Learning to Recommend Quotes for Writing

Jiwei Tan, Xiaojun Wan* and Jianguo Xiao
Institute of Computer Science and Technology, Peking University, Beijing 100871, China
The MOE Key Laboratory of Computational Linguistics, Peking University, Beijing 100871, China
{tanjiwei, wanxiaojun, xiaojianguo}@pku.edu.cn

Abstract

In this paper, we propose and address a novel task of recommending quotes for writing. Quote is short for quotation, which is the repetition of someone else's statement or thoughts. It is a common case in our writing when we would like to cite someone's statement, like a proverb or a statement by some famous people, to make our composition more elegant or convincing. However, sometimes we are so eager to make a citation of quote somewhere, but have no idea about the relevant quote to express our idea. Because knowing or remembering so many quotes is not easy, it is exciting to have a system to recommend relevant quotes for us while writing. In this paper we tackle this appealing AI task, and build up a learning framework for quote recommendation. We collect abundant quotes from the Internet, and mine real contexts containing these quotes from large amount of electronic books, to build up a dataset for experiments. We explore the particular features of this task, and propose a few useful features to model the characteristics of quotes and the relevance of quotes to contexts. We apply a supervised learning to rank model to integrate multiple features. Experiment results show that, our proposed approach is appropriate for this task and it outperforms other recommendation methods.

Introduction

When we are writing articles, it is a common case that we want to cite someone else's statement, like a proverb or a saying by some famous people. The cited content is called *quotation*, or *quote* for short, meaning the repetition of someone else's statement or thoughts. As citing this kind of famous sayings can make our composition more elegant or convincing, the occurrence of quote is common in writing language. Figure 1 is a snippet in (Smiles 1875) showing the quote usage in the writing. The famous saying of Benjamin Franklin, *diligence is the mother of good luck*, is cited in the article to prove the importance of hard work. However, making an elegant quote in writing is not an easy job. Because knowing or remembering so many quotes is not easy, many times we are so eager to make a citation of quote somewhere to support our viewpoint, but have no idea about the relevant quote to express our idea. Even though we have the help of search engine today, it is hard to find the keywords to search if we do not have an impression of the quotes we need. As shown in Figure 1, the real idea the author wants to express is the sentence following the quote. We can see that the author's expression and the quote have different word usages, for example, "effort" versus "diligence", and "success" versus "good luck". Can we build an AI system to recommend relevant quotes for us according to the contexts while we are writing? In this paper we aim to tackle this appealing AI task. There are several websites in the Internet that specially collect and provide quotes for users. However, as far as we know, there is no attempt to recommend relevant quotes for people while writing.

> There has long been a popular belief in "good luck;" but, like many other popular notions, it is gradually giving way. The conviction is extending that ***diligence is the mother of good luck***; in other words, that a man's success in life will be proportionate to his efforts, to his industry, to his attention to small things.

Figure 1: A snippet of quote usage in literature.

In this paper, we propose and define the quote recommendation task. Given a snippet of context, a quote recommendation system should recommend quotes which are relevant with the context. The word "relevant" means semantic relatedness or expressing similar idea. The task is very challenging because contexts and quotes are usually short, and it is less likely that they share some keywords. Moreover, the meaning of words used in quotes may be different from common cases, by means of metaphor and so on. In the example of Figure 1, besides the different words used, *the mother of good luck* has a latent meaning of "the source of good luck", which makes the task more interesting and challenging.

We explore the particular features of the quote recommendation task, and propose to leverage a learning to rank framework to address this task. We investigate three groups of features for the task: quote-based features, relevance-based features and context-based features. Quote-based features reflect the particular features of quote data; relevance-based features model the relevance between quotes and contexts

*Xiaojun Wan is the corresponding author.

directly, and context-based features are second-order relevance features. Experiment results on a large real dataset show that our proposed approach is very effective for the quote recommendation task, and it outperforms a few strong baselines. The usefulness of each group of features is also validated.

The main contributions of this paper are three-fold. First, we propose a novel quote recommendation task and discuss the particular features of the task. We also construct a large real dataset for the task. Second, we propose to leverage a learning to rank framework to address the task, and develop a few useful features for the task. Third, we conduct a series of experiments on a large real dataset and the evaluation results verify the efficacy of our proposed approach and the usefulness of the features. We show that quote recommendation for writing is a meaningful task and is potentially useful in practice.

