QuoteR: A Benchmark of Quote Recommendation for Writing

Anonymous ACL submission

Abstract

It is very common to use quotations (quotes) to make our writings more elegant or convincing. To help people find appropriate quotes more efficiently, the task of quote recommendation is presented, aiming to recommend quotes that fit the current context of writing. There have been various quote recommendation approaches, but they are evaluated on different unpublished datasets. To facilitate the research on this task, we build a large and fully open quote recommendation dataset called QuoteR, which comprises three parts including English, standard Chinese and classical Chinese. Any part of it is larger than previous unpublished counterparts. We conduct an extensive evaluation of existing quote recommendation methods on QuoteR. Furthermore, we propose a new quote recommendation model that significantly outperforms previous methods on all three parts of QuoteR. All the code and data of this paper will be released.

1 Introduction

A quotation, or quote for short, is a sequence of words that someone else has said or written.[1] Quotes are quite useful in writing — they can not only help illuminate and emphasize the meaning we want to convey, but also endow our writing with elegance and credibility (Cole, 2008). As a result, the use of quotes is very common and, moreover, universal among all languages.

However, it is not an easy job for ordinary people to promptly come up with appropriate quotes that fit the current context of writing, due to the huge number of quotes. Search engines can provide some help in finding quotes by keyword matching, but it is not enough. Quotes generally express their meanings implicitly by rhetorical devices like metaphor and have different word usages from modern and everyday writing, as illustrated in Figure 1, for which quote search based on keyword matching is ineffective. In addition, some quote repository websites organize quotes by topic. However, even after filtering by topic, there are still too many candidate quotes, and selecting a suitable one does not become much easier.

> "There's an old Bible verse my dad used to say all the time that says **sufficient unto the day is the evil thereof**," Pyron said. "In other words — today has its own set of problems, we can't do anything about yesterday, and I don't want to jump too far into tomorrow."

Figure 1: An example of usage of quotes.

To tackle these challenges, Tan et al. (2015) introduce the task of quote recommendation, aiming to automatically recommend suitable quotes given the context of writing.[2] Afterward, a series of studies propose various approaches to this task (Ahn et al., 2016; Tan et al., 2016, 2018). However, these studies use different evaluation datasets, and none of them are publicly available. The lack of a standard and open dataset is undoubtedly a serious obstacle to the quote recommendation research.

In this paper, to solve this problem, we build a large quote recommendation dataset that will be publically released. This dataset is named QuoteR (abbreviated from **Quote R**ecommendataion) and composed of three parts: (1) the English part that comprises 6,108 English quotes with 126,713 contexts; (2) the standard Chinese (Mandarin) part, which contains 3,004 standard Chinese quotes with 40,842 contexts; and (3) the classical Chinese (Wenyan) part, which comprises 4,438 classical Chinese quotes (including classical poems) and 116,537 contexts. Any part of this dataset is absolutely larger than, or even doubles, previous closed-source counterparts.

We conduct a fair and extensive evaluation of existing quote recommendation methods on QuoteR

[1] In this paper, we focus on the famous quotes that mainly include proverbs, maxims and other famous sayings.

[2] This task also has great value to research, as a touchstone for NLP models' abilities in language understanding, semantic matching and linguistic coherence estimation.

with a thorough set of metrics. By analyzing these methods and their evaluation results, we find two weaknesses of these methods and propose a new method by making corresponding improvements, which we hope would serve as a strong baseline for quote recommendation.

First, most existing methods encode contexts and quotes into vectors for quote-context matching, using LSTM (Hochreiter and Schmidhuber, 1997) or CNN (Kim, 2014) as the encoders. These encoders have proven inferior to the pre-trained language models like BERT (Devlin et al., 2019), which limits the final quote recommendation performance. Therefore, we try to utilize a pre-trained language model, specifically BERT, as the sentence encoders to learn representations of quotes and contexts. Considering the huge compute resulting from the large scale of the dataset and the BERT model, it is nontrivial to train the context and quote encoders simultaneously. We design an ingenious training strategy to address this issue.

Second, it is harder to learn good representations for quotes compared with contexts, because most quotes are quite pithy, and their words usually carry rich semantics, as shown in Figure 1. Existing methods, however, do not address this challenge well. To handle this challenge, we incorporate a kind of general lexical knowledge, namely *sememes*, into the quote encoder, aiming to improve the representations of quotes. A sememe is defined as the minimum semantic unit in linguistics (Bloomfield, 1926), and the sememes of a word atomically interpret the meaning of the word. Incorporating sememes can bring more semantic information for quote representation learning and conduce to a better quote vector.

In experiments, we demonstrate that both the utilization of BERT and the incorporation of sememes substantially improve quote recommendation performance. And the sememe-incorporated BERT-based model significantly outperforms all previous methods on QuoteR. Moroever, ablation and case studies as well as human evaluation further prove its effectiveness.

To conclude, our contributions are threefold: (1) building a large and the first open quote recommendation dataset; (2) conducting an extensive and fair evaluation of existing quote recommendation methods; (3) proposing a quote recommendation model that outperforms all previous methods and can serve as a strong baseline for future research.

2 Related Work

2.1 Quote Recommendation

The task of quote recommendation is originally presented in Tan et al. (2015). They propose a learning-to-rank framework for this task, which integrates 16 hand-crafted features. Tan et al. (2016) and Tan et al. (2018) introduce neural networks to the quote recommendation task. They use LSTMs to learn distributed vector representations of contexts and quotes and conduct sentence matching with these vectors. Ahn et al. (2016) combine four different quote recommendation approaches including matching granularity adjustment (a statistical context-quote relevance prediction method), random forest, CNN and LSTM.

In addition quote recommendation for writing, some studies focus on recommending quotes in dialog. Lee et al. (2016) propose an LSTM-CNN combination model to recommend quotes according to Twitter dialog threads, i.e., sequences of linked tweets. Wang et al. (2020) utilize an encoder-decoder framework to generate quotes as response, based on the separate modeling of the dialog history and current query. Wang et al. (2021) adopt a semantic matching fashion, which encodes the multi-turn dialog history with Transformer (Vaswani et al., 2017) and GRU (Cho et al., 2014) and encodes the quote with Transformer.

In terms of the datasets of quote recommendation for writing, Tan et al. (2015) construct an English dataset comprising 3,158 quotes and 64,323 contexts extracted from e-books in Project Gutenberg.[3] Ahn et al. (2016) build a similar English dataset that contains 400 most frequent quotes with contexts from e-books in Project Gutenberg and blogs. Tan et al. (2018) build a classical Chinese poetry quotation dataset that comprises over 9,000 poem sentences with 56,949 contexts extracted from Chinese e-books on the Internet. Unfortunately, all these datasets are not publicly available.

