On the Effect of Low-Frequency Terms on Neural-IR Models

Sebastian Hofstätter
TU Wien
s.hofstaetter@tuwien.ac.at

Navid Rekabsaz*
Idiap Research Institute
navid.rekabsaz@idiap.ch

Carsten Eickhoff
Brown University
carsten@brown.edu

Allan Hanbury
TU Wien
hanbury@ifs.tuwien.ac.at

ABSTRACT

Low-frequency terms are a recurring challenge for information retrieval models, especially neural IR frameworks struggle with adequately capturing infrequently observed words. While these terms are often removed from neural models – mainly as a concession to efficiency demands – they traditionally play an important role in the performance of IR models. In this paper, we analyze the effects of low-frequency terms on the performance and robustness of neural IR models. We conduct controlled experiments on three recent neural IR models, trained on a large-scale passage retrieval collection. We evaluate the neural IR models with various vocabulary sizes for their respective word embeddings, considering different levels of constraints on the available GPU memory.

We observe that despite the significant benefits of using larger vocabularies, the performance gap between the vocabularies can be, to a great extent, mitigated by extensive tuning of a related parameter: the number of documents to re-rank. We further investigate the use of subword-token embedding models, and in particular FastText, for neural IR models. Our experiments show that using FastText brings slight improvements to the overall performance of the neural IR models in comparison to models trained on the full vocabulary, while the improvement becomes much more pronounced for queries containing low-frequency terms.

ACM Reference Format:
Sebastian Hofstätter, Navid Rekabsaz, Carsten Eickhoff, and Allan Hanbury. 2019. On the Effect of Low-Frequency Terms on Neural-IR Models. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '19), July 21–25, 2019, Paris, France.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3331184.3331344

1 INTRODUCTION

Neural network approaches for Information Retrieval have been showing promising performance in a wide range of document retrieval tasks. Various studies apply neural methods by introducing pre-trained word embeddings into classical IR models [13, 14], adapting word embeddings to retrieval tasks [4, 6], or proposing altogether novel neural IR models [3, 7, 8, 16]. An essential core building block of all these approaches is the word embedding model, which defines the semantic relations between the terms.

Typically, word embeddings are defined on a fixed vocabulary. As a common practice in neural network approaches, terms with very low collection frequencies are pruned from the vocabulary, becoming Out-Of-Vocabulary (OOV) terms. The reason for limiting the vocabulary in this way often stems from (GPU) memory constraints, efficiency considerations, or noise reduction efforts.

However, in the context of retrieval modeling, low-frequency terms are known to bear high degrees of informativeness or salience, and therefore play an important role in identifying relevant documents. In classical IR, the importance of such terms is quantified by term salience measures, such as the Inverse Document Frequency. Removing low-frequency terms in the training stage of neural IR models potentially harms the effectiveness and robustness of the derived models, especially for queries containing the affected terms. Even if neural IR models cover the full collection vocabulary, two issues remain: (1) The model performs poorly on previously unseen terms appearing at retrieval time (OOV terms). (2) Due to the lack of training data for low-frequency terms, the learned vectors may not be semantically robust.

In this study, we explore the effect of low-frequency terms on the effectiveness and robustness of neural IR models. We conduct an extensive range of controlled experiments on three recent neural IR models, namely *KNRM* [16], *CONV-KNRM* [3], and *MatchPyramid* [8], evaluated on the MS MARCO [1] passage ranking collection, and finally propose potential solutions to the general underlying vocabulary issue in neural IR models.

The novel contributions of this paper are two-fold: We begin by exploring the performance of neural IR models trained on different vocabularies (Section 4). We observe that despite the significant benefits of using larger vocabularies, model performance is highly sensitive to another essential parameter common to virtually all neural IR models: the re-ranking threshold, which defines how many of the initially retrieved documents are re-ranked by a neural IR model. We investigate the relationship between vocabulary size and re-ranking threshold, noting the sensitivity of the models to the latter, especially for models with smaller vocabularies. Our results suggest that a well-tuned re-ranking threshold can largely mitigate the negative effect of pruned vocabularies.