Related Work

Quote recommendation can be viewed as a content-based recommendation task, and the most closely related work is content-based citation recommendation for scientific writing. Because both tasks are based on text content for recommendation, some citation recommendation methods can be applied to quote recommendation. Shaparenko and Joachims (2009) address the relevance of citation context and the paper content, and apply language model to the recommendation task. Lu et al. (2011) propose to use translation model to bridge the gap between two heterogeneous languages. He et al. (2010) propose a context-aware approach which measures the context-based relevance between a citation context and a document. Tang, Wan, and Zhang (2014) propose to learn a low-dimensional joint embedding space for both contexts and citations.

Supervised learning for citation recommendation is a relatively less explored research area. These methods explore multiple features to model the relevance between citation and context, and try to combine the features with a supervised learning method. Bethard and Jurafsky (2010) present the early approach, which utilizes a linear classifier to learn a recommendation model. Rokach et al. (2013) present a supervised learning framework for citation recommendation. They use classifiers to utilize multiple features of three types. To the best of our knowledge, no prior studies have employed learning to rank algorithms for citation recommendation yet.

Other related works include news recommendation (Lv et al. 2011; Phelan, McCarthy, and Smyth 2009), and other kinds of content-based recommendations (Chen et al. 2007; Kardan and Ebrahimi 2013; Lops et al. 2013). Moreover, for the quote itself there are also some works which study the origin, spread and theory of quote (Anton 2009; Finnegan 2011; Leskovec, Backstrom, and Kleinberg 2009; Morson 2006).

Problem Definition

We introduce several definitions as follows.

Definition 1 (*Quote Recommendation*). *Given a snippet of context, the task aims to return a ranked list of quotes which are relevant to the context and may be cited in the context.*

Definition 2 (*Query Context*). *Query context is the context provided by the user while writing. Quotes are recommended according to the* ***query context****. In our work we treat a window of characters before and after a quote as the context of the quote. All query contexts make up the* ***query context set****.*

Definition 3 (*Candidate Quote*). *Any quote which may be fit for a given query context is a candidate quote for the query context. All candidate quotes make up the* ***candidate quote set****.*

Definition 4 (*Gold Quote*). *A gold quote for a given query context is the correct quote for the query context. In our work gold quote is the quote actually appearing with the query context in the real article.*

Definition 5 (*Matching Function*). *The matching function F estimates the matching score of a candidate quote q to a query context s. In the recommendation system candidate quotes are ranked according to the matching scores.*

The framework of quote recommendation is like typical content-based recommendation tasks. We first retrieve a candidate quote set from the whole quote data, and then rank the candidate quotes according to matching function scores. Therefore, the quote recommendation framework consists of two stages: constructing candidate quote set and learning the matching function to rank the candidate quotes.

Data Construction

Dataset

Since the task addressed in this paper is a new task, we constructed the evaluation dataset by ourselves. We collected quotes from the website of Library of Quotes[1]. The quotes are categorized according to their tags, like *love*, *life*, *friendship* and so on. The quotes also have the information of authors and the votes by the users. In our experiments we collected about 380,000 quotes.

In order to get real contexts of the quotes, we collected about 20GB raw texts of e-books from Project Gutenberg[2] (Hart 1971) as corpus. Then we searched the quotes in the corpus and got the occurrences of quotes in real literatures with their contexts. In the experiments we took the concatenation of around 300 characters[3] before and after each quote as the context of a quote. After filtering quotes that appear less than 5 times, there are 3,158 unique quotes, and 64,323 context-quote pairs in the dataset. We made these quotes and contexts as our whole dataset in the experiments. The 64,323 context-quote pairs were randomly split, according to the proportion of 8:1:1, as training set, validation set and test set, respectively. The statistics of dataset are listed in Table 1. In Table 1, "C-Q pairs" stands for "context-quote pairs".

[1]http://www.libraryofquotes.com

[2]http://www.gutenberg.org

[3]Several characters more or less to avoid truncation of a word.

Table 1: Statistics of experiment dataset.