2.2 Content-based Recommendation

Quote recommendation is essentially a kind of content-based recommendation task (Pazzani and Billsus, 2007), which is aimed at recommending products to users according to product descriptions and users' profiles.

A closely related and widely studied task is content-based citation recommendation (Strohman

[3] https://www.gutenberg.org/

et al., 2007), especially local citation recommendation that recommends related papers given a particular context of academic writing (He et al., 2010; Huang et al., 2012, 2015). Compared with quote recommendation, this task is targeted at structured documents (papers), which are much longer and possess abundant information such as title, abstract and citation relations that are useful for recommendation. Quotes are shorter and usually have no available information except the text, which renders quote recommendation more challenging.

Another highly related but niche task is idiom recommendation (Liu et al., 2018, 2019), which aims to recommend appropriate idioms for a given context. Existing idiom recommendation methods are essentially covered by the quote recommendation methods described in §2.1. Liu et al. (2018) recommend idioms by learning representations of the contexts and idioms, similar to the context-quote relevance-based quote recommendation methods (Ahn et al., 2016; Tan et al., 2018). The difference lies in the use of word embeddings of idioms rather than a sentence encoder. Liu et al. (2019) regard idiom recommendation as a context-to-idiom machine translation problem and use an LSTM-based encoder-decoder framework, which is similar to Wang et al. (2020).

2.3 Other Quote-related Tasks

In addition to quote recommendation, there are some other quote-related tasks. For example, quote detection (or recognition) that is aimed at locating spans of quotes in text (Pouliquen et al., 2007; Scheible et al., 2016; Pareti et al., 2013; Papay and Padó, 2019), and quote attribution that intends to automatically attribute quotes to speakers in the text (Elson and McKeown, 2010; O'Keefe et al., 2012; Almeida et al., 2014; Muzny et al., 2017). Different from quote recommendation that focuses on famous quotes, these tasks mainly deal with the general quotes of utterance.

3 Task Formulation

Before describing our dataset and model, we first formulate the task of quote recommendation for writing and introduce several basic concepts, most of which follow previous work (Tan et al., 2015).

For a piece of text containing a quote q, the text segment occurring before the quote is named *left context* c_l while the text segment occurring after the quote is named *right context* c_r. The concatenation of left and right contexts form the *quote context* $c = [c_l; c_r]$. Suppose there is a *quote set* that comprises all the known candidate quotes $Q = \{q_1, \cdots, q_{|Q|}\}$, where $|\cdot|$ denotes the cardinality of a set.

In the task of quote recommendation for writing, a *query context* c is given, and the *gold quote* q_c is wanted, where the query context is the context provided by the user and the gold quote is the quote in the quote set that fits the query context best. Theoretically, a query context may have more than one gold quote because there are some quotes that convey almost the same meaning. Following previous work (Tan et al., 2015; Ahn et al., 2016), for simplicity, we only regard the quote that actually appears together with the query context in corpora as the gold quote.

For a quote recommendation model, given the quote set Q, its input is a query context $c = [c_l; c_r]$, and it is supposed to calculate a rank score for each candidate quote in Q and output a quote list according to the descending rank scores.

4 Dataset Construction

In this section, we present the building process and details of the QuoteR dataset.

4.1 The English Part

We begin with the English part. We choose the popular and free quote repository website Wikiquote[4] as the source of English quotes. We download its official dump and extract over 60,000 English quotes in total to form the quote set. We notice that previous work (Tan et al., 2015; Ahn et al., 2016) collects quotes from another website named Library of Quotes, but this website has closed down.

To obtain real contexts of quotes, we use three corpora. The first is the Project Gutenberg corpus that previous studies use, which comprises over 50,000 e-books. The second corpus is BookCorpus containing about 11,000 e-books (Zhu et al., 2015). In addition to the two book corpora, we use the OpenWebText corpus (Gokaslan and Cohen, 2019) which is composed of text from web pages and has different text styles from books. The total size of the raw text of the three corpora reaches 48.8 GB.

We search all the corpora for the occurrences of quotes in the quote set. Some quotes are composed of multiple sentences, and only part of them are cited in some cases. To cope with this situation,

[4] https://en.wikiquote.org

Part	Train	Validation	Test	Total
English	101,171/6,008	12,771/6,108	12,771/6,108	126,713/6,108
sChinese	32,472/2,904	4,185/3,004	4,185/3,004	40,842/3,004
cChinese	93,031/4,338	11,753/4,438	11,753/4,438	116,537/4,438

Table 1: Statistics of the three parts of QuoteR. sChinese and cChinese refer to standard and classical Chinese, respectively. Each item like m/n means m context-quote pairs involving n quotes. Appendix A gives more detailed statistics.

we split each quote into sentences using Stanza (Qi et al., 2020) and then search for each constituent sentence in the corpora. If multiple constituent sentences of a quote appear sequentially, we combine them into an occurrence of the quote. Compared with previous work that searches for quotes as a whole (Tan et al., 2015; Ahn et al., 2016), we can find more quote occurrences.

For each quote occurrence, we take the 40 words preceding and following it as its left and right contexts, respectively. The concatenation of the left and right contexts forms a context, and a context and the corresponding quote form a context-quote pair. We remove the repeated context-quote pairs and filter out the quotes appearing less than 5 times in the corpora. To avoid dataset imbalance, we randomly select 200 context-quote pairs for a quote appearing more than 200 times and discard its other context-quote pairs. Finally, we obtain 126,713 context-quote pairs involving 6,108 different quotes, which form the English part of QuoteR.

We split all the context-quote pairs into training, validation and test sets roughly in the ratio 8:1:1, making sure that all the quotes appear in the validation and test sets while 100 quotes do not appear in the training set. We split the dataset in this way in order to observe how quote recommendation models perform in the zero-shot situation, where the model has never seen the gold quote of some validation/test contexts during training. The statistics of the final split dataset are listed in Table 1.

4.2 The Standard Chinese Part

We gather standard Chinese quotes from a large quote collection website named Juzimi[5]. More than 32,000 standard Chinese quotes are collected altogether. To obtain quote contexts, we use two corpora including a corpus composed of answer text from a Chinese QA website[6] and a large-scale book corpus that we specifically build and comprises over 8,000 free Chinese e-books. The total size of the two corpora is about 32 GB.

Then we use the same method in building the English part to extract quote occurrences from the corpora. Since Chinese is not naturally word-segmented, we take the 50 characters (rather than words) before and after a quote occurrence as the left and right contexts. In addition, since there are fewer quotes and contexts for the standard Chinese part, we reduce the minimum number of occurrences for a selected quote to 3, and the maximum number of retained contexts per quote to 150. After deduplication and filtering, we obtain the standard Chinese part of QuoteR, which has 40,842 context-quote pairs involving 3,004 quotes.