Secondly, we study the effect of embedding sub-word tokens in comparison to using the full vocabulary of word-level tokens (Section 5). In particular, we investigate the use of FastText [2], a model based on the composition of character n-gram vector representations, designed to address OOV issues. Our results suggest that the overall performance of the model with FastText remains close to the results of using the full vocabulary. However, character-level models achieve significantly better performance on queries

*The contribution of the second author is based upon the work supported by the Office of the Director of National Intelligence (ODNI), Intelligence Advanced Research Projects Activity (IARPA), via AFRL Contract #FA8650-17-C-9116. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the ODNI, IARPA, or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon.

SIGIR '19, July 21–25, 2019, Paris, France

ACM ISBN 978-1-4503-6172-9/19/07…$15.00
https://doi.org/10.1145/3331184.3331344

containing low-frequency terms. We argue that this is due to better generalization of the character-level model that benefits from other words with similar n-gram contexts. This early-stage study therefore recommends the use of sub-word token embeddings as a strategy for retaining the effectiveness and robustness of neural IR models, especially with regard to low-frequency query terms.

2 BACKGROUND AND RELATED WORK

In this section, we briefly explain the sub-word embeddings, followed by discussing related work to our study.

Sub-word embedding models produce a vector representation of a word based on composing embeddings of the character n-grams composing the word. In this way, the models can provide a semantically meaningful embedding vector even for unseen terms by exploiting the contexts of the observed terms with similar character n-grams and there are virtually no out-of-vocabulary terms. The FastText model [2], an effective and efficient sub-word embedding model, simply sums up the character n-gram vectors to build the word embedding. For highly frequent terms, FastText directly assigns a vector per word. ELMo [11] is another well-known character-based embedding model, which in addition, takes into account the context around the word. In this work, we use FastText due its direct comparability to traditional word embeddings.

In more traditional retrieval models, Woodland et al. [15] explore the role of OOV terms for spoken document retrieval, proposing query and document expansion approaches. To the best of our knowledge there is no existing research on the effect of low-frequency terms on neural IR models.

Other studies explore related aspects of neural IR models. Pyreddy et al. [12] investigate the variance and consistency of kernel-based neural models over various parameter initializations. Zamani et al. [18] propose a method to skip the re-ranking step, and directly retrieve documents from an index of sparse representations. In contrast, in this paper, we analyze the sensitivity of the neural IR models to the re-ranking threshold parameter, since most recently proposed neural models are based on the re-ranking mechanism.

3 EXPERIMENT DESIGN

We conduct our experiments on the MS MARCO [1] passage re-ranking collection. The collection provides a large set of informational question-style queries from Bing's search logs, accompanied by human-annotated relevant/non-relevant passages. Besides training data, MS MARCO provides a development set – containing queries and relevance data for evaluation – in two sizes: sample[1] and full. In our experiments, we use the queries from the sample as our validation set and the rest of the full development set as our test set. In total, the collection consists of 8,841,822 documents, 6,980 queries for validation, and 48,598 queries for test purposes.

Resources. We use GloVe [10] word embeddings with 300 dimensions[2], and the FastText model, trained on the Wikipedia corpus of August 2015, with trigram-character subwords in 200 dimensions.

We create several vocabularies based on varying thresholds to the collection frequency of terms. In our experiments, we refer to the set of terms with frequency greater or equal to n, as Voc-n. Voc-Full uses all the terms in the collection. The details of the resulting vocabularies, as well as the corresponding statistics of OOV terms, are shown in Table 1.

Table 1: Left: Details of the vocabularies. Right: Percentage and absolute number of test set queries with ≥ 1 OOV term

Name - Min #	# Terms	% Covered Terms	Size	OOV Queries %	OOV Queries #
Voc-Full	3,525,473	100.0	4.23 GB	0	0
Voc-5	542,878	15.4	651 MB	1.23	596
Voc-10	314,607	8.9	378 MB	1.98	962
Voc-25	169,983	4.8	204 MB	5.07	2464
Voc-50	111,815	3.2	134 MB	8.06	3917
Voc-100	75,805	2.2	91 MB	12.07	5864
FastText	2,950,302	100.0	2.36 GB	0	0

Evaluation. We evaluate our models with the main metric for the MS MARCO ranking challenge: the Mean Reciprocal Rank measure (MRR), as well as Recall, both at rank 10. Statistical significance tests are done using a two sided paired t-test ($p < 0.05$).