Dataset	C-Q pairs	Quotes	Authors	Categories
Training	51457	3061	748	814
Validation	6433	3090	754	816
Test	6433	3102	760	820
All	64323	3158	762	822

We preprocessed all quotes and contexts, by removing stopwords and stemming words with the Porter stemmer[4]. There are 15008 unique words in the dataset after preprocessing.

Selecting Candidate Quotes

For efficiency, recommendation systems usually quickly construct a candidate set from the large whole dataset in practice. In this paper we construct the candidate set by using the bag-of-words similarity. However, retrieving candidate quotes simply by the bag-of-words similarity between quotes and query context performs badly. We make an improvement to the candidate selection as follows.

Given a query context, we decide the candidate quotes by calculating the similarity of the query context with the content-enriched quotes. A quote is enriched by concatenating the quote with its contexts in the training set. This is under the intuition that contexts are semantically related to the quote, so they can to some degree express the idea of a quote more explicitly. Note that when the query context is in the context set of a quote (which means this quote is the gold quote of the query context, and this information actually should not be known when the query context is not from the training set), the query context itself should not be concatenated. In our experiments we select the 1000 quotes with largest similarities to the query context as candidate quotes. Experiments showed that about 92.6% of gold quotes are in the candidate set by selecting in this way. Then, both our approach and baseline methods work on the same candidate quote set.

Our Approach

In this section we introduce defining the matching function with a learning to rank model. We first introduce the framework of learning to rank and the learning algorithm, and then we introduce the features used for learning.

Learning to Rank Framework

After constructing the candidate quote set, we propose to apply the learning to rank technique to learn the matching function. A learning to rank model is first learned in the training process. Then the learned model is used to predict rank values for test examples. The predicted rank value is treated as matching score for every test context-quote pair.

Given the training dataset of context–gold quote pairs, we construct the data for the learning to rank model as follows. Every query context corresponds to a query for learning to rank. Supposing for query context s, the candidate set is Q, and for every candidate quote $q \in Q$, the context-quote pair s-q corresponds to an example of current query. Every example should be assigned with its target value and feature values. There are only two rank levels in the task: relevant (gold) quote and irrelevant quote to the query context. If a context-quote pair is a correct match, we call the context-quote pair a positive example and assign a target value of 2; otherwise the pair is called a negative example and assigned with a target value of 1. The feature values of the examples will be introduced later.

Note that the number of negative examples is much larger than that of positive examples. If we use a classification model, there will be a serious class imbalance problem. For learning to rank models there is no need to worry about this. However, for better computational efficiency, we also randomly sample 4 negative examples for the training data, like in (Rokach et al. 2013). Our experiments showed that a sample of 4 negative examples reaches similar results with using all negative examples.

Learning to Rank Algorithm

To date, a variety of learning to rank algorithms have been proposed, including pointwise, pairwise and listwise algorithms. In particular, listwise methods minimize a direct loss (an appropriate evaluation measure) between the true ranks of the list and the estimated ranks of the list (Agarwal et al. 2012). It has been shown that listwise methods outperform pointwise and pairwise methods like SVMRank (Joachims 2002) for information retrieval tasks (Quoc and Le 2007). In this paper we leverage a listwise method—the linear feature-based model (Metzler and Croft 2007)—for learning to rank.

Linear feature-based models focus on approaches that directly maximize the evaluation metric by optimizing the objective functions. Therefore the linear feature-based models usually lead to better results. The goal of linear feature-based models is to find a parameter setting Λ that maximizes the evaluation metric E over the parameter space. It can be formally stated as:

$$\begin{aligned} \widehat{\Lambda} &= \arg\max_{\Lambda} E\left(R_\Lambda; T\right) \\ s.t. \quad & R_\Lambda \sim F_\Lambda\left(D; Q\right) \\ & \Lambda \in M_\Lambda \end{aligned}$$

where $R_\Lambda \sim F_\Lambda(D;Q)$ denotes that the orderings in R_Λ are induced using matching function F. D is the set of candidates to be ranked and Q is a query. T denotes the training data. M_Λ is the parameter space over Λ.