We split the standard Chinese part in the same way as the English part, and the statistics are also shown in Table 1.

4.3 The Classical Chinese Part

Classical Chinese quotes, including classical poems and proverbs, are often cited in standard Chinese writing. Considering that classical Chinese is very different from standard Chinese, we separate classical Chinese quotes from standard Chinese ones. We collect over 17,000 classical Chinese quotes from Gushiwenwang,[7] a classical Chinese poetry and literature repository website, and aforementioned Juzimi.[8]

Then we adopt the same way as standard Chinese to extract context-quote pairs from the two Chinese corpora and conduct deduplication and filtering. Finally, we obtain the classical Chinese part of QuoteR that comprises 116,537 context-quote pairs of 4,438 quotes. The statistics of this part after splitting are also in Table 1.

4.4 Quality Assessment by Human

After the construction of QuoteR, we assess its quality by human. For each part, we randomly sample 100 context-quote pairs, and ask three annotators to independently determine whether each quote fits the corresponding context. The final results are obtained by voting. Finally, 99/98/94 context-quote pairs are regard as suitable for the three parts, respectively. The results verify the quality of QuoteR, which is expected because the data are extracted from high-quality corpora like books.

[5] https://www.juzimi.com/

[6] https://github.com/brightmart/nlp_chinese_corpus

[7] https://www.gushiwen.org/

[8] Juzimi provides the dates when the quotes appear so that we can distinguish classical and standard Chinese quotes.

5 Methodology

In this section, we elaborate on our proposed quote recommendation model. This model is based on the representative pre-trained language model BERT (Devlin et al., 2019), but can be readily adapted to other pre-trained language models.

5.1 Basic Framework

Similar to most previous methods (Tan et al., 2016; Ahn et al., 2016), we use BERT as the text encoder to learn vector representations of contexts and quotes, and then calculate the similarity between the representations of the query context and a candidate quote as the rank score of the quote.

Learning Representations of Quotes

We first obtain the representations of quotes. Formally, for a candidate quote comprising m tokens $q = \{x_1, \cdots, x_m\} \in Q$, we feed it into BERT and obtain a series of hidden states:

$$\mathbf{h}^q_{\texttt{[C]}}, \mathbf{h}^q_1, \cdots, \mathbf{h}^q_m = \text{BERT}^q(\texttt{[C]}, x_1, \cdots, x_m), \quad (1)$$

where `[C]` denotes the special `[CLS]` token in BERT that is added to the front of a sequence. Following Devlin et al. (2019), we use the hidden state of `[C]` as the representation of the quote: $\mathbf{q} = \mathbf{h}^q_{\texttt{[C]}}$. The representations of all quotes form the quote representation matrix $\mathbf{Q} = [\mathbf{q}_1, \cdots, \mathbf{q}_{|Q|}]$.

Learning Representations of Contexts

We can use another BERT as the context encoder to obtain the representation of the query context $c = [c_l; c_r]$. Considering the context is composed of left and right contexts that are not naturally joined, we can insert an additional separator token between them before feeding them into BERT:

$$\mathbf{h}^c_{\texttt{[C]}}, \cdots = \text{BERT}^c(\texttt{[C]}, c_l, \texttt{[S]}, c_r), \quad (2)$$

where `[S]` is the sentence separator token `[SEP]` in BERT. We can also use the hidden state of `[C]` as the representation of the context: $\mathbf{c} = \mathbf{h}^c_{\texttt{[C]}}$.

However, it is actually inconsistent with the general use of BERT. Whether in pre-training or fine-tuning, when the input to BERT is two text segments connected by the separator token, the hidden state of `[CLS]` is only used to classify the relation between the two segments, e.g., to predict whether the second segment is the actual next sentence of the first segment in the next sentence prediction (NSP) pre-training task (Devlin et al., 2019).

We turn to another pre-training task of BERT, masked language modeling (MLM), which is a cloze task (Taylor, 1953) aimed at predicting masked tokens. Specifically, some tokens in a text sequence are randomly substituted by the special `[MASK]` tokens and the hidden states of the `[MASK]` tokens are fed into a classifier to predict the original tokens. Quote recommendation given context can be regarded as a special cloze task whose object of prediction is quotes rather than tokens. Inspired by the MLM pre-training task, we propose another way to learn the context representation by inserting an additional `[MASK]` token:

$$\mathbf{h}^c_{\texttt{[C]}}, \cdots, \mathbf{h}^c_{\texttt{[M]}}, \cdots = \text{BERT}^c(\texttt{[C]}, c_l, \texttt{[M]}, c_r), \quad (3)$$

where `[M]` is the `[MASK]` token. We use the hidden state of `[M]` as the representation of the query context: $\mathbf{c} = \mathbf{h}^c_{\texttt{[M]}}$.[9]

Calculating Rank Scores of Candidate Quotes

After obtaining the representations of all candidate quotes and the query context, the rank score of a candidate quote can be calculated by softmax:

$$\mathbf{p} = \text{softmax}(\mathbf{Q}^\top \mathbf{c}), \quad (4)$$

where $\mathbf{p}$ is a normalized probability vector whose i-th element is the rank score of the i-th quote.

5.2 Training Strategy

As in previous work (Tan et al., 2016), we can simply use the cross-entropy loss to train the quote and context encoders simultaneously. However, there are two problems. (1) For each context in the training set, the quote encoder needs to be updated for every quote in the quote set. In other words, the BERT-based quote encoder would be fine-tuned thousands of times per training instance, which requires formidably big GPU memory and long training time.[10] (2) The huge imbalance between positive and negative samples (one vs. several thousands) would weaken the capacity of the quote encoder and, in turn, impair the final quote recommendation performance.

A simple solution is to freeze the quote encoder during training, i.e., use the raw pre-trained BERT as the quote encoder, and train the context encoder only. But the untrained quote encoder would decrease final quote recommendation performance, as demonstrated in later experiments. To address these issues, inspired by the study on noise contrastive

[9] The hidden state of `[M]` can also be regarded as the representation of the *required* quote for the query context. In this view, the rank score in Eq. (4) is actually calculated by the similarity between a candidate quote and the required quote.

