Semantic search with domain-specific word-embedding and production monitoring in Fintech

Mojtaba Farmanbar, Nikki van Ommeren, Boyang Zhao
ING
Amsterdam, The Netherlands
{mojtaba.farmanbar, nikki.van.ommeren, boyang.zhao}@ing.com

Abstract

We present an end-to-end information retrieval system with domain-specific custom language models for accurate search terms expansion. The text mining pipeline tackles several challenges faced in an industry-setting, including multi-lingual jargon-rich unstructured text and privacy compliance. Combined with a novel statistical approach for word embedding evaluations, the models can be monitored in a production setting. Our approach is used in the real world in risk management in the financial sector and has wide applicability to other domains.

1 Introduction

Many industry-grade search engines exist (e.g. Elasticsearch) as general-purpose systems. While they have a general understanding of languages and includes search algorithms for document retrieval, their accuracy can be sub-optimal. This accuracy ultimately depends on domain specificity and the terms provided by the users.

This issue is particularly apparent in domains such as risk management. A critical function of the risk officers is knowing the relevant set of search terms to retrieve documents related to a specific topic. However, constructing a search criteria with all the appropriate search terms/phrases and their logical relations in the search engine remains complex, and can be prune to false-negatives and false-positives. While entity set expansion systems exist (among the earliest includes Google Sets), they are mostly using general-purpose language models (Zhang et al., 2020; Mamou et al., 2018) and thus not domain-specific.

In addition, building such an industry-grade domain-specific semantic search engine is challenging as this involves several additional considerations beyond simply an accurate language model. This includes: 1) extensive use of jargons, abbreviations, and unstructured data, 2) multiple languages relating to the same topic, 3) privacy compliance, and 4) performance monitoring of models on production.

In this paper, we describe an end-to-end information retrieval (IR) system with custom embeddings to address the aforementioned challenges. The system utilizes multiple advancements in Natural Language Processing (NLP) to process anonymized banking data. Capitalizing on our custom language models (in multiple languages and with multiple *n*-grams), we enable the automatic suggestion of highly relevant similar keywords, to ease the burden of requiring the user to build complex search queries. Additional pre-trained contextual embeddings are utilized for ranking by relevance. Lastly, we develop novel statistical methodologies for monitoring the stability of our language models on production.

2 Approach

We develop an end-to-end IR system, as depicted in Figure 1. The system, implemented in python, consists of 1) a text mining pipeline that preprocesses the records for indexing and for building language models, and 2) a front-end where the user queries the database with one or more search terms. The search terms are processed the same way by the text mining pipeline as for the records, and are further expanded to include similar words based on the trained domain-specific language models. The similar

Proceedings of the 27th International Conference on Computational Linguistics, pages 28–33
Barcelona, Spain (Online), December 12, 2020.

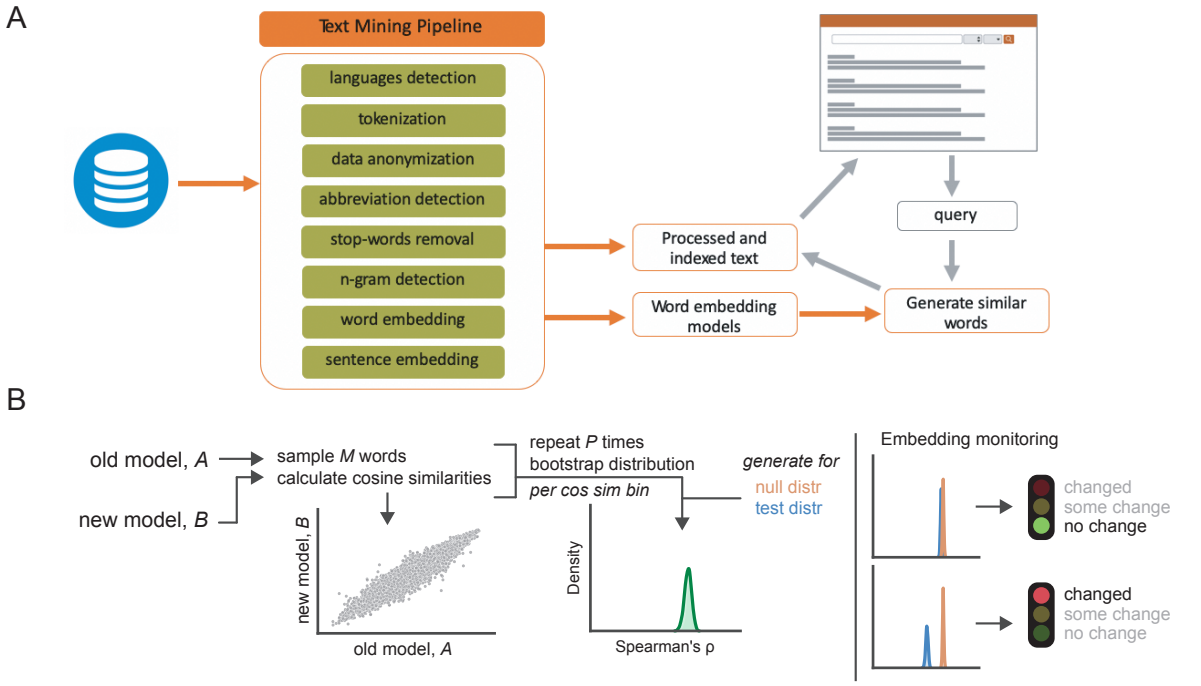

Figure 1: (A) IR system overview. Orange arrows indicate the preprocessing steps. Grey arrows indicate processing during real-time user queries. (B) Statistical method for monitoring word embeddings.

word expansion is thus a corpus-based approach and is based on the premise that similar words are distributionally related in similar contexts (known as the distributional hypothesis (Harris, 1954)).

2.1 Text mining pipeline

The text mining pipeline consists of several key components (Figure 1A). First, we integrate records from different data sources and unify them. As the records can be in different languages, we automatically detect languages based on specific patterns and with *spaCy*[1] so the appropriate language models can be used. Sensitive and identifiable information (e.g. names, phone numbers, employee identification numbers, etc) is retrieved and removed from the records. Abbreviations are extracted using the Shwartz-Hearst method (Schwartz and Hearst, 2003). The text is then cleaned using a combination of custom regular expressions and removal of stop-words and punctuations. We derive *n*-grams using *Gensim*[2] based on a PMI-like scