Enriching Article Recommendation with Phrase Awareness

Chia-Wei Chen
Academia Sinica
Taipei, Taiwan
ss87021456@iis.sinica.edu.tw

Sheng-Chuan Chou
Academia Sinica
Taipei, Taiwan
angelocsc@iis.sinica.edu.tw

Lun-Wei Ku
Academia Sinica
Taipei, Taiwan
lwku@iis.sinica.edu.tw

ABSTRACT

Recent deep learning methods for recommendation systems are highly sophisticated. For article recommendation task, a neural network encoder which generates a latent representation of the article content would prove useful. However, using raw text with embedding for models could degrade sentence meanings and deteriorate performance. In this paper, we propose PhrecSys (Phrase-based Recommendation System), which injects phrase-level features into content-based recommendation systems to enhance feature informativeness and model interpretability. Experiments conducted on six months of real-world data demonstrate that phrase features boost content-based models in predicting both user click and view behavior. Furthermore, the attention mechanism illustrates that phrase awareness benefits the learning of textual focus by putting the model's attention on meaningful text spans, which leads to interpretable article recommendation.

CCS CONCEPTS

• **Information systems → Recommender systems**; **Document representation**; • **Computing methodologies → Natural language processing**; **Information extraction**;

KEYWORDS

article recommendation, content-based recommendation, phrase mining

1 INTRODUCTION

Learning from humans has shown to be effective for designing better AI systems [9]. Intuitively, humans cannot memorize lengthy article information, which serves as a clue for article recommendation systems. That is, after reading through an article, there is a high possibility that only a few parts of the article will attract user attention. However, deep learning models that adopt raw text with embedding might degrade the meaning of sentences, which violates this precept.

The phrase, a small group of words standing together as a conceptual unit, can serve as a useful component of these attractive parts, or chunks. An example of the phrase is shown in Figure 1. Strong data-driven methods for extracting quality phrases have been developed recently [16, 22]. With these, high-quality phrases can be extracted automatically without additional human effort. However, despite the regular utilization of handcrafted keywords, article recommendation seldom leverages phrase mining. Therefore, we propose PhrecSys, which utilizes phrase-level features in content-based recommendation systems. We add phrase mining to state-of-the-art content-based recommendation models and compare their performance with the original models. Moreover, we use the attention mechanism [14] and visualize the changes in attention weights during training for a clear view of the model focus.

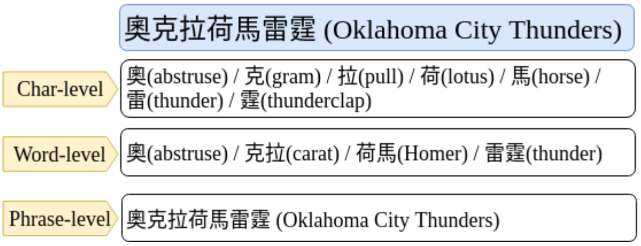

奧克拉荷馬雷霆 (Oklahoma City Thunders)
Char-level
奧(abstruse) / 克(gram) / 拉(pull) / 荷(lotus) / 馬(horse) / 雷(thunder) / 霆(thunderclap)
Word-level
奧(abstruse) / 克拉(carat) / 荷馬(Homer) / 雷霆(thunder)
Phrase-level
奧克拉荷馬雷霆 (Oklahoma City Thunders)

Figure 1: Different level feature comparison example

This work includes two contributions. First, to the best of our knowledge, we are the first to incorporate phrase mining techniques to boost content-based article recommendation systems. Second, we visualize the change of attention weights during training to show systems when given phrase features: systems learn to read in a more focused way. As a great output, these focused chunks also help us interpret how these systems are producing recommendations.

2 RELATED WORK

As the recommended targets are news articles, and the system is attempting to recommend articles based on the article currently being read, content-based recommendation models are the most related. Content-based models inherently sidestep both the cold-start [21] issue and the problem of proposing unrelated content, as the content is the primary information needed to make good recommendations. This is especially useful for article recommendation in environments where new articles arrive regularly. Conventionally, articles are treated as bags of words, whose similarity is calculated by scoring functions such as TF-IDF [18] and BM25 [20]. Recent advances in textual feature extraction with NN models [5, 13] make it possible to generate article representations using NN encoders [3]. With the help of word embedding, it is now possible to aggregate various neural network architectures in the textural features for different tasks. Information retrieval models can also be used for recommendation applications [1, 4, 23], additionally incorporating user feedback [17] and past behavior [6] to generate personalized recommendations.

