RUDI: Real-Time Learning to Update Dense Retrieval Indices

Sophia Althammer[1]

[1] *TU Vienna, Austria, Karlsplatz 13, Vienna, 1040, Austria*

Keywords
Dense retrieval, Real-time update, In-production systems

Dense retrieval models demonstrate great effectiveness gains for retrieval and re-ranking by learning a vector space embedding for the queries and the documents in the corpus [1, 2, 3, 4, 5, 6]. At the same time dense retrieval improves inference speed at query time with fast approximate nearest neighbor search [7, 8, 9] compared to exact k-nearest neighbor search by moving most of the computational effort to the indexing phase.

For search indices in production, there are continously millions of new data points which need to be included in the index in real-time [10, 11]. With the constantly new content to be indexed, the overall content of the whole corpus shifts gradually. With this it also shifts what should be considered relevant for a given query. For example the global COVID-19 pandemic resulted in an explosion of COVID-19 related websites, news articles and scientific publications [12]. In order to track, understand and seek this rapidly growing, novel information, the information retrieval community created a continously growing research dataset containing COVID-19 related publications [13].

To be able to find high-quality, relevant, and recent results, the novel content not only needs to be included in the search index, but the indexing model needs to account for the content shift and update the index in real-time. To incorporate this content shift in real-time in production systems, the dense retrieval model needs to be re-trained and the corpus re-indexed with the re-trained dense retrieval model. Real-time user interactions provide labels for re-training the dense retrieval model in real-time. However updating the search index in real-time remains an open challenge. In production systems the search indices have a size up to 100 millions of terabytes, thus re-indexing the whole corpus is computationally expensive and not feasible in real-time scenarios.

In this paper we propose the concept RUDI for Real-time learning to Update Dense retrieval Indices with simple transformations. In RUDI a computationally lightweight vector space transformation function $T : \mathbb{V} \to \mathbb{V}_r$ between the vector embedding space of the previous retrieval model $\mathbb{V}$ and of the re-trained dense retrieval model $\mathbb{V}_r$ is used to transform the vector embeddings of the previous index to the embeddings of the re-trained indexing model. The advantage of RUDI is that the index embedding does not need to be fully re-indexed with the re-trained dense retrieval model, but the index is updated with a learned, computationally lightweight transformation function. This allows updating the dense retrieval index in real-time.

First the dense retrieval model is re-trained in real-time with new labels accounting for the shift in the corpus. These new labels are determined by indexing the new content with the original retrieval model and getting implicit feedback through user interaction. Re-indexing the whole corpus with the re-trained dense retrieval model would give the vector embedding space of the re-trained dense retrieval model $\mathbb{V}_r$. To approximate the embeddings in $\mathbb{V}_r$, the transformation function T takes the embedding $v^d \in \mathbb{V}$ of the document d from the previous embedding space as input and outputs the approximated vector space embedding v_a^d. This vector v_a^d approximates the vector space embedding $v_r^d \in \mathbb{V}_r$ of document d of the re-trained dense retrieval model. The approximated vector space embedding of document d v_a^d is then the updated embedding of vector space $\mathbb{V}_r$. The transformation function T is learned in real-time on a small, sampled fraction $\mathbb{D}$ of the documents in the corpus. For these training documents $d \in \mathbb{D}$ the updated vector $v_r^d \in \mathbb{V}_r$ of the re-trained dense retrieval model is computed. Then the transformation function is trained on v^d and v_r^d with the objective of minimizing the distance between the approximate vector space embedding v_a^d and v_r^d

$$min \left\| v_a^d - v_r^d \right\| .$$

With this learned, lightweight transformation function the whole index can be updated in real-time while accounting for the temporal content shift in the corpus.

We plan to first analyze the shift of the vector embedding space between the previous and the re-trained dense

DESIRES 2021 – 2nd International Conference on Design of Experimental Search Information REtrieval Systems, September 15–18, 2021, Padua, Italy

sophia.althammer@tuwien.ac.at (S. Althammer)
https://sophiaalthammer.github.io/ (S. Althammer)

CEUR Workshop Proceedings (CEUR-WS.org)

retrieval model. Furthermore we plan study to what extent we can learn a simple, lightweight transformation function between the embedding space of the previous and the re-trained dense retrieval model. We investigate different transformation functions from one fully connected layer to exponential transformation functions and compare their approximation performance. Also we plan to investigate how the overall retrieval effectiveness is influenced by updating the retrieval index with RUDI compared to re-indexing the whole index.

As re-indexing the training documents $d \in \mathbb{D}$ for training the transformation function in real-time is computationally expensive, we plan to analyze the trade-off between number of training documents and overall retrieval quality on the updated index. Furthermore we investigate different sampling strategies for sampling the training documents from the overall index. We plan to compare random sampling with strategies aiming to sample documents from the index with maximal orthogonal embeddings. We plan to compare the effectiveness of the transformation functions trained with the different sampling strategies. Furthermore we plan to do speed comparisons between updating the dense retrieval index with different size of training samples for the transformation function and between re-indexing the whole index.