[10] We find that four 16-GB GPUs would be out of memory during training even though we set the batch size to 1.

estimation (NCE) (Gutmann and Hyvärinen, 2012), we adopt the negative sampling strategy in training. For each context-quote pair, we select some non-gold quotes as negative samples, and calculate a pseudo-rank score of the gold quote among the selected quotes. Formally, for a context-quote pair (c, q), the pseudo-rank score of q is

$$p^* = \frac{e^{\mathbf{q}\cdot\mathbf{c}}}{e^{\mathbf{q}\cdot\mathbf{c}} + \sum_{q^*\in\mathbb{N}(q)} e^{\mathbf{q}^*\cdot\mathbf{c}}}, \quad (5)$$

where $\mathbb{N}(q)$ is the set of quotes selected as negative samples. Then the training loss is the cross-entropy based on the pseudo-rank score: $\mathcal{L} = -\log(p^*)$.

The problem about quote encoder training has been largely solved, but the context encoder may be under-trained. The context encoder needs to process lots of contexts and thus requires more training than the quote encoder. Therefore, we adopt a two-stage training strategy. After the simultaneous training of quote and context encoders in the first stage, we continue to train the context encoder while freezing the quote encoder in the second stage. The training loss of the second stage is the cross-entropy loss among all quotes.

5.3 Incorporation of Sememes

Most quotes are quite pithy, and thus it is usually hard to learn their representations well. To obtain better quote representations, previous work tries incorporating external information, including the topic and author information of quotes, in the quote encoder (Tan et al., 2016, 2018). Although helpful, this external information is not always available or accurate — quite a few quotes are anonymous, and the topics attributed to quotes are usually from crowdsourcing and uninspected.

We propose to incorporate sememe knowledge into quote representation learning, which is more general (every word can be annotated with sememes) and credible (the sememe annotations of words are given by experts). A sememe is the minimum semantic unit of human languages (Bloomfield, 1926), and it is believed that meanings of all words can be represented by a limited set of sememes. Sememe knowledge bases like HowNet (Dong and Dong, 2006) use a set of predefined sememes to annotate words, so that the meaning of a word can be precisely expressed by its sememes. With the help of such sememe knowledge bases, sememe knowledge has been successfully utilized in various NLP tasks (Qi et al., 2021).

Inspired by the studies on incorporating sememes into recurrent neural networks (Qin et al., 2020) and transformers (Zhang et al., 2020) to improve their representation learning ability, we adopt a similar way to incorporate sememes into the quote encoder. We simply add the average embedding of a word's sememes to every token embedding of the word in BERT. Formally, for a word in a quote that is divided into n tokens after tokenization $w = x_1, \cdots, x_n$, the embedding of its each token $\mathbf{x}_i$ is transformed into

$$\mathbf{x}_i \to \mathbf{x}_i + \frac{\alpha}{|\mathbb{S}(w)|} \sum_{s_j\in\mathbb{S}(w)} \mathbf{s}_j, \ \forall i = 1, \cdots, n \quad (6)$$

where $\mathbb{S}(w)$ is the sememe set of the word w, and α is a hyper-parameter controlling the weight of sememe embeddings. Following previous work (Qin et al., 2020), the sememe embeddings are randomly initialized and updated during training.

6 Experiments

In this section, we evaluate our model and previous quote recommendation methods on QuoteR.

6.1 Approaches for Comparison

We have three groups of approaches for comparison. The first group consists of two methods that widely serve as baselines in previous studies. (1.1) **CRM**, namely context-aware relevance model (He et al., 2010) that recommends the quote whose known contexts are most similar to the query context. (1.2) **LSTM**, which uses two LSTM encoders to learn representations of quotes and contexts.

The second group includes representative approaches proposed in previous studies. (2.1) **top-k RM**, namely top-k rank multiplication (Ahn et al., 2016), which is a rank aggregation method based on the ensemble of a statistical method, random forest, CNN and LSTM. (2.2) **NNQR** (Tan et al., 2016), which reforms LSTM by incorporating additional quote information (topic and author) into the quote encoder and perturbing the word embeddings of quotes. (2.3) **N-QRM** (Tan et al., 2018), which further improves NNQR mostly by adjusting the training loss to prevent overfitting. (2.4) **Transform** (Wang et al., 2021), which uses Transformer+GRU to encode contexts and transforms context embeddings into the space of quote embeddings learned from another Transformer.[11]

The third group comprises two BERT-based approaches that are frequently utilized in sen-

[11] It is originally designed to recommend quotes in dialog, and we adapt it to the writing situation. It is also the only adaptable method of other content-based recommend tasks.

Dataset	English				Standard Chinese				Classical Chinese			
Model	MRR	NDCG	$\tilde{R}$/$\bar{R}$/σ_R	Recall@1/10/100	MRR	NDCG	$\tilde{R}$/$\bar{R}$/σ_R	Recall@1/10/100	MRR	NDCG	$\tilde{R}$/$\bar{R}$/σ_R	Recall@1/10/100
CRM	0.192	0.193	599/1169/1408	16.51/23.66/32.78	0.397	0.407	13/325/584	33.60/49.32/61.70	0.198	0.203	166/548/811	14.52/28.79/44.51
LSTM	0.321	0.320	30/334/727	27.23/40.78/62.47	0.292	0.290	48/338/574	24.78/37.71/58.06	0.247	0.245	56/341/633	20.08/33.23/56.96
top-k RM	0.422	0.431	6/548/1243	35.99/53.31/66.20	0.480	0.494	3/377/774	40.17/60.67/72.26	0.294	0.299	48/511/980	23.54/39.58/56.90
NNQR	0.318	0.319	31/359/773	26.78/41.10/61.29	0.271	0.271	54/348/595	22.94/35.72/57.18	0.272	0.270	41/310/620	22.03/36.59/60.63
N-QRM	0.365	0.368	28/777/1465	32.24/44.41/58.26	0.343	0.347	55/575/890	30.20/41.22/54.15	0.287	0.288	98/917/1373	24.88/35.02/49.49
Transform	0.561	0.568	1/241/749	50.11/65.88/79.98	0.512	0.519	2/271/576	45.50/60.31/72.83	0.449	0.453	5/269/663	39.01/55.78/73.58
BERT-Sim	0.526	0.529	2/487/1064	49.38/58.05/67.75	0.500	0.508	2/229/511	44.47/59.07/72.21	0.439	0.443	7/320/711	38.85/53.04/68.32
BERT-Cls	0.310	0.329	7/134/453	18.15/57.11/82.05	0.378	0.395	5/152/413	26.88/57.90/78.38	0.330	0.345	8/**135/377**	21.93/54.27/78.75
Ours	**0.572**	**0.580**	1/**123/433**	50.74/**69.03/83.84**	**0.541**	**0.548**	2/**139/370**	**47.91/64.97/79.35**	**0.484**	**0.490**	3/146/422	**41.67/60.78/79.38**
-Sememe	0.568	0.574	1/145/492	**51.05**/67.07/82.34	0.535	0.543	2/160/402	47.62/63.66/77.68	0.475	0.481	3/152/435	40.93/60.26/78.39
-ReTrain	0.299	0.307	12/176/503	20.46/47.89/75.74	0.255	0.260	20/210/435	16.87/42.94/68.43	0.265	0.269	17/184/450	17.87/43.56/72.89
-SimTrain	0.529	0.532	2/467/1060	49.31/58.97/69.48	0.519	0.526	2/204/489	46.00/62.03/75.34	0.465	0.470	4/310/713	41.40/55.53/70.09

Table 2: Quote recommendation results of different models on the three parts of QuoteR. Recall@1/10/10 is percentage. The **boldfaced** results exhibit statistically significant improvement over the other results with $p<0.1$ given by paired t-tests, and the underlined results mean no significant difference.

tence matching and sentence pair classification. (3.1) **BERT-Sim**, which is the vanilla BERT-based model discussed in §5.1. It directly uses the hidden states of the `[CLS]` tokens as the representations of both quotes and contexts, and freezes the quote encoder during training, as explained in §5.2. (3.2) **BERT-Cls**, which conducts a binary classification for the concatenation of the query context and a candidate quote.

6.2 Evaluation Metrics

Following previous work (Ahn et al., 2016; Tan et al., 2018), we use three evaluation metrics: (1) Mean reciprocal rank (**MRR**), the average reciprocal values of the ranks of the gold quotes; (2) Normalized discounted cumulative gain (**NDCG@K**) (Järvelin and Kekäläinen, 2002), a widely used measure of ranking quality and is computed by

$$\text{NDCG@K} = Z_K \sum_{i=1}^{K} \frac{2^{r(i)}-1}{\log_2(i+1)}, \qquad (7)$$

where $r(i) = 1$ if the i-th quote is the gold quote, otherwise $r(i) = 0$, $Z_K = 1$ is a normalization constant. We report the average of NDCG@5 scores of all the evaluated query contexts. (3) **Recall@K**, the proportion of query contexts whose gold quotes are ranked in respective top K candidate quotes, $K = \{1, 10, 100\}$.

Besides, we use another three evaluation metrics: (4) **Median Rank** ($\tilde{R}$), (5) **Mean Rank** ($\bar{R}$) and (6) **Rank Variance** (σ_R), the median, average and standard deviation of the ranks of gold quotes.

The higher MRR, NDCG@K and Recall@K and the lower $\tilde{R}$, $\bar{R}$ and σ_R are, the better a model is.

6.3 Implementation Details

We use $\text{BERT}_{\text{BASE}}$ for both English and Chinese from Transformers (Wolf et al., 2020). We use the AdamW optimizer (Loshchilov and Hutter, 2018) with an initial learning rate 5e-5 that gradually declines to train our model. We randomly select N negative samples, and N is tuned in {4,9,19,29,39} on the validation set. The weight of sememe embeddings α is tuned in {0.1, 0.2, 0.5, 1.0, 2.0}. The underlined numbers are final picks. For the previous methods, we use their original hyperparameters and experimental settings given in the papers.

6.4 Experimental Results

Table 2 lists the evaluation results of different methods on the three parts of QuoteR.[12] We observe that (1) our method achieves the best overall results and displays its superiority to other methods; (2) the two BERT-based models, especially BERT-Sim, yield quite high performance, which reflects the importance of a powerful sentence encoder to quote recommendation; (3) among the three parts, almost all methods perform worse on Classical Chinese, which is presumably because Chinese BERT is pre-trained on standard Chinese corpora and not suitable to encode the classical Chinese quotes.

Ablation Study

We conduct ablation studies to investigate the effectiveness of our training strategy and the incorporation of sememes. We first remove the incorporation of sememes (-Sememe), then further do not separately train the context encoder after the simultaneous training of the context and quote encoders (-ReTrain), and finally discard the simultaneous training of the two encoders and train the context encoder only (-SimTrain). -SimTrain differs BERT-Sim only in the choice of context representation (`[MASK]` vs. `[CLS]`).

The results of ablation studies are given in the last three rows of Table 2. We have the follow-

[12] We also conduct evaluation in the setting where only the left context is given. The results are in Appendix B.

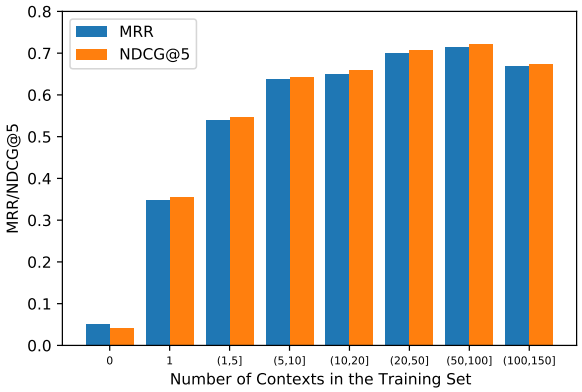

Figure 2: Recommendation performance for quotes within different occurrence frequency ranges. The quote numbers in the ranges are 100, 843, 985, 437, 283, 225, 74 and 47, respectively.

ing observations: (1) -Sememe causes consistent performance decline as compared to Ours, which demonstrates the role of sememes in improving quote encoding, thereby benefiting quote recommendation; (2) the performance of -ReTrain is pretty poor, which reflects the necessity of separate training for the context encoder after simultaneous training; (3) -SimTrain is inferior to -Sememe, which displays the usefulness of simultaneously training the two encoders; (4) -SimTrain outperforms BERT-Sim, proving the superiority of choosing `[MASK]` to represent contexts in our method.

6.5 Effect of Occurrence Frequency

In this subsection, we investigate the effect of the gold quote's occurrence frequency on recommendation performance. Figure 2 shows MRR and NDCG@5 results for quotes that have different numbers of contexts in the training set of the standard Chinese part. We observe that the occurrence frequency has great impact on quote recommendation performance. Basically, increasing occurrences of quotes in the training set can increase recommendation performance, because we can learn better representations for the quotes with more adequate training. But the most frequent quotes does not have the best performance, possibly because these quotes carry very rich semantics and can be cited in various contexts , which makes it very hard to correctly recommend them. In addition, the performance for the unseen quotes is very limited. It reflects the weakness of our model in the zero-shot situation, whose solution is left for future work.