We use phrase mining as a technique to extract the concepts that stick in readers' minds after reading. Phrase-level features

RecNLP 2019, January 27-28, 2019, Hawaii, USA

ACM ISBN 123-4567-24-567/08/06…$15.00
https://doi.org/10.475/123_4

have proved useful in many different natural language tasks. Liu et al. [15] use phrases to enhance the interpretability of domain-specific documents, and demonstrate effectiveness and efficiency via various experiments. Huang [10] proposes a phrase-level topic model which improves semantic coherence within each topic. Further, he conducts a human study which shows that the outputs of the phrase-level topic model are easier to understand. Wang et al. [25] propose a method to translate phrases in neural machine translation (NMT), enhancing the BLEU scores on a translation task from Chinese to English. In this paper, we leverage the power of phrases in briefing and interpreting to build a better article recommendation system.

3 SYSTEM OVERVIEW

An overview of our PhrecSys is shown in Fig. 2 The first step is extracting high-quality phrases in each article. For the efficiency and scalability needed by an automatic general article recommendation system, we use Autophrase [22], a data-driven and distant supervised learning algorithm.

Phrases from Autophrase with scores no lower than 0.5 are collected as candidates for phrase labeling. In this step, the phrase with the longest length is given a higher priority, i.e., longest match. We then tokenized the articles by the phrases with higher priority, which make the overall information of each token more compact. Finally, phrases are labeled in each article.

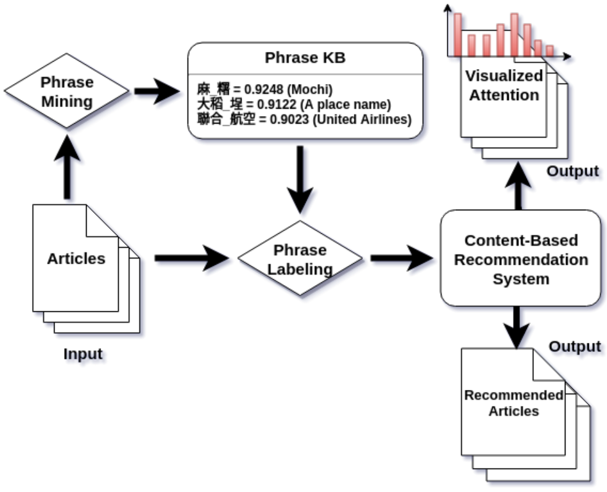


Figure 2: PhrecSys system overview

The labeling process yields phrase-based articles, that is, articles injected with phrase information. We apply Glove [19] to train the phrase embeddings for the next step. Note that punctuation is dropped in articles.

The five content-based recommendation models listed below are utilized to evaluate whether the added phrase mining component really benefits recommendation performance: TextCNN, CDSSM, MV-LSTM, K-NRM, and BiLSTM-SA. Given the article currently being read and a pool of candidate articles, these models generate the recommendation list ordered by recommendation score:

TextCNN. Similar to [11], convolutional neural networks with different kernel sizes are used to mimic unigram, bigram, and trigram language models, followed by a max-pooling layer and an asymmetric cosine similarity layer to calculate the score between two articles.

CDSSM [1]. Uses a single convolutional neural network to learn low-dimensional semantic vectors followed by multi-layer perceptrons (MLP), after which the normalized dot product is used to calculate the score between two articles.

MV-LSTM [23]. Incorporates Bi-LSTM to capture contextualized local information in each positional sentence and then uses the normalized dot product to generate an interaction tensor. K-Max pooling is used to obtain the top-k important information, after which MLP is used to produce the final score.

K-NRM [4]. A kernel-based neural model for document ranking which uses a translation matrix that models word-level similarities via word embeddings and utilizes kernels to extract multi-level soft-match features. A ranking layer is followed to generate ranking scores.