We apply the coordinate ascent algorithm to optimize the linear feature-based model. Coordinate ascent is a commonly used optimization technique for unconstrained optimization problems. In our experiments we use a learning to rank tool called RankLib[5] which implements the coordinate ascent algorithm for learning to rank. The RankLib tool supports an input of validation file and can automatically find the best parameters on the validation set. The parameters of RankLib we use are "-norm zscore" and "–metric2t NDCG@5", which mean normalizing each feature by

[4]http://tartarus.org/martin/PorterStemmer/

[5]http://people.cs.umass.edu/˜vdang/ranklib.html

its standard deviation and optimizing the ranking measurement of NDCG@5, because NDCG@5 is one of the evaluation metrics used in our experiments.

Features

We explore the particular features of the quote recommendation task, and investigate three groups of features as follows. There are a total of 16 features used in our framework, each feature corresponds to a value normalized in [-1,1].

Quote-Based Features Similar to IR and web search tasks, there may be higher probability for popular quotes to be cited. Therefore, we propose the following 6 features to measure the popularity of each quote itself.
Frequency: Number of contexts in the training set where the quote is cited.
Vote: Number of votes by users who like the quote, obtained from the Library of Quotes website.
Author-Quotes: Number of quotes that the quote's author has in the training set.
Author-Popularity: Number of contexts citing the quotes of current quote's author in the training set.
Web-Popularity: We search the quote by Bing[6] and treat the number of retrieved results as the web popularity of the quote.
Quote-Rank: We apply the PageRank (Brin and Page 1998; Page et al. 1999) algorithm on the candidate quote set, and treat the rank value as one feature to measure the importance of quotes. We build a graph based on the quotes. The vertices in the graph are all the candidate quotes, and the weight associated with each arc is the cosine similarity between the quotes. A weighted PageRank algorithm is applied on the graph.

Relevance-Based Features The most intuitive feature is to measure the relevance of quotes to the contexts by the similarity between them. In order to better capture the relevance of quotes to the contexts, we use several kinds of similarity metrics. All the similarities are calculated by the cosine angles of vectors representing the texts, as:

$$sim\left(v_1, v_2\right) = \frac{v_1 \cdot v_2}{\sqrt{v_1 \cdot v_1}\sqrt{v_2 \cdot v_2}}$$

where v_1 and v_2 are the vector representations of texts. As a quote is typically one or several sentences, and the vocabulary used between quotes and contexts is different, the bag-of-words similarity usually cannot measure the relevance well. In order to address the vocabulary mismatch problem, we also try to model quotes and contexts into semantic representations. We introduce 6 text similarity metrics in total based on 6 kinds of vector representations as follows.
Bag-of-Words: The most popular bag-of-words text representation is used. A text is represented by the TF-IDF word vector. The dimension of vectors is 15008.
Smoothed Bag-of-Words: As a quote is usually short, its bag-of-words similarity to the context is likely to be zero. Because the quotes of same category may express similar idea, we use the concatenation of all quotes that have same category tag as the smoothed representation of a quote.
LSA: The latent semantic analysis (LSA) (Landauer, Foltz, and Laham 1998) is used to represent the text vector in the latent semantic space. The dimension of latent semantic vectors is set to 1000.
LDA: Quotes and contexts are modeled into the distribution of topics with latent Dirichlet allocation (LDA) (Blei, Ng, and Jordan 2003; Griffiths and Tenenbaum 2004). The number of topics is set to 1000.
ESA: Quotes and contexts are modeled into concept vectors with Explicit Semantic Analysis (ESA) (Gabrilovich and Markovitch 2006) on the corpus of Wikipedia. The dimension of explicit semantic vectors is 202037.
Word2Vec: In order to bridge the vocabulary gap, a word-to-vector model (Mikolov et al. 2013) is trained on the whole dataset to learn vectors for all words. Then a text is represented by averaging the vectors of words it contains, as similar to (Huang et al. 2012; Tang et al. 2014). The dimension of word vectors is set to 500.

