USEing Transfer Learning in Retrieval of Statistical Data

Anton Firsov
Knoema Corporation
afirsov@knoema.com

Vladimir Bugay
Knoema Corporation
vb@knoema.com

Anton Karpenko
Knoema Corporation
akarpenko@knoema.com

ABSTRACT

DSSM-like models showed good results in retrieval of short documents that semantically match the query. However, these models require large collections of click-through data that are not available in some domains. On the other hand, the recent advances in NLP demonstrated the possibility to fine-tune language models and models trained on one set of tasks to achieve a state of the art results on a multitude of other tasks or to get competitive results using much smaller training sets. Following this trend, we combined DSSM-like architecture with USE (Universal Sentence Encoder) and BERT (Bidirectional Encoder Representations from Transformers) models in order to be able to fine-tune them on a small amount of click-through data and use them for information retrieval. This approach allowed us to significantly improve our search engine for statistical data.

CCS CONCEPTS

• **Information systems → Information retrieval**; *Language models*; • **Computing methodologies → Transfer learning**.

KEYWORDS

information retrieval, transfer learning, language model

ACM Reference Format:
Anton Firsov, Vladimir Bugay, and Anton Karpenko. 2019. USEing Transfer Learning in Retrieval of Statistical Data. 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, 2 pages. https://doi.org/10.1145/3331184.3331427

1 COMPANY PORTRAIT

Knoema is a startup which provides access to the most comprehensive source of global decision-making data in the world. Our tools allow individuals and organizations to discover, visualize, model, and present their data and the world's data to facilitate better decisions and better outcomes.

Currently, our repository contains more than 2.4 billion data series organized into approximately 50 thousand datasets from more than a thousand sources such as the International Monetary Fund, Organization for Economic Co-operation and Development, World Bank, etc. and covers topics like economy, demographics, energy, education and so on.

At the heart of Knoema lies the search engine for data which is used by tens of thousand people around the world monthly.

2 INTRODUCTION

The search for statistical data is characterized by short documents (in our case, documents are time series names, for example, "United States - GDP") and availability of structure in the data. The shortness of documents makes the necessity of semantic matching even more important than it is in case of web searches, because in longer documents there is a higher probability that the same concept will be mentioned by different synonyms and the search engine will be able to match at least one of them.

Previously, we used an ontology to solve the problem of semantic matching [3]. However, there are two disadvantages to this approach. First, it requires a significant amount of manual labor. And, second, we don't always have access to the data that our users upload into their on-site data repositories, hence, we don't know concepts and synonyms that may be required for the search to work well.

The ideal solution would be to automatically infer semantic relations from the data itself or from the user interaction with the data. However, the experiments with DSSM model [4] didn't yield good results because of the comparatively small amount of click-through data (90K samples). Transfer learning turned up to be the answer to our problems: our experiments show that semantic relations can be inferred on much larger datasets and fine-tuned for our task.

3 MODEL

We experimented with generating embeddings for queries and time series names using two pre-trained models: BERT (Bidirectional Encoder Representations from Transformers)[2] and USE (Universal Sentence Encoder)[1]. Both of these models have different variations. For BERT we chose the base, uncased model for the English language. For USE we took transformer-based variation. As a measure of semantic "closeness", we used cosine similarity between vectors.

Even without fine-tuning USE shows surprisingly good results. In particular:

- It's able to resolve synonyms and synonymous phrases. For example, query "Number of people being killed in Canada" embedding is close to the embedding of time series "Canada - Homicide rate".
- In case of absent data it finds more general available data: "BMW theft in Japan" => "Japan - Car theft".
- It returns time series for semantically close countries depending on topic. For example, "US militarily spending" is nearby Afghanistan and Iraq military spending, and "US oil production" is near OPEC and EU oil production.

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

ACM ISBN 978-1-4503-6172-9/19/07…$15.00
https://doi.org/10.1145/3331184.3331427

- Queries in the form of questions are close to time series names with the answer. For example, "What is France GDP?" is close to "France - GDP".

However, it has difficulties with:

- Domains-specific terminology and abbreviations ("GDP", "CDR", "NRI", etc.).
- Time series names that contained auxiliary words ("France - Gross Domestic Product, Current Prices, National Currency" or "United Arab Emirates - Production of Crude Oil including Lease Condensate").
- Time series with other regions were closer than some of the more relevant results for the region in query.

Results with BERT without fine-tuning are much worse. Also, BERT is 3 times slower in embedding generation. In further experiments we tried out USE model first, because it looked more promising at this stage.