6.6 Human Evaluation

As mentioned in §3, there may be other quotes that are suitable for a query context besides the gold quote. Hence, we conduct a human evaluation on the recommendation results of our method. We randomly select 50 contexts from the validation set of the standard Chinese part and list the top 10 quotes recommended by our method for each context. Then we ask annotators to make a binary suitability decision for the quotes. Each quote is annotated by 3 native speakers and the final decision is made by voting. For each context, we regard the suitable quote with the highest ranking as the gold quote, and re-evaluate the recommendation performance: NDCG@5=0.661, Recall@1/10=0.50/0.92.[13] In contrast, the original evaluation results among the 50 contexts are NDCG@5=0.439 , Recall@1/10=0.36/0.64. By comparison, we can conclude that the real performance of our method is substantially underestimated. We also count the average number of suitable quotes among the top 10 quotes, which is 1.76.

Rank	Quote	Score
1	sufficient for the day is its own trouble	0.723
2	**sufficient unto the day is the evil thereof**	0.124
3	you can never plan the future by the past	0.060
4	tomorrow will be a new day	0.025
5	the darkest hour is just before the dawn	0.008

Table 3: Top 5 results for the context in Figure 1.

6.7 Case Study

We feed the context in Figure 1 into our model, and print the top 5 recommended quotes and their rank scores in Table 3. We find that the gold quote is ranked 2nd, but the first one is actually another statement version of the gold quote and has exactly the same meaning. In addition, the 3rd and 4th quotes are also related to the context. More cases are given in Appendix D due to space limit.

7 Conclusion and Future Work

In this paper, we build a large and the first open quote recommendation dataset QuoteR and conduct an extensive evaluation of existing quote recommendation methods on it. We also propose a new model that achieves absolute outperformance over previous methods, and its effectiveness is proved by ablation studies. In the future, we will try to improve our model in handling classical Chinese quotes by using a special classical Chinese pre-trained model to encode them. We will also consider boosting the performance of our model in the zero-shot situation.

[13] Since we only annotate the top 10 results, there are no other available metrics than NDCG@5 and Recall@1/10.

8 Ethical Statements

In this section, we discuss the ethical considerations of this paper from four perspectives.

Dataset and Human Evaluation In terms of our QuoteR dataset, all the quotes are collected from free and open quote repository websites. Besides, all the contexts are extracted from open corpora, including free public domain e-books and other open corpora. Therefore, there is no intellectual property problem for the dataset. In addition, we conduct the human evaluation by a reputable data annotation company. The annotators are fairly compensated by the company, based on the previous annotation tasks. Further, we do not directly communicate with the annotators, so that their privacy is well preserved. Finally, the dataset and the human evaluation are not sensitive and thus do not need to be approved by the institutional review board (IRB).

Application Quote recommendation is a practical task and our model can be put into service. In actual use cases, users just need to input a query context and our model should output a list of candidate quotes that fit the given context. All people may benefit from our model during writing. If our model fails, some inappropriate quotes that cannot fit the query context would be output, but no one would be harmed. There are indeed biases in the dataset we build. Some quotes are very frequent while the others are not, as illustrated in §6.5. The infrequent quotes are less recommended and may cause the failure of our model in some cases. In terms of misuse, to the best of our knowledge, such a quote recommendation model is hardly misused. After the deployment of our model, the system would not collect data from users. It does not have any potential harm to vulnerable populations, either.

Energy Saving To save energy, we use the base version of BERT rather than larger pre-trained language models, although the larger ones would probably yield better performance. Besides, as discussed in §5.2, we find that the simultaneous training of the context and quote encoders requires very big memory and computation resources, and thus we adopt the strategy of negative sampling in training.

Use of Identity Characteristics In this work, we do not use any demographic or identity characteristics information.

References

Yeonchan Ahn, Hanbit Lee, Heesik Jeon, Seungdo Ha, and Sang-goo Lee. 2016. Quote recommendation for dialogs and writings. In *Proceedings of CBRecSys*.

Mariana SC Almeida, Miguel B Almeida, and André FT Martins. 2014. A joint model for quotation attribution and coreference resolution. In *Proceedings of EACL*.

Leonard Bloomfield. 1926. A set of postulates for the science of language. *Language*, 2(3):153–164.

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

Peter Cole. 2008. *How to Write: News Writing*. The Guardian.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL-HLT*.

Zhendong Dong and Qiang Dong. 2006. *HowNet and the Computation of Meaning (With CD-Rom)*. World Scientific.

David K Elson and Kathleen R McKeown. 2010. Automatic attribution of quoted speech in literary narrative. In *Proceedings of AAAI*.

Aaron Gokaslan and Vanya Cohen. 2019. Openwebtext corpus. http://Skylion007.github.io/OpenWebTextCorpus.

Michael U Gutmann and Aapo Hyvärinen. 2012. Noise-contrastive estimation of unnormalized statistical models, with applications to natural image statistics. *Journal of Machine Learning Research*, 13(2).

Qi He, Jian Pei, Daniel Kifer, Prasenjit Mitra, and Lee Giles. 2010. Context-aware citation recommendation. In *Proceedings of WWW*.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Computation*, 9(8):1735–1780.

Wenyi Huang, Saurabh Kataria, Cornelia Caragea, Prasenjit Mitra, C Lee Giles, and Lior Rokach. 2012. Recommending citations: translating papers into references. In *Proceedings of CIKM*.

Wenyi Huang, Zhaohui Wu, Chen Liang, Prasenjit Mitra, and C Lee Giles. 2015. A neural probabilistic model for context based citation recommendation. In *Proceedings of AAAI*.

Kalervo Järvelin and Jaana Kekäläinen. 2002. Cumulated gain-based evaluation of ir techniques. *ACM Transactions on Information Systems (TOIS)*, 20(4):422–446.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *Proceedings of EMNLP*.

Hanbit Lee, Yeonchan Ahn, Haejun Lee, Seungdo Ha, and Sang-goo Lee. 2016. Quote recommendation in dialogue using deep neural network. In *Proceedings of SIGIR*.

Yuanchao Liu, Bingquan Liu, Lili Shan, and Xin Wang. 2018. Modelling context with neural networks for recommending idioms in essay writing. *Neurocomputing*, 275:2287–2293.

Yuanchao Liu, Bo Pang, and Bingquan Liu. 2019. Neural-based chinese idiom recommendation for enhancing elegance in essay writing. In *Proceedings of ACL*.

Ilya Loshchilov and Frank Hutter. 2018. Fixing weight decay regularization in adam.

Grace Muzny, Michael Fang, Angel Chang, and Dan Jurafsky. 2017. A two-stage sieve approach for quote attribution. In *Proceedings of EACL*.

Tim O'Keefe, Silvia Pareti, James R Curran, Irena Koprinska, and Matthew Honnibal. 2012. A sequence labelling approach to quote attribution. In *Proceedings of EMNLP-CoNLL*.