BiLSTM-SA [14]. Uses Bi-LSTM followed by a self-attention mechanism comprised of two linear layers with a softmax layer to generate multiple probabilities for each timestep.

In addition to the recommendation task itself, PhrecSys can visualize the learned focus of each article via the attention technique, which helps models learn to focus on important parts while dealing with long sequences such as news articles. We then observe how the model learns with features of different granularity. Here we select BiLSTM-SA [14] for attention visualization: we store the attention weights learned in each training epoch, and then plot the tokens in the current article with different color depths according to their attention weights. Given these different color depths, we can easily perceive which part of the article the model is focusing on for recommendation in the training process. Figure 3 shows an example; more details are provided below.

4 DATASET

Two datasets – Central News Agency (CNA)[1] and Taiwan People News (TPN)[2] – were provided by our collaborating partner cacaFly[3] for experiments. TPN and CNA are Taiwanese news websites articles written in Mandarin.

Two types of user behavior – clicks and views – are logged. Specifically, a *View* log is recorded when the user reads the current article a_c. When the user scrolls to the bottom of the web page to reach the recommendations, the partner system interprets this to mean that the user has finished reading a_c; a *Click* log is recorded when the user clicks on any of the recommendations. We generate *Click* log sequences to predict the articles that the user will click on next, and *View* log sequences to predict the articles that will be read next, accordingly.

For a *Click* log sequence, a positive article pair $(a_c, \hat{a_r})$ is generated from the article sequence if a_i appears in the *View* log and a_{i+1} in the *Click* log, $1 \leq i \leq n-1$. As recency is often taken into account for recommendation [2, 8], we generate no more than the

[1] http://www.cna.com.tw
[2] http://www.peoplenews.tw
[3] https://cacafly.com

Dataset	*Click*			*View*		
	Train	Val	Test	Train	Val	Test
CNA	28,950	4,825	4,825	635,495	97,832	89,498
TPN	17,041	2,841	2,841	137,029	21,305	20,122

Table 1: *Click* and *View* article pairs

latest 8 positive pairs per user for experiments if the user has read many articles. For the *View* log sequence, we select those articles in the article sequence with the *View* log, and pair them to generate it. That is, $(a_c, \hat{a_r})$ is a pair of successive articles that are both being read by the user. Along with each positive pair, there are m negative pairs $(a_c, \check{a_r})$, where the $\check{a_r}$'s are the un-clicked recommended articles for a_c.

Let l be the number of article pairs in each user's log sequence. For each user's *Click* or *View* log sequence, the last article pair $(a_c, a_r)_l$ is for testing, the second-to-last is for validation, and the rest are for training. Table 1 shows the statistics of the training, validation, and testing data.

5 EXPERIMENTS AND RESULTS

In this section, we first describe the datasets, model settings, and evaluation metrics, after which we discuss the performance and present the visualization of attention weights during training on word-level and phrase-level systems.

Dataset	CNA	TPN
Articles	257,705	54,033

Table 2: Number of articles in each dataset

5.1 Settings and Metrics

For each dataset, we pre-train 50-dimensional phrase embeddings and word embeddings with GloVe [19] separately, using only the articles within each dataset. The total number of articles used to train word/phrase embeddings for each dataset are listed in Table 2. Below we describe the parameter settings for different recommendation systems. Note that hinge loss is used as objective function.

TextCNN. We set CNN filter number to 32, the maximum article length to 512, and α in the asymmetric cosine to 0.85. During training we used the Adam optimizer with a learning rate of 0.1.

CDSSM. We set the CNN filter number to 32 and selected 128 as the MLP output dimension for article semantic representation. During training we used the Adam optimizer with a learning rate of 0.01.

MV-LSTM. We set the LSTM hidden size to 32 and the top-k value to 512. Two layer MLPs were applied at the end of model, and 64 and 1 were the number of output dimensions respectively. During training we used the Adam optimizer with a learning rate of 0.01.

KNRM. We set the kernel number to 32 and the sigma value to 0.05. During training we used the Adam optimizer with a learning rate of 0.001.

BiLSTM-SA. We set the LSTM hidden size to 32 for each direction, d_a to 100 for the fully-connected layer inside self-attention, and the attention number r to 15. During training we used the Adam optimizer with a learning rate of 0.01.