Context-Based Features The known contexts of quotes in the training data are useful information for the quote recommendation task. In a word, contexts of the same quote are likely to be semantically related. Context-based features take use of the awareness of the context-quote relatedness in the training set. The relevance of a query context to a candidate quote is measured by the similarity of the query context with the known contexts of the candidate quote. For a query context s and a candidate quote q, suppose C_q is the set of contexts in the training set whose gold quote is q. If s itself is in C_q, s should be removed out of C_q. Then we define 4 context-based features as follows:
Max-Similarity: $f_{cmax}\left(s, q\right) = \max_{c \in C_q} sim\left(s, c\right)$
Min-Similarity: $f_{cmin}\left(s, q\right) = \min_{c \in C_q} sim\left(s, c\right)$
Avg-Similarity: $f_{cavg}\left(s, q\right) = \text{avg}_{c \in C_q} sim\left(s, c\right)$
Square-Similarity: $f_{csqr}\left(s, q\right) = \frac{1}{|C_q|} \sum_{c \in C_q} sim\left(s, c\right)^2$

When measuring context-to-context similarity $sim\left(s, c\right)$ we use the bag-of-words similarity.

Experiments

Evaluation Metrics

Recall@k: For recommendation systems, users usually will only view several top ranked candidates. Therefore whether the gold quote exists in the top ranked candidates is important, and Recall@k is designed for measuring this. Suppose C_{test} is the set of test query contexts, and G is the set of gold quotes which are recommended in the top k results, then the total recall is:

$$\text{Recall@}k = |G| / |C_{test}|$$

MRR: Recall@k considers whether gold quote is in the top k results, but it does not consider the exact ranking position of gold quote. MRR (Voorhees and others 1999) is typically used to evaluate the position of the first correct candidate, and in our task MRR is a good metric to evaluate the system performance. Let $rank\left(s\right)$ represent the ranking position of the gold quote of query context s, the MRR score is com-

[6]http://www.bing.com

puted as:

$$\text{MRR} = \frac{1}{|C_{test}|} \sum_{s \in C_{test}} \frac{1}{rank\,(s)}$$

NDCG@k: Normalized discounted cumulative gain (NDCG) is widely used in recommendation systems. It also considers the ranking position of the gold candidate, and desires highly relevant candidates to appear earlier in the top k list. The NDCG@k for a query context is computed as:

$$\text{NDCG@}k = Z_k \sum_{i=1}^{k} \frac{2^{r(i)} - 1}{\log\,(1+i)}$$

where $r\,(i)$ is the rating of the i-th candidate quote in the ranked list, and Z_k is a normalization constant to make the perfect ranked list get a NDCG score of 1. In our experiment $r\,(i) = 1$ if the i-th quote is the gold quote, and otherwise $r\,(i) = 0$. The overall NDCG score is the average NDCG score of all test query contexts.

Baselines

We compare our approach with two simple baselines and several state-of-the-art content-based recommendation methods, which have shown good performances for the task of citation recommendation. The methods are: random baseline, cosine similarity, context-aware relevance model (CRM) (He et al. 2010), translation model (TM) (Lu et al. 2011) and bilingual context-citation embedding model (BLSRec) (Tang, Wan, and Zhang 2014). We implement these methods and set parameters in the same way with that in the respective papers. The Recall at different ks, MRR and NDCG@5 are compared. For the MRR and NDCG scores of every method, we conduct two-tailed t-test to see whether our proposed approach is significantly better than others. We introduce these methods as follows:

Random: It ranks the candidate quotes randomly.

Cosine Similarity: It ranks the candidate quotes according to their bag-of-words cosine similarities to the query context.

CRM: It recommends quotes according to the similarities between query context and contexts of each candidate quote in the training set.

TM: As we believe there is a gap between the vocabulary of quotes and contexts, we adapt the translation model for citation recommendation by treating quotes and contexts as different "languages".

BLSRec: The BLSRec model tries to bridge the gap of different languages by mapping different languages into a learned low-dimensional joint embedding space. Because BLSRec can take use of the training information through a supervised learning procedure, it proves to achieve good performance in cross-language recommendation tasks. We adapt the BLSRec-I model to our quote recommendation task based on the code provided by the author, by treating quotes and contexts as different "languages".

Results and Discussion

Methods Comparison The comparison results of different methods are given in Table 2 and Figure 2. As we can see from the results, our learning to rank model achieves the best results, and it is significantly better than other methods. Recommending according to the similarity between quotes and contexts does not perform well, proving the vocabulary gap between quotes and contexts. The translation model performs better, showing that bridging the vocabulary gap is effective to this task. The BLSRec model performs much better than TM, proving that taking better use of the supervised information helps better bridging the vocabulary gap and learning better semantic representations. The CRM model performs even better than BLSRec, showing that context information is quite useful in the quote recommendation task. As the learning to rank model can integrate multiple useful features, it achieves the best performance.