Neural Retrieval Models. ***KNRM*** [16] establishes a similarity matching matrix using the embeddings of query and document terms. The model then estimates the relevance score based on the outputs of a set of Gaussian kernel functions, applied on the matching matrix.
CONV-KNRM [3] extends *KNRM* by adding a Convolutional Neural Network (CNN) layer on top of the word embedding matrix, enabling learning word-level n-gram representations.
MatchPyramid [8] ranking model is inspired by deep neural image processing architectures. Similar to *KNRM*, the model first computes the similarity matching matrix, which is used for several stacked layers of CNN with dynamic max-pooling. Like the two previous models, this architecture facilitates end-to-end training.

Implementation and Parameter Setting. We use the Anserini [17] toolkit to compute the *BM25*, and *RM3* models. The model parameters are tuned on the validation set, resulting in $k_1 = 0.6$, $b = 0.8$ for *BM25*. We observe no significant performance increase for *RM3*, therefore we only report *BM25* baseline results. The *BM25* rankings are used as a starting point for the neural re-ranking models.

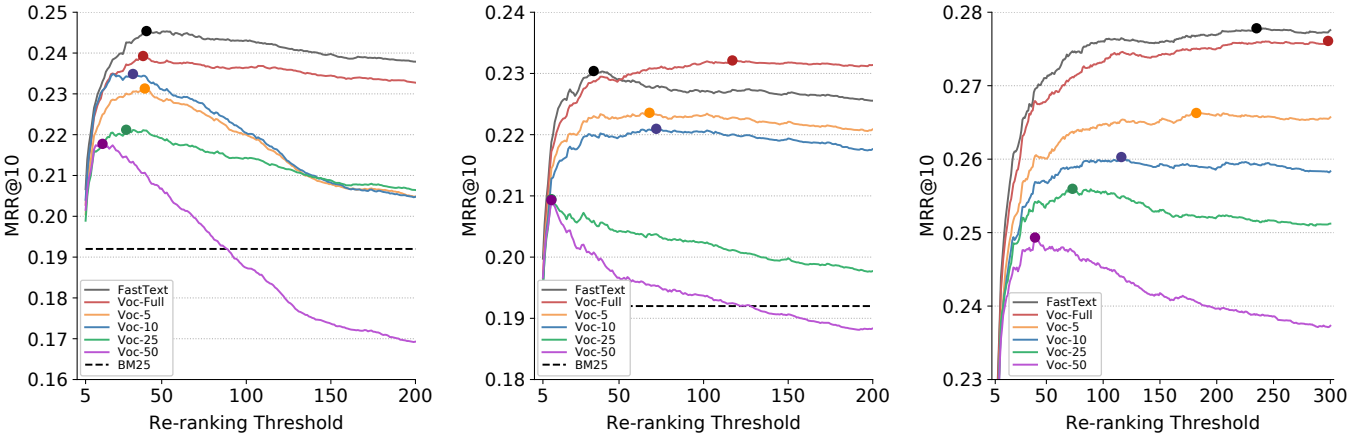
We implement the neural models in PyTorch [9].[3] We project all characters to lower case and apply tokenization using the WordTokenizer provided by AllenNLP [5]. We use the Adam optimizer with learning rate 0.001, 1 epoch, and early stopping. We use a batch size of 64, and the maximum word length of queries and documents is set to 30 and 180, respectively. In all models, the pre-trained word embeddings are updated during training.

Regarding model-specific parameters, for *KNRM* and *CONV-KNRM*, we set the number of kernels to 11 with the mean values of the Gaussian kernels varying from −1 to +1 in steps of 0.2 (one extra kernel is added for exact matching), and standard deviation of 0.1 for all kernels. The dimension of the CNN vectors in *CONV-KNRM* is set to 128. In the MatchPyramid model, we set the number of convolution layers to 5, each with kernel size 3×3 and 16 convolution channels. For each model, we find the best re-ranking threshold parameter by extensively tuning it on a range from 1 to 300, based on the MRR results of the validation set.