We adopted several evaluation metrics to fairly evaluate the recommendation performance. To evaluate the performance of recommendation, per the literature we used the mean reciprocal rank (MRR) [24], accuracy (Acc) [7], hit ratio at 3 (h@3), and hit ratio at 5 (h@5) [12]. The hit ratio h@k intuitively measures whether the test item is present in the top-k list.

$$\begin{gathered} Acc = \frac{1}{|U|} \sum_{u=1}^{|U|} accuracy(u) \\ accuracy(u) = \begin{cases} 1, & \text{if } Rank(score_{c,\hat{r}}) = 1 \\ 0, & \text{otherwise} \end{cases} \end{gathered} \tag{1}$$

$$h@N = \frac{1}{|U|} \sum_{u=1}^{|U|} h(u,N).\ h(u,N) = \begin{cases} 1, & \text{if } Rank(score_{c,\hat{r}}) \leq N \\ 0, & \text{otherwise} \end{cases} \tag{2}$$

Assuming $|U|$ testing instances in the testing set, each testing instance $u \in U$ contained one positive pair $(a_c, \hat{a_r})$ and m negative pairs $(a_c, \check{a_r})$. The average MRR was calculated as

$$MRR = \frac{1}{|U|} \sum_{u=1}^{|U|} \frac{1}{Rank(score_{c,\hat{r}})}, \tag{3}$$

where $Rank(score_{c,\hat{r}})$ was the rank of the positive pair score among all scores of pairs in u.

5.2 Recommendation Performance

We examined the performance on two datasets (CNA, TPN), both of which exhibit two types of behavior – *Click* and *View*. The five models described in Section 3 and 5.1 were evaluated to compare their performance with and without phrase mining: the results are shown in Table 3.

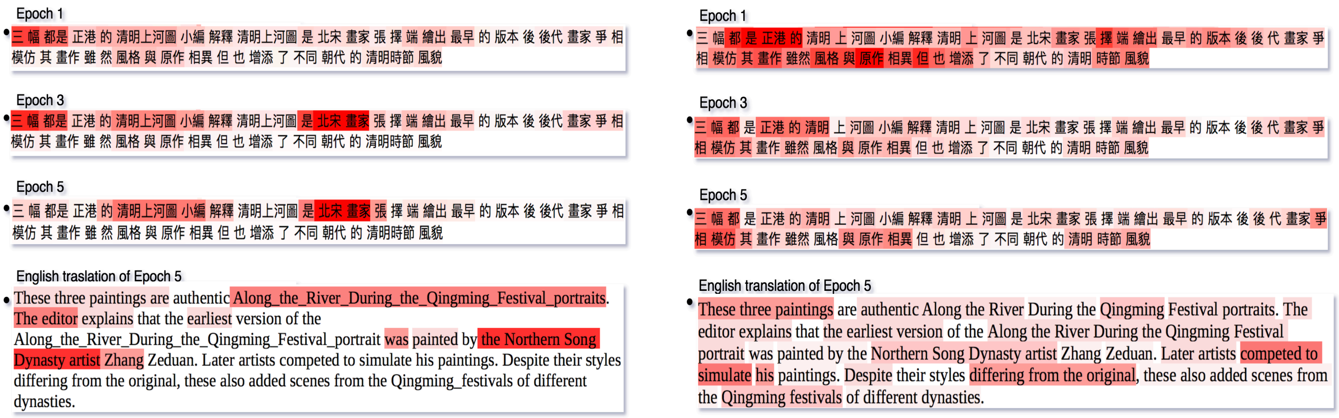
For the prediction of the next clicked article (the article that the user clicks on next), most models are slightly enhanced by phrase mining. However, for the next viewed article (the article that the user reads next), the performance of all models is improved significantly when leveraging phrase-level features. This may be because text content might not be the first priority to users when clicking, as eye-catching titles or images may hold a special attraction. In this case, though phrases capture the key points of an article, their benefit is limited. For view prediction, in contrast, the text content is essential. We observe that a large amount of views come from websites other than the current news media and very often from search engines. In these cases, the users' intention is clearly a specific topic that interests them and hence they tend to read contents that are logically related successively. Another reason is that as users read through the current article, its content becomes a important factor in the surfing history. The notable enhancement of phrase-level features shows that phrases can enhance the model's ability to learn the relatedness of contents between articles.