Table 2: Comparison results of different methods.

Method	MRR	NDCG@5
Random	0.006	0.002
Cosine Similarity	0.106	0.105
CRM	0.324	0.330
TM	0.197	0.203
BLSRec	0.272	0.282
Our Approach	**0.360**	**0.372**

(Two-tailed t-tests show that the differences between Our Approach and other methods are all statistically significant with $p \ll 0.001$)

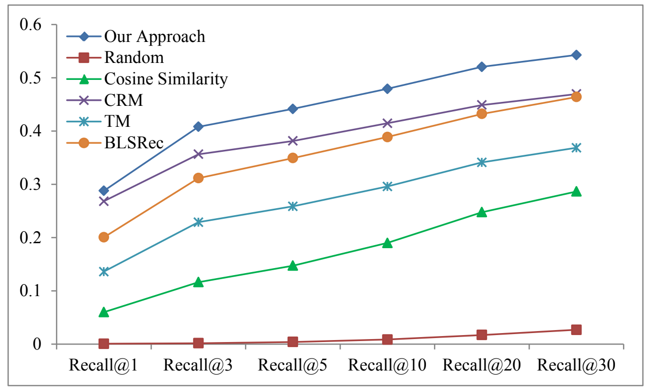

Figure 2: Results of Recall@k of different methods.

Feature Validation In order to validate the usefulness of different groups of features, we show comparison results of different feature combinations in Table 3 and Figure 3. We conduct experiments by removing each group of features. In the table, "w/o" means removing the corresponding group of features.

As we can see from the results, all the three groups of features contribute to the learning to rank model. Context-based features affect the results most significantly, because they can supplement more latent semantic information to candidate quotes. The effect of relevance-based features is relatively less significant, though we use different semantic representations, proving that modeling the similarity between contexts and quotes is a hard job.

Table 3: Results of feature validation.

Feature	MRR	NDCG@5
w/o Quote-Based	0.337	0.348
w/o Relevance-Based	0.352	0.361
w/o Context-Based	0.097	0.095
All Features	**0.360**	**0.372**

(Two-tailed t-tests show that the differences between All Features and others are all statistically significant with $p \ll 0.001$)

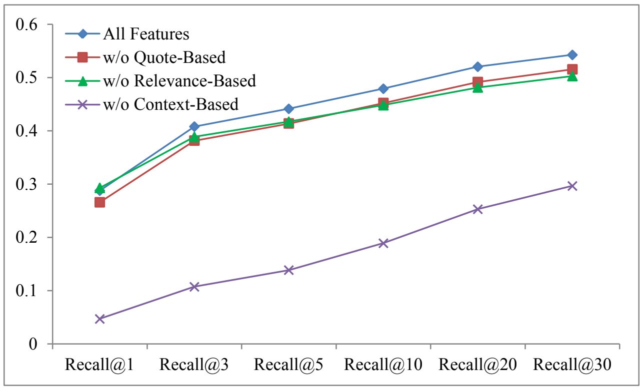

Figure 3: Results of Recall@k of feature validation.

Comparing with Other Learning Models To prove the effectiveness of the learning to rank model, we compare it with the classification model. In addition, we also compare the linear feature-based model (denoted by Coordinate Ascent) with SVMRank. For the classification model, positive examples are labeled +1 and negative examples are labeled -1. All the features are the same. Results are shown in Table 4 and Figure 4.

As can be seen from the results, the Coordinate Ascent learning to rank model reaches better results than the SVMClassify and SVMRank. The results prove that the listwise learning to rank method, which optimizes directly on the evaluation metric, performs better than a pairwise learning to rank method or a classification method. Results of SVMRank are slightly better than results of SVMClassify, because it better models the ranking task and does not suffer from the class imbalance problem.

System Demonstration Figure 5 shows the recommendation results for a context in (Wilson 1879) in the test set. Given the context in the left, we can know that the author wants to express the idea of "diligence leads to good luck". According to the context, our system recommends a list of

Table 4: Results of different learning models.