[1] Provided in the form of evaluation tuples: top1000.dev.tsv
[2] 42B lower-cased (CommonCrawl) from: *https://nlp.stanford.edu/projects/glove/*
[3] Our code is available at *https://github.com/sebastian-hofstaetter/sigir19-neural-ir*

Table 2: Evaluation results using different vocabulary sizes & best re-ranking threshold on the validation set. Top: best performance on the validation set. Bottom: results on the test set. The best results per model are shown in bold

	Model	Voc-100		Voc-50		Voc-25		Voc-10		Voc-5		Voc-Full		FastText	
		MRR	Recall	MRR	Recall	MRR	Recall	MRR	Recall	MRR	Recall	MRR	Recall	MRR	Recall
Val.	***MatchPyramid***	0.220	0.423	0.218	0.427	0.224	0.442	0.234	0.446	0.231	0.459	0.239	0.467	**0.245**	0.477
	KNRM	0.209	0.404	0.209	0.404	0.209	0.404	0.221	0.454	0.224	0.457	**0.232**	**0.467**	0.230	0.456
	CONV-KNRM	0.253	0.469	0.249	0.474	0.256	0.480	0.261	0.492	0.266	0.504	0.276	0.526	**0.278**	0.519
Test	***MatchPyramid***	0.223	0.426	0.227	0.431	0.221	0.442	0.237	0.458	0.232	0.458	0.239	0.465	**0.247**	**0.472**
	KNRM	0.210	0.407	0.211	0.407	0.211	0.407	0.222	0.453	0.225	0.455	**0.235**	**0.468**	0.227	0.451
	CONV-KNRM	0.248	0.472	0.249	0.470	0.250	0.481	0.260	0.488	0.264	0.503	0.273	0.518	**0.275**	**0.519**



(a) ***MatchPyramid*** (b) ***KNRM*** (c) ***CONV-KNRM***

Figure 1: Sensitivity of the models to the re-ranking threshold parameter for different vocabularies. The best performing parameter setting are indicated on the plots.

4 EFFECT OF THE VOCABULARY SIZE

The performance of the neural ranking models, trained on various vocabularies as well as on FastText embeddings, on both validation and test sets are shown in Table 2. We calculate tests of significance between the pairs of rankings, and mention the results in the following. We also evaluate *BM25*, achieving an MRR of 0.192 and Recall of 0.407 on the test set. Consistent with previous studies, the *BM25* model is outperformed by all neural ranking models, and *CONV-KNRM* shows the best overall performance [3, 8, 16].

Comparing the results over each model, in two out of the three models, the FastText embedding significantly outperforms Voc-Full, while FastText only requires 55% of the memory needed by Voc-Full (based on the statistics in Table 1). Looking at the results of the models with various vocabulary sizes, using Voc-Full brings significant advantages in comparison to using smaller vocabularies. However, their differences become marginal, especially for the models with Voc-5 and Voc-10 vocabulary sets, taking into account that the embeddings of the Voc-5 and Voc-10 vocabularies require much less memory space, namely only 15% (Voc-5) and 8% (Voc-10) of the memory used by the Voc-Full embeddings.

While the reported results are based on an exhaustive tuning of hyper-parameters on the validation set, in the following we study the sensitivity of the models to the re-ranking threshold, an important – but not well studied – hyper-parameter of the neural IR models. Figure 1 demonstrates the sensitivity of the three neural IR models to the changes of the re-ranking threshold parameter. Looking at the trends in the plots, as the performance improves, either by using a better performing model or a bigger vocabulary size, the models become less sensitive to the re-ranking threshold. Such that the optimal re-ranking thresholds also become larger, indicating that the model is able to effectively generalize over a larger set of non-relevant documents. Since increasing the re-ranking threshold mostly adds non-relevant documents. On the other hand, models with lower performances (*MatchPyramid* and *KNRM*), especially with smaller vocabularies, are highly sensitive to the re-ranking threshold. For such models, an exhaustive parameter search provides a significant enhancement. This indicates the importance of well-tuning the re-ranking threshold parameter, especially in scenarios with constrained memory resources.