5.3 Visualization

To visualize the focus points being learned by the model, we record the attention weights of composite units (words or phrases plus words) of the current article in each training epoch and observe their change over time. In the experiment, we selected BiLSTM-SA described in Section 3 to demonstrate the result. Figure 3 shows

Model	CNA-click				TPN-click				CNA-view				TPN-view			
	MRR	Acc	h@3	h@5	MRR	Acc	h@3	h@5	MRR	Acc	h@3	h@5	MRR	Acc	h@3	h@5
K-NRM	**.340**	.132	.393	.612	.282	.095	.281	.477	.263	.087	.259	.428	.293	.104	.309	.495
MV-LSTM	.334	.127	.385	.606	.335	.140	.363	.571	.476	.283	.527	.738	.484	.293	.574	.741
CDSSM	**.384**	**.169**	**.460**	**.683**	.371	.172	.419	.625	.563	.385	.667	.792	.515	.319	.623	.789
BiLSTM-SA	.396	.182	**.474**	**.694**	.382	.178	.438	.642	.723	.621	.798	.848	.695	.555	.759	.856
TextCNN	.392	.175	.478	.686	.384	.181	**.440**	.648	.585	.421	.673	.804	.562	.379	.666	.809
K-NRM + Phrase	.337	**.136**	**.415**	**.618**	**.353**	**.135**	**.369**	**.502**	**.294**	**.114**	**.302**	**.479**	**.312**	**.115**	**.329**	**.556**
MV-LSTM + Phrase	**.388**	**.180**	**.454**	**.679**	**.360**	**.157**	**.410**	**.621**	**.699**	**.611**	**.772**	**.837**	**.657**	**.500**	**.759**	**.831**
CDSSM + Phrase	.369	.164	.431	.650	**.377**	**.175**	**.433**	**.631**	**.609**	**.437**	**.716**	**.845**	**.607**	**.437**	**.709**	**.839**
BiLSTM-SA + Phrase	**.400**	**.190**	.464	.676	**.404**	**.186**	**.455**	**.653**	**.727**	**.632**	**.791**	**.879**	**.706**	**.582**	**.798**	**.875**
TextCNN + Phrase	**.395**	**.180**	**.481**	**.700**	**.391**	**.184**	**.440**	**.650**	**.709**	**.625**	**.769**	**.860**	**.704**	**.588**	**.788**	**.866**

Table 3: Result of click and view predictions when *len* = 8



Epoch 1

- 三 幅 都是 正港 的 清明上河圖 小編 解釋 清明上河圖 是 北宋 畫家 張 擇 端 繪出 最早 的 版本 後 後代 畫家 爭 相 模仿 其 畫作 雖 然 風格 與 原作 相異 但 也 增添 了 不同 朝代 的 清明時節 風貌

Epoch 3

- 三 幅 都是 正港 的 清明上河圖 小編 解釋 清明上河圖 是 北宋 畫家 張 擇 端 繪出 最早 的 版本 後 後代 畫家 爭 相 模仿 其 畫作 雖 然 風格 與 原作 相異 但 也 增添 了 不同 朝代 的 清明時節 風貌

Epoch 5

- 三 幅 都是 正港 的 清明上河圖 小編 解釋 清明上河圖 是 北宋 畫家 張 擇 端 繪出 最早 的 版本 後 後代 畫家 爭 相 模仿 其 畫作 雖 然 風格 與 原作 相異 但 也 增添 了 不同 朝代 的 清明時節 風貌

English traslation of Epoch 5

- These three paintings are authentic Along_the_River_During_the_Qingming_Festival_portraits. The editor explains that the earliest version of the Along_the_River_During_the_Qingming_Festival_portrait was painted by the Northern Song Dynasty artist Zhang Zeduan. Later artists competed to simulate his paintings. Despite their styles differing from the original, these also added scenes from the Qingming_festivals of different dynasties.

(a) Phrase

Epoch 1

- 三 幅 都 是 正港 的 清明 上 河圖 小編 解釋 清明 上 河圖 是 北宋 畫家 張 擇 端 繪出 最早 的 版本 後 後 代 畫家 爭 相 模仿 其 畫作 雖然 風格 與 原作 相異 但 也 增添 了 不同 朝代 的 清明 時節 風貌

Epoch 3

- 三 幅 都 是 正港 的 清明 上 河圖 小編 解釋 清明 上 河圖 是 北宋 畫家 張 擇 端 繪出 最早 的 版本 後 後 代 畫家 爭 相 模仿 其 畫作 雖然 風格 與 原作 相異 但 也 增添 了 不同 朝代 的 清明 時節 風貌

Epoch 5

- 三 幅 都 是 正港 的 清明 上 河圖 小編 解釋 清明 上 河圖 是 北宋 畫家 張 擇 端 繪出 最早 的 版本 後 後 代 畫家 爭 相 模仿 其 畫作 雖然 風格 與 原作 相異 但 也 增添 了 不同 朝代 的 清明 時節 風貌

English translation of Epoch 5

- These three paintings are authentic Along the River During the Qingming Festival portraits. The editor explains that the earliest version of the Along the River During the Qingming Festival portrait was painted by the Northern Song Dynasty artist Zhang Zeduan. Later artists competed to simulate his paintings. Despite their styles differing from the original, these also added scenes from the Qingming festivals of different dynasties.

(b) Word

Figure 3: Changes in attention weights during training process

the effect of phrase information. Figure 3a is the article parsed and trained with phrase-level units, whereas Figure 3b is the article parsed and trained with word-level units. The attention weights of epoch 5 illustrate the overall result: when phrase information is provided, attention weights saturate to a reasonable number of units. In comparison, without phrase mining, weights are evenly distributed, with no obvious key units to be found in the visualization.

Comparing Figure 3a and Figure 3b from Epochs 1 and 3 to Epoch 5, we observe that the attention weights are both evenly distributed and show no particular focus on anything in the beginning. Later, in Epoch 3, training with phrase-level features causes weights to be more concentrated and accurately focused on meaningful parts; finally, in Epoch 5, training with phrase-level information successfully leads to fewer but more informative cues. We conclude that phrases contain more compact information and are more easily utilized for recommendation models.

6 CONCLUSION

In this paper, we propose PhrecSys – a system that first extracts high-quality phrases from data-driven phrase mining and tokenizes articles with these phrases, and then utilizes selected content-based systems for recommendation. We show that the proposed approach yields improvements in both click and view prediction. Additionally, we visualize the learning process of both phrase-level and word-level models to illustrate that the merit of phrase mining for recommendation is its ability to put the focus on key units. With this advantage, the visualization itself can be interpreted and is of great value to public opinion analytics.

ACKNOWLEDGMENTS

The authors would like to thank our collaborating partner cacaFly, the leader of digital advertising in Taiwan, for providing user website log in this paper. Research of this paper was partially supported by cacaFly, under the collaborative project contract O5T-1060629-1Cb.

REFERENCES

[1] 2014. *Learning Semantic Representations Using Convolutional Neural Networks for Web Search.* WWW 2014.

[2] Amr Ahmed, Choon Hui Teo, SVN Vishwanathan, and Alex Smola. 2012. Fair and balanced: Learning to present news stories. In *Proceedings of the fifth ACM international conference on Web search and data mining*. ACM, 333–342.

[3] Ting Chen, Liangjie Hong, Yue Shi, and Yizhou Sun. 2017. Joint Text Embedding for Personalized Content-based Recommendation. *arXiv preprint arXiv:1706.01084* (2017).

[4] Xiong Chenyan, Dai Zhuyun, Callan Jamie, Liu Zhiyuan, and Power Russell. 2017. End-to-End Neural Ad-hoc Ranking with Kernel Pooling. *CoRR* abs/1706.06613 (2017). arXiv:1706.06613

[5] Alexis Conneau, Holger Schwenk, Loïc Barrault, and Yann Lecun. 2017. Very deep convolutional networks for text classification. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, Vol. 1. 1107–1116.