Model	MRR	NDCG@5
Coordinate Ascent	**0.360**	**0.372**
SVMRank	0.348	0.358
SVMClassify	0.345	0.357

(Two-tailed t-tests show that the differences between Coordinate Ascent and others are all statistically significant with $p \ll 0.001$)

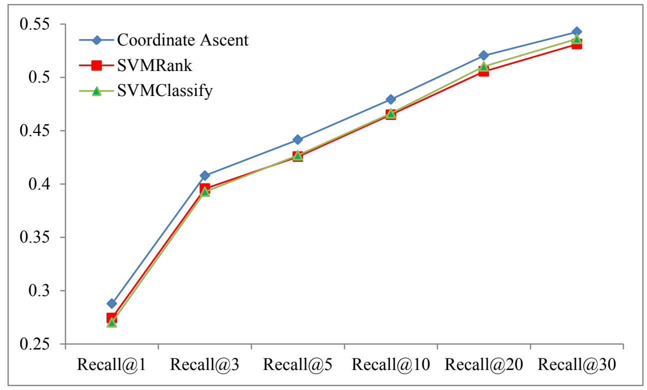

Figure 4: Results of Recall@k of different learning models.

and what's to be done now, I'm sure I don't know. Some people's born to luck; 'tain't my case." "Nay, Jim," cried the other, "you're out there: there's no such thing as luck, and no one's born to good luck. But there's an old proverb which comes pretty near the truth, and it's this, [***quote***] I don't believe in luck or chance myself, but I believe in diligence, with god's blessing. It says in the bible, 'the hand of the diligent maketh rich."' "Well, and I have been diligent," exclaimed Jim: "I've never been away from my work a day scarcely. But see what a lot of children I've got	**[1] Diligence is the mother of good luck.** [2] There is truth in wine. [3] Good Night. And, Good Luck. [4] It was good luck, [5] No use crying over spilt milk. [6] Diligence is the mother of good fortune. [7] Little strokes fell great oaks. [8] All work and no play makes Jack a dull boy. [9] There was a star danced, and under that was I born. [10] Lost time is never found again.

Figure 5: A recommendation example in the test set.

quote candidates in the right. As we can see, the first candidate is indeed the correct one. Moreover, candidate [6] is the paraphrase of [1] and is also recommended in the list. It can be concluded that the quote recommendation system is meaningful and the approach we propose is useful.

Conclusion and Future Work

In this paper we present a novel task of recommending quotes for citing in the writing. We investigate the particular features of this task and propose several useful features. We apply a learning to rank model to integrate multiple features and show that the learning to rank model is appropriate for this task. We show that quote recommendation for writing is a meaningful task and is potentially useful in practice. In the future, we will explore specifically on better tackling the vocabulary gap of quotes and contexts. We will also investigate using deep learning techniques to improve the recommendation performance.

Acknowledgments

This work was supported by National Hi-Tech Research and Development Program (863 Program) of China (2014AA015102, 2012AA011101), National Natural Science Foundation of China (61170166, 61331011) and Beijing Nova Program (2008B03).

References

Agarwal, A.; Raghavan, H.; Subbian, K.; Melville, P.; Lawrence, R. D.; Gondek, D. C.; and Fan, J. 2012. Learning to rank for robust question answering. In *Proceedings of the 21st ACM international conference on Information and knowledge management*, 833–842. ACM.

Anton, C. 2009. Words to live by: Scholarly quotations as proverbial theory. *ETC.: A Review of General Semantics* 66(2):167.

Bethard, S., and Jurafsky, D. 2010. Who should i cite: learning literature search models from citation behavior. In *Proceedings of the 19th ACM international conference on Information and knowledge management*, 609–618. ACM.

Blei, D. M.; Ng, A. Y.; and Jordan, M. I. 2003. Latent dirichlet allocation. *the Journal of machine Learning research* 3:993–1022.

Brin, S., and Page, L. 1998. The anatomy of a large-scale hypertextual web search engine. *Computer networks and ISDN systems* 30(1):107–117.

Chen, T.; Han, W.-L.; Wang, H.-D.; Zhou, Y.-x.; Xu, B.; and Zang, B.-Y. 2007. Content recommendation system based on private dynamic user profile. In *Machine Learning and Cybernetics, 2007 International Conference on*, volume 4, 2112–2118. IEEE.

Finnegan, R. 2011. *Why do we quote?: the culture and history of quotation*. Open Book Publishers.