Sean Papay and Sebastian Padó. 2019. Quotation detection and classification with a corpus-agnostic model. In *Proceedings of RANLP*.

Silvia Pareti, Tim O'keefe, Ioannis Konstas, James R Curran, and Irena Koprinska. 2013. Automatically detecting and attributing indirect quotations. In *Proceedings of EMNLP*.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. 2019. Pytorch: An imperative style, high-performance deep learning library. In *Proceedings of NeurIPS*.

Michael J Pazzani and Daniel Billsus. 2007. Content-based recommendation systems. In *The Adaptive Web*, pages 325–341. Springer.

Bruno Pouliquen, Ralf Steinberger, and Clive Best. 2007. Automatic detection of quotations in multilingual news. In *Proceedings RANLP*.

Fanchao Qi, Ruobing Xie, Yuan Zang, Zhiyuan Liu, and Maosong Sun. 2021. Sememe knowledge computation: a review of recent advances in application and expansion of sememe knowledge bases. *Frontiers of Computer Science*, 15(5):1–11.

Peng Qi, Yuhao Zhang, Yuhui Zhang, Jason Bolton, and Christopher D Manning. 2020. Stanza: A python natural language processing toolkit for many human languages. In *Proceedings of ACL*.

Yujia Qin, Fanchao Qi, Sicong Ouyang, Zhiyuan Liu, Cheng Yang, Yasheng Wang, Qun Liu, and Maosong Sun. 2020. Improving sequence modeling ability of recurrent neural networks via sememes. *TASLP*, 28:2364–2373.

Christian Scheible, Roman Klinger, and Sebastian Padó. 2016. Model architectures for quotation detection. In *Proceedings of ACL*.

Trevor Strohman, W Bruce Croft, and David Jensen. 2007. Recommending citations for academic papers. In *Proceedings of SIGIR*.

Jiwei Tan, Xiaojun Wan, Hui Liu, and Jianguo Xiao. 2018. Quoterec: Toward quote recommendation for writing. *ACM Transactions on Information Systems (TOIS)*, 36(3):1–36.

Jiwei Tan, Xiaojun Wan, and Jianguo Xiao. 2015. Learning to recommend quotes for writing. In *Proceedings of AAAI*.

Jiwei Tan, Xiaojun Wan, and Jianguo Xiao. 2016. A neural network approach to quote recommendation in writings. In *Proceedings CIKM*.

Wilson L Taylor. 1953. "cloze procedure": A new tool for measuring readability. *Journalism Quarterly*, 30(4):415–433.

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

Lingzhi Wang, Jing Li, Xingshan Zeng, Haisong Zhang, and Kam-Fai Wong. 2020. Continuity of topic, interaction, and query: Learning to quote in online conversations. In *Proceedings of EMNLP*.

Lingzhi Wang, Xingshan Zeng, and Kam-Fai Wong. 2021. Quotation recommendation and interpretation based on transformation from queries to quotations. In *Proceedings of ACL-IJCNLP*.

Thomas Wolf, Julien Chaumond, Lysandre Debut, Victor Sanh, Clement Delangue, Anthony Moi, Pierric Cistac, Morgan Funtowicz, Joe Davison, Sam Shleifer, et al. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of EMNLP*.

Yuhui Zhang, Chenghao Yang, Zhengping Zhou, and Zhiyuan Liu. 2020. Enhancing transformer with sememe knowledge. In *Proceedings of the 5th Workshop on Representation Learning for NLP*.

Yukun Zhu, Ryan Kiros, Rich Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *Proceedings of ICCV*.

A More Statistics of QuoteR

We count the numbers of quotes within different ranges of context-quote pair numbers, and the results are shown in Table 4. We can see the long tail, i.e., most quotes occur a few times while a small amount of quotes appear very frequently, which demonstrates the necessity of restricting the maximum number of contexts for a quote during the construction of QuoteR.

English Part					
#Context	[5,10]	(10,20]	(20,50]	(50,100]	(100,200]
#Quote	2,994	1,456	1,233	257	168
Standard Chinese Part					
#Context	[5,10]	(10,20]	(20,50]	(50,100]	(100,150]
#Quote	2,207	371	272	79	75
Classical Chinese Part					
#Context	[3,10]	(10,20]	(20,50]	(50,100]	(100,150]
#Quote	1,995	1,074	761	316	292

Table 4: The distribution of quotes within different occurrence frequency (the number of context-quote pairs) ranges of the three parts of QuoteR.

B Quote Recommendation with Left Context Only

Following previous work (Tan et al., 2015; Ahn et al., 2016; Tan et al., 2018), the evaluation experiments are conducted in the setting where both the left and right contexts are given. However, in practical terms, quote recommendation given the left context only might be more useful. Therefore, we also conduct experiments in the setting where only the left context is given. Table 6 shows the results. We can see that our method is still the best one on all three parts. In addition, the performance of all methods decreases substantially, which indicates that both the left and right contexts provide important information for quote recommendation.

C Effect of Negative Sample Number

In this subsection, we investigate the effect of the negative sample number (#NS), a hyper-parameter of our method, on quote recommendation performance. Table 5 gives the results of different negative sample numbers on the validation set of the standard Chinese part of QuoteR.

We can see that increasing negative samples (from 4 to 19) can increase quote recommendation performance, which is because the quote encoder can be trained more sufficiently. However, when the negative samples continue increasing, the performance fluctuates or even decreases. That is possibly because of the imbalance of positive and negative samples (there is only one positive sample, namely the gold quote), as explained in §5.2. Therefore, taking both performance and computation efficiency into consideration, we choose 19 as the final negative sample number.

#NS	MRR	NDCG	$\tilde{R}$ / $\bar{R}$ / σ_R	Recall@1/10/100
4	0.533	0.540	2 / 161 / 412	47.48 / 63.23 / 77.68
9	0.534	0.541	2 / 148 / 381	47.50 / 63.97 / 78.83
19	0.541	0.548	2 / 139 / 370	**47.91** / 64.97 / 79.35
29	**0.545**	**0.552**	2 / 174 / 434	47.06 / 63.58 / 76.92
39	0.535	0.543	2 / 132 / **357**	47.17 / 64.97 / 79.43

Table 5: Quote recommendation results with different negative sample numbers (#NS). The **boldfaced** results exhibit statistically significant improvement over the other results with $p<0.1$ given by paired t-tests, and the underlined results mean no significant difference.