One could include additional features in the embedding space for hyperparameters like date or version, in order to include the recency of the results in the embedding space and make additional filter systems redundant.

Another open challenge is the evaluation of updated indices. As in the real-time scenario the query and content distribution gradually shifts, the evaluation with fixed test collections lacks to account for this shift. Therefore it is an interesting question how to evaluate an in-production system for example with A/B testing.

We conclude that our goal is to update dense retrieval indices in real-time while incorporating the temporal content shift. Therefore we propose RUDI for updating dense retrieval indices with transformations in real-time. We outline which research questions are necessary to investigate the effectiveness and efficiency of RUDI.

Acknowledgments

This work was supported by the EU Horizon 2020 ITN/ETN on Domain Specific Systems for Information Extraction and Retrieval (H2020-EU.1.3.1., ID: 860721).

References

[1] N. Reimers, I. Gurevych, Sentence-BERT: Sentence embeddings using Siamese BERT-networks, in: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), Association for Computational Linguistics, Hong Kong, China, 2019, pp. 3982–3992. URL: https://www.aclweb.org/anthology/D19-1410. doi:10.18653/v1/D19-1410.

[2] V. Karpukhin, B. Oguz, S. Min, P. Lewis, L. Wu, S. Edunov, D. Chen, W.-t. Yih, Dense passage retrieval for open-domain question answering, in: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), Association for Computational Linguistics, Online, 2020, pp. 6769–6781. URL: https://www.aclweb.org/anthology/2020.emnlp-main.550. doi:10.18653/v1/2020.emnlp-main.550.

[3] L. Xiong, C. Xiong, Y. Li, K.-F. Tang, J. Liu, P. N. Bennett, J. Ahmed, A. Overwijk, Approximate nearest neighbor negative contrastive learning for dense text retrieval, in: International Conference on Learning Representations, 2021. URL: https://openreview.net/forum?id=zeFrfgyZln.

[4] S. Hofstätter, S.-C. Lin, J.-H. Yang, J. Lin, A. Hanbury, Efficiently teaching an effective dense retriever with balanced topic aware sampling, 2021. arXiv:2104.06967.

[5] O. Khattab, M. Zaharia, Colbert: Efficient and effective passage search via contextualized late interaction over bert, in: Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '20, Association for Computing Machinery, New York, NY, USA, 2020, p. 39–48. URL: https://doi.org/10.1145/3397271.3401075. doi:10.1145/3397271.3401075.

[6] L. Gao, Z. Dai, T. Chen, Z. Fan, B. V. Durme, J. Callan, Complementing lexical retrieval with semantic residual embedding (2020). URL: http://arxiv.org/abs/2004.13969.

[7] J. Johnson, M. Douze, H. Jégou, Billion-scale similarity search with gpus, IEEE Transactions on Big Data (2019) 1–1. doi:10.1109/TBDATA.2019.2921572.

[8] R. Guo, P. Sun, E. Lindgren, Q. Geng, D. Simcha, F. Chern, S. Kumar, Accelerating large-scale inference with anisotropic vector quantization, in: H. D. III, A. Singh (Eds.), Proceedings of the 37th International Conference on Machine Learning, volume 119 of *Proceedings of Machine Learning Research*, PMLR, 2020, pp. 3887–3896. URL: http://proceedings.mlr.press/v119/guo20h.html.

[9] C. Fu, C. Xiang, C. Wang, D. Cai, Fast approximate nearest neighbor search with the navigating spreading-out graph, Proc. VLDB Endow. 12 (2019) 461–474. URL: https://doi.org/10.14778/3303753.3303754. doi:10.14778/3303753.

3303754.

[10] I. L. Stats, Total number of websites, https://www.internetlivestats.com/total-number-of-websites/, 2021. [Online; accessed 17-June-2021].

[11] S. Brin, L. Page, The anatomy of a large-scale hypertextual web search engine, Comput. Netw. ISDN Syst. 30 (1998) 107–117. URL: http://dx.doi.org/10.1016/S0169-7552(98)00110-X. doi:10.1016/S0169-7552(98)00110-X.

[12] H. Poon, Domain-specific language model pretraining for biomedical natural language processing, https://www.microsoft.com/en-us/research/blog/domain-specific-language-model-pretraining-for\-biomedical-natural-language-processing/, 2020. [Online; accessed 11-June-2021].

[13] L. L. Wang, K. Lo, Y. Chandrasekhar, R. Reas, J. Yang, D. Burdick, D. Eide, K. Funk, Y. Katsis, R. Kinney, Y. Li, Z. Liu, W. Merrill, P. Mooney, D. Murdick, D. Rishi, J. Sheehan, Z. Shen, B. Stilson, A. Wade, K. Wang, N. X. R. Wang, C. Wilhelm, B. Xie, D. Raymond, D. S. Weld, O. Etzioni, S. Kohlmeier, Cord-19: The covid-19 open research dataset, 2020. arXiv:2004.10706.