Gabrilovich, E., and Markovitch, S. 2006. Overcoming the brittleness bottleneck using wikipedia: Enhancing text categorization with encyclopedic knowledge. In *AAAI*, volume 6, 1301–1306.

Griffiths, D., and Tenenbaum, M. 2004. Hierarchical topic models and the nested chinese restaurant process. *Advances in neural information processing systems* 16:17.

Hart, M. 1971. *Project gutenberg*. Project Gutenberg.

He, Q.; Pei, J.; Kifer, D.; Mitra, P.; and Giles, L. 2010. Context-aware citation recommendation. In *Proceedings of the 19th international conference on World wide web*, 421–430. ACM.

Huang, E. H.; Socher, R.; Manning, C. D.; and Ng, A. Y. 2012. Improving word representations via global context and multiple word prototypes. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1*, 873–882. Association for Computational Linguistics.

Joachims, T. 2002. Optimizing search engines using clickthrough data. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, 133–142. ACM.

Kardan, A. A., and Ebrahimi, M. 2013. A novel approach to hybrid recommendation systems based on association rules mining for content recommendation in asynchronous discussion groups. *Information Sciences* 219:93–110.

Landauer, T. K.; Foltz, P. W.; and Laham, D. 1998. An introduction to latent semantic analysis. *Discourse processes* 25(2-3):259–284.

Leskovec, J.; Backstrom, L.; and Kleinberg, J. 2009. Meme-tracking and the dynamics of the news cycle. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, 497–506. ACM.

Lops, P.; De Gemmis, M.; Semeraro, G.; Musto, C.; and Narducci, F. 2013. Content-based and collaborative techniques for tag recommendation: an empirical evaluation. *Journal of Intelligent Information Systems* 40(1):41–61.

Lu, Y.; He, J.; Shan, D.; and Yan, H. 2011. Recommending citations with translation model. In *Proceedings of the 20th ACM international conference on Information and knowledge management*, 2017–2020. ACM.

Lv, Y.; Moon, T.; Kolari, P.; Zheng, Z.; Wang, X.; and Chang, Y. 2011. Learning to model relatedness for news recommendation. In *Proceedings of the 20th international conference on World wide web*, 57–66. ACM.

Metzler, D., and Croft, W. B. 2007. Linear feature-based models for information retrieval. *Information Retrieval* 10(3):257–274.

Mikolov, T.; Sutskever, I.; Chen, K.; Corrado, G. S.; and Dean, J. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems*, 3111–3119.

Morson, G. S. 2006. Bakhtin, the genres of quotation, and the aphoristic consciousness. *The Slavic and East European Journal* 213–227.

Page, L.; Brin, S.; Motwani, R.; and Winograd, T. 1999. The pagerank citation ranking: Bringing order to the web.

Phelan, O.; McCarthy, K.; and Smyth, B. 2009. Using twitter to recommend real-time topical news. In *Proceedings of the third ACM conference on Recommender systems*, 385–388. ACM.

Quoc, C., and Le, V. 2007. Learning to rank with nonsmooth cost functions. *NIPS07* 19:193.

Rokach, L.; Mitra, P.; Kataria, S.; Huang, W.; and Giles, L. 2013. A supervised learning method for context-aware citation recommendation in a large corpus. *INVITED SPEAKER: Analyzing the Performance of Top-K Retrieval Algorithms* 1978.

Shaparenko, B., and Joachims, T. 2009. Identifying the original contribution of a document via language modeling. In *Machine Learning and Knowledge Discovery in Databases*. Springer. 350–365.

Smiles, S. 1875. *Thrift*. J. Murray.

Tang, D.; Wei, F.; Qin, B.; Zhou, M.; and Liu, T. 2014. Building large-scale twitter-specific sentiment lexicon: A representation learning approach. In *COLING*, 172–182.

Tang, X.; Wan, X.; and Zhang, X. 2014. Cross-language context-aware citation recommendation in scientific articles. In *Proceedings of the 37th international ACM SIGIR conference on Research & development in information retrieval*, 817–826. ACM.

Voorhees, E. M., et al. 1999. The trec-8 question answering track report. In *TREC*, volume 99, 77–82.

Wilson, T. P. 1879. *True to His colours; or, The life that wears best*. Thomas Nelson and Sons.