D More Case Studies

Table 7-9 show three quote recommendation cases for English, standard and classical Chinese, respectively. (1) For the standard Chinese case in Table 7, the gold quote is also ranked first properly. Moreover, the 2nd and 5th recommendations, which convey the meaning of "change is the only constant thing in the world", also fit the given context. (2) For the English case in Table 8, the gold quote is correctly ranked first. And the 2nd and 5th recommended quotes have the same meaning as the gold one, and thus suitable for the context as well. (3) For the classical Chinese case in Table 9, the gold quote receives the highest rank score once again. And the 2nd recommended quote actually suits the context too. In addition, the 4th quote is also semantically related to the meaning of the context. The three cases can demonstrate the effectiveness and practicability of our quote recommendation model.

E Reproducibility

In this section, we report more experimental details to ensure the reproducibility of this paper.

All the experiments are conducted on a server that has 32 Intel(R) Xeon(R) Platinum 8163 @2.50GHz CPUs and 4 16-GB Nvidia Tesla V100 GPUs. The operation system is Ubuntu 18.04. We use Python v3.6.9 and PyTorch (Paszke et al., 2019) v1.7.1 to implement our model. More details about the implementation, e.g., dependency libraries, can

Dataset	English				Standard Chinese				Classical Chinese			
Model	MRR	NDCG	$\bar{R}/\tilde{R}/\sigma_R$	Recall@1/10/100	MRR	NDCG	$\bar{R}/\tilde{R}/\sigma_R$	Recall@1/10/100	MRR	NDCG	$\bar{R}/\tilde{R}/\sigma_R$	Recall@1/10/100
CRM	0.154	0.156	353/948/1297	11.88/21.78/33.66	0.292	0.296	124/401/524	25.28/35.39/48.43	0.141	0.146	276/587/763	9.88/19.75/34.57
LSTM	0.272	0.271	89/552/992	23.38/33.87/51.12	0.210	0.208	146/483/662	18.26/27.67/45.50	0.182	0.178	117/465/750	13.87/25.44/47.80
top-k RM	0.360	0.366	30/833/1497	31.20/44.55/56.80	0.350	0.358	38/620/926	29.77/44.40/55.53	0.276	0.280	77/645/1088	22.61/36.16/52.57
NNQR	0.267	0.266	98/592/1043	22.82/33.48/50.28	0.224	0.223	145/495/683	17.16/27.67/45.81	0.189	0.187	98/441/766	14.18/26.86/50.29
N-QRM	0.270	0.272	156/1145/1735	23.40/33.18/46.54	0.266	0.270	287/778/946	21.27/30.63/42.32	0.215	0.215	356/1232/1505	17.72/27.13/40.73
Transform	0.438	0.443	6/429/1036	38.47/53.43/68.65	0.371	0.374	29/465/748	32.54/44.83/58.04	0.331	0.334	29/435/842	27.76/42.87/60.85
BERT-Sim	0.399	0.401	44/839/1407	36.95/44.75/54.32	0.364	0.370	41/431/695	31.71/44.28/56.18	0.310	0.313	56/522/902	26.32/39.05/54.56
BERT-Cls	0.265	0.275	15/**237**/**640**	16.75/45.37/71.77	0.213	0.220	24/318/646	12.47/40.53/64.67	0.204	0.208	25/253/568	11.50/38.27/66.73
Ours	**0.456**	**0.462**	**4**/254/685	**39.62/56.21/73.26**	**0.413**	**0.419**	**7/97/186**	**34.64/53.29/75.91**	**0.409**	**0.411**	**9/196/419**	**35.22/51.47/70.82**

Table 6: Quote recommendation results of different models on the three parts of QuoteR, **given the left context only**. Recall@1/10/10 is percentage. The **boldfaced** results exhibit statistically significant improvement over the other results with $p<0.1$ given by paired t-tests.

Rank	Quote	Score
1	**人不能两次踏进同一条河流** **No man ever steps in the same river twice**	0.995
2	世界上唯一不变的就是变化 The only constant in life is change	0.002
3	萧瑟秋风今又是，换了人间 The autumn wind still sighs, but the world has changed	0.001
4	前途是光明的，道路是曲折的 The road is tortuous, but the future is bright	0.001
5	只有变化是永恒的 Change is the only constant	0.001

Table 7: A standard Chinese quote recommendation case. Top 5 recommended quotes (the gold quote is in boldface) are listed for the context: 从盘面上看，股票价格会呈现某种带漂移的无规则行走，涨跌无常，难以捉摸。[Quote]，这话放在投资领域也同样受用。事物是在不断变化的，历史数据只能起一定程度的参考作用。投资者想凭借历史数据准确预测未来几乎是不可能的。(*The stock price shows some kind of irregular walk with drift, up and down unpredictably. The saying that [Quote] is also applicable to investment. Things are constantly changing, and historical data have limited reference value. It is almost impossible for investors to accurately predict the future based on historical data.*)

Rank	Quote	Score
1	**Truth is always strange**	0.984
2	Truth is always stranger than fiction	0.005
3	Truth is dangerous	0.002
4	Truth is subjectivity	0.001
5	Fact is stranger than fiction	0.001

Table 8: An English quote recommendation case. Top 5 recommended quotes (the gold quote is in boldface) are listed for the context: *We've talked about some of the prophecies that have already come true in our cities from science fiction. What are some prophecies that have yet to come true? In a way, while sci-fi is fascinating, [Quote]. The transformation around surveillance is already mimicking a lot of the predictions in, say, minority report, which was very much emphasizing how surveillance and marketing were becoming completely tailored to the individual.*

Rank	Quote	Score
1	**道不同，不相为谋** **Persons walking different paths cannot work together**	0.412
2	话不投机半句多 One word is too much for someone uncongenial	0.270
3	惺惺惜惺惺 The wise appreciate one another	0.111
4	白头如新，倾盖如故 You may know a little about old acquaintances and make close friends with a stranger soon	0.033
5	近朱者赤，近墨者黑 One takes the behavior of one's company	0.024

Table 9: A classical Chinese quote recommendation case. Top 5 recommended quotes (the gold quote is in boldface) are listed for the context: 我是少数群体中的一员，谈不上饱受社会不文明的欺压，却也受到主流文化对边缘群体的排斥。你不认可我，所谓[Quote]，我哪里还能跟你热情。相反，认可我的人就会享受我的回应，真诚也好，善良也好，温柔也好，我会把我好的一面展示给他们。(*I am a member of a minority group, not suffering from the oppression of uncivilized people in society, but being ostracized by mainstream culture. If you don't approve of me, as the saying goes, [Quote], I can't be enthusiastic about you. In contrast, the persons who approve of me will enjoy my good side, including sincerity, goodness and gentleness.*)

be found in the README file of the Software in the supplementary materials.

In addition, our models for English, standard Chinese and classical Chinese have about 308M, 308M and 329M parameters, respectively. And the average training time is 7.5h, 26h and 29h, respectively.