Finally, to confirm whether the effects of tuning the re-ranking threshold on the validation set is also transferred to the test set, we compare the results on the validation and test set in Table 2. As shown, even on models with high sensitivity to the re-ranking threshold, the results are highly similar, indicating the effectiveness of extensive tuning of the re-ranking threshold[4].

[4]We should note that the cause of this effect might be due to high underlying similarities between the validation and test set of the particular collection. However, further investigations on this aspect is out of the scope of this paper.

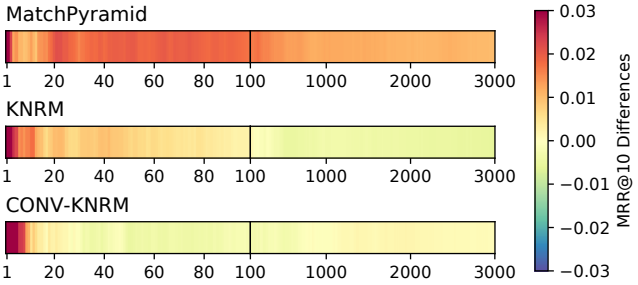


Figure 2: MRR differences of the models, trained on the FastText embeddings and the embeddings with full vocabularies, over the queries with minimum collection frequency of their terms smaller or equal to the X-axis
(red = FastText is better, blue=Vocab-Full is better)

5 QUERIES WITH LOW-FREQUENCY TERMS

In this section, we take a closer look at the differences between the models trained on the traditional embeddings (GloVe in our experiments) using different vocabularies, and the ones trained on the FastText embeddings.

Figure 2 shows the MRR differences of the neural ranking models, using the traditional embeddings with the Voc-Full vocabularies, to the ones using FastText, over the range of collection frequencies. For each point on the X axes, we calculate the MRR values for the queries, which at least have one term with collection frequency of equal to or smaller than the corresponding value of that point. The figure reveals strong contrast between the area, related to the queries with very low-frequency terms, and the rest, indicating higher performances of the models with FastText for these queries.

Let us have a closer look at this area. Figure 3 shows the MRR of the *CONV-KNRM* models, using the traditional word embeddings with different vocabularies, as well as the one using the FastText embeddings, for queries with very infrequent terms. The MRR values are calculated in the same fashion as in Figure 2.

As shown, the model with FastText by a large margin improves all other models, especially until a collection frequency of around 10 to 15. Interestingly, BM25, as an exact term matching model, shows better performance than neural IR models with traditional embeddings, especially on very low values. We argue that the low performance of the models with traditional embeddings is due to the lack of enough contexts for learning meaningful representations, which causes ineffective semantic similarity estimations. On the other hand, the subword embeddings exploit the contexts of other observed terms in the collection with similar character n-grams. Therefore, the neural ranking models with subword embeddings still benefit from meaningful semantic relations between very infrequent words, outperforming the ranking models based on traditional embeddings as well as exact matching.

6 CONCLUSION

Our work takes a first step to understanding the effects of infrequent terms in neural ranking models, and exploit novel representation learning approaches to address it. We first study the sensitivity of the neural IR models to their vocabulary size, pointing out the importance of fine-grained tuning of the re-ranking threshold. We then investigate the effects of using subword embeddings in neural

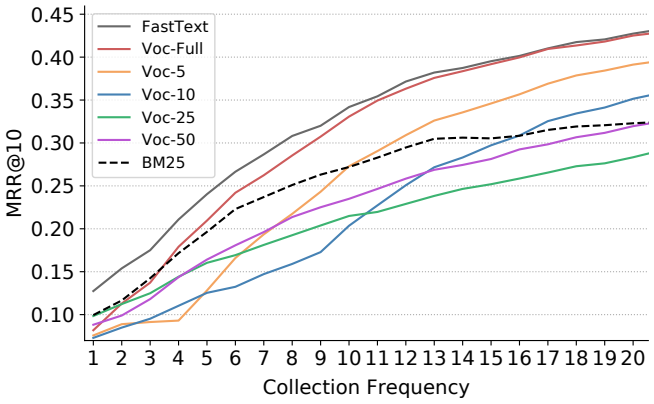


Figure 3: MRR results of *CONV-KNRM* over the queries with at least one term with collection frequency smaller than or equal to the values on the X-axis

IR models, showing that using these embeddings in particular brings remarkable improvements to the performance of queries containing very low-frequency terms. As future work, we aim to pursue the investigations of this study into the area of query performance prediction of neural IR models.

