His contributions to PaddlePaddle have been widely recognized by open-source communities. He has authored extensively in AI conference and journals.

Dr. Chen was a co-recipient of the 2021 Annual Technical Advancement Award from Baidu Inc.

Baohua Lai is currently a partner at TOPFLYtech, Shenzhen, China, specializing in smart cockpits, ADAS, and related fields. He was the Technical Manager of Baidu PaddlePaddle Tool Suite Team. He holds over 50 patents. His research interests mainly include 3-D vision, large visual models, and NLP.