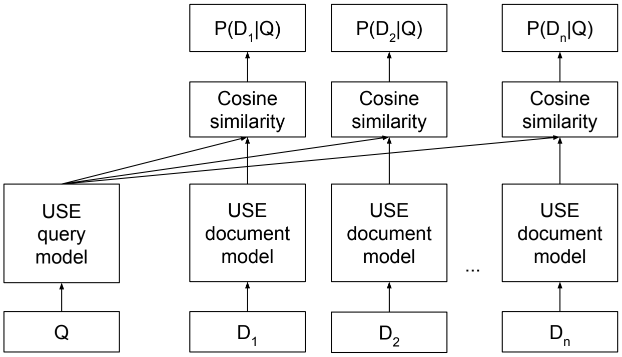

Figure 1: The network architecture for USE fine-tuning on click-though data.

To mitigate USE deficiencies we fine-tuned it on click-through data using DSSM-like architecture (Figure 1). We use two separate USE instances to get embeddings for queries $\overline{Q} = USE_Q(Q)$ and documents $\overline{D} = USE_D(D)$. The weights of this two instances are initialized from the original USE model in order to achieve transfer learning from the tasks used in pre-training.

The similarity between query Q and document D is measured by

$$S(Q,D) = \frac{\overline{Q}^T \overline{D}}{||\overline{Q}||||\overline{D}||} \tag{1}$$

The posterior probability of document D given query Q is calculated using softmax function:

$$P(D|Q) = \frac{exp(S(Q,D))}{\sum_{D' \in \mathbf{D}} exp(S(Q,D'))} \tag{2}$$

Where **D** should ideally be a set of all documents. However, it would be very inefficient to calculate. So, following [4], we approximate it by choosing 4 random unclicked documents (together denoted by D^-) for query Q in addition to the document that was clicked D^+.

The fine-tuning objective of the whole model is

$$L(W) = -\sum_{Q,D^+} log(P(D^+|Q)) \tag{3}$$

For BERT fine-tuning we used the same architecture with 2 variations in pooling strategy: sum of token embeddings returned by BERT and the embedding of the first token as the vector corresponding to a query or a document.

4 IMPLEMENTATION DETAILS

We created a training set of 12704 query - clicked document pairs. To decrease a fraction of accidentally clicked documents we included in the training set only documents that were clicked at least 2 times for a given query.

We fine-tuned the described models for 5 epochs using Adam optimizer with a learning rate of 10^{-5}, a batch size equal to 128 (32 for BERT), and 4 random negative samples per one clicked document.

After that, we generated vectors for 400 million time series using fine-tuned document model. As exhaustive k-nearest neighbors search is very slow for the such number of vectors, we used FAISS library [1] [5] to create IVFPQ index with 262144 centroids and HNSW quantizer for a fast *approximate* search of k-nearest neighbors.

5 RESULTS

The fine-tuned USE model preserved the above mentioned positive qualities that the original USE has. The detrimental effect of the auxiliary words in time series names was almost completely negated. Relevant time series with regions from query became closer to the query than time series with semantically close regions. The improvement with domain-specific terminology depends on specific terms and abbreviations and was proportional to a number of occurrences of them in the training set.

The A/B test showed that users clicked results returned by USE model 18% more frequently than results returned by our original search engine.

BERT results with first token pooling strategy were rather poor. The sum pooling gave results comparable to USE model, however, much faster vector calculations make USE model more preferable.

In conclusion, we demonstrated that IR can be significantly improved by transfer learning from deep language models and other NLP tasks.

REFERENCES

[1] Daniel Cer, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St. John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, Yun-Hsuan Sung, Brian Strope, and Ray Kurzweil. 2018. Universal Sentence Encoder. *CoRR* abs/1803.11175 (2018). arXiv:1803.11175 http://arxiv.org/abs/1803.11175

[2] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *arXiv preprint arXiv:1810.04805* (2018).

[3] Anton Firsov. 2017. Traditional IR Meets Ontology Engineering in Search for Data. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '17)*. ACM, New York, NY, USA, 1355–1355. https://doi.org/10.1145/3077136.3096473

[4] Po-Sen Huang, Xiaodong He, Jianfeng Gao, Li Deng, Alex Acero, and Larry Heck. 2013. Learning deep structured semantic models for web search using clickthrough data. In *Proceedings of the 22nd ACM international conference on Conference on information &#38; knowledge management (CIKM '13)*. ACM, New York, NY, USA, 2333–2338. https://doi.org/10.1145/2505515.2505665

[5] Jeff Johnson, Matthijs Douze, and Hervé Jégou. 2017. Billion-scale similarity search with GPUs. *arXiv preprint arXiv:1702.08734* (2017).

[1] https://github.com/facebookresearch/faiss