[6] Souvik Debnath, Niloy Ganguly, and Pabitra Mitra. 2008. Feature weighting in content based recommendation system using social network analysis. In *Proceedings of the 17th international conference on World Wide Web*. ACM, 1041–1042.

[7] Robin Devooght and Hugues Bersini. 2017. Long and Short-Term Recommendations with Recurrent Neural Networks. In *Proceedings of the 25th Conference on User Modeling, Adaptation and Personalization*. ACM, 13–21.

[8] Doychin Doychev, Aonghus Lawlor, Rachael Rafter, and Barry Smyth. 2014. An analysis of recommender algorithms for online news. In *CLEF 2014 Conference and Labs of the Evaluation Forum: Information Access Evaluation Meets Multilinguality, Multimodality and Interaction, 15-18 September 2014, Sheffield, United Kingdom*. 177–184.

[9] Kelvin Guu, Tatsunori B Hashimoto, Yonatan Oren, and Percy Liang. 2017. Generating sentences by editing prototypes. *arXiv preprint arXiv:1709.08878* (2017).

[10] Weijing Huang. 2018. PhraseCTM: Correlated Topic Modeling on Phrases within Markov Random Fields. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. 521–526.

[11] Yoon Kim. 2014. Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882* (2014).

[12] Vaibhav Kumar, Dhruv Khattar, Shashank Gupta, Manish Gupta, and Vasudeva Varma. 2017. Deep Neural Architecture for News Recommendation.

[13] Quoc Le and Tomas Mikolov. 2014. Distributed representations of sentences and documents. In *International Conference on Machine Learning*. 1188–1196.

[14] Zhouhan Lin, Minwei Feng, Cícero Nogueira dos Santos, Mo Yu, Bing Xiang, Bowen Zhou, and Yoshua Bengio. 2017. A Structured Self-attentive Sentence Embedding. *CoRR* abs/1703.03130 (2017). arXiv:1703.03130

[15] Jialu Liu, Xiang Ren, Jingbo Shang, Taylor Cassidy, Clare R. Voss, and Jiawei Han. 2016. Representing Documents via Latent Keyphrase Inference. In *Proceedings of the 25th International Conference on World Wide Web (WWW '16)*. 1057–1067.

[16] Jialu Liu, Jingbo Shang, Chi Wang, Xiang Ren, and Jiawei Han. [n. d.]. Mining Quality Phrases from Massive Text Corpora. In *Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data - SIGMOD 15*.

[17] Jianmo Ni, Zachary C Lipton, Sharad Vikram, and Julian McAuley. 2017. Estimating Reactions and Recommending Products with Generative Models of Reviews. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, Vol. 1. 783–791.

[18] Michael J Pazzani and Daniel Billsus. 2007. Content-based recommendation systems. In *The adaptive web*. Springer, 325–341.

[19] Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*. 1532–1543.

[20] S. E. Robertson and S. Walker. 1994. Some Simple Effective Approximations to the 2–Poisson Model for Probabilistic Weighted Retrieval. In *ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*. 345–354.

[21] Andrew I Schein, Alexandrin Popescul, Lyle H Ungar, and David M Pennock. 2002. Methods and metrics for cold-start recommendations. In *Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, 253–260.

[22] Jingbo Shang, Jialu Liu, Meng Jiang, Xiang Ren, Clare R. Voss, and Jiawei Han. 2017. Automated Phrase Mining from Massive Text Corpora. *CoRR* abs/1702.04457 (2017). arXiv:1702.04457

[23] Wan Shengxian, Lan Yanyan, Guo Jiafeng, Xu Jun, Pang Liang, and Cheng Xueqi. 2015. A Deep Architecture for Semantic Matching with Multiple Positional Sentence Representations. *CoRR* abs/1511.08277 (2015). arXiv:1511.08277

[24] Ellen M Voorhees et al. 1999. The TREC-8 Question Answering Track Report.. In *Trec*, Vol. 99. 77–82.

[25] Xing Wang, Zhaopeng Tu, Deyi Xiong, and Min Zhang. 2017. Translating Phrases in Neural Machine Translation. *CoRR* abs/1708.01980 (2017). arXiv:1708.01980