REFERENCES

[1] Payal Bajaj, Daniel Campos, Nick Craswell, Li Deng, Jianfeng Gao, Xiaodong Liu, Rangan Majumder, Andrew Mcnamara, Bhaskar Mitra, and Tri Nguyen. 2016. MS MARCO : A Human Generated MAchine Reading COmprehension Dataset. In *Proc. of NIPS*.

[2] Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching Word Vectors with Subword Information. *Tr. of the ACL* 5 (2017).

[3] Zhuyun Dai, Chenyan Xiong, Jamie Callan, and Zhiyuan Liu. 2018. Convolutional Neural Networks for Soft-Matching N-Grams in Ad-hoc Search. In *Proc. of WSDM*.

[4] Fernando Diaz, Bhaskar Mitra, and Nick Craswell. 2016. Query Expansion with Locally-Trained Word Embeddings. In *In Proc. of ACL*.

[5] Matt Gardner, Joel Grus, Mark Neumann, Oyvind Tafjord, Pradeep Dasigi, Nelson F. Liu, Matthew Peters, Michael Schmitz, and Luke S. Zettlemoyer. 2017. AllenNLP: A Deep Semantic Natural Language Processing Platform.

[6] Sebastian Hofstätter, Navid Rekabsaz, Mihai Lupu, Carsten Eickhoff, and Allan Hanbury. 2019. Enriching Word Embeddings for Patent Retrieval with Global Context. In *Proc. of ECIR*.

[7] Liang Pang, Yanyan Lan, Jiafeng Guo, Jun Xu, and Xueqi Cheng. 2017. A Deep Investigation of Deep IR Models. In *Proc. of SIGIR Neu-IR'17*.

[8] Liang Pang, Yanyan Lan, Jiafeng Guo, Jun Xu, Shengxian Wan, and Xueqi Cheng. 2016. Text Matching as Image Recognition. In *Proc of. AAAI*.

[9] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. 2017. Automatic differentiation in PyTorch. In *NIPS-W*.

[10] Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *In Proc of EMNLP*.

[11] Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proc. of NAACL*.

[12] Mary Arpita Pyreddy, Varshini Ramaseshan, Narendra Nath Joshi, Zhuyun Dai, Chenyan Xiong, Jamie Callan, and Zhiyuan Liu. 2018. Consistency and Variation in Kernel Neural Ranking Model. In *Proc. of SIGIR*.

[13] Navid Rekabsaz, Mihai Lupu, Allan Hanbury, and Hamed Zamani. 2017. Word Embedding Causes Topic Shifting; Exploit Global Context!. In *In Proc. of SIGIR*.

[14] Navid Rekabsaz, Mihai Lupu, Allan Hanbury, and Guido Zuccon. 2016. Generalizing translation models in the probabilistic relevance framework. In *In Proc. of CIKM*.

[15] Philip Woodland, Sue Johnson, Pierre Jourlin, and Karen Spärck Jones. 2000. Effects of out of vocab. words in spoken document retrieval. In *In Proc. of SIGIR*.

[16] Chenyan Xiong, Zhuyun Dai, Jamie Callan, Zhiyuan Liu, and Russell Power. 2017. End-to-End Neural Ad-hoc Ranking with Kernel Pooling. In *Proc. of SIGIR*.

[17] Peilin Yang, Hui Fang, and Jimmy Lin. 2017. Anserini: Enabling the use of Lucene for information retrieval research. In *Proc. of SIGIR*.

[18] Hamed Zamani, Mostafa Dehghani, W Bruce Croft, Erik Learned-Miller, and Jaap Kamps. 2018. From Neural Re-Ranking to Neural Ranking: Learning a Sparse Representation for Inverted Indexing. In *In Proc. of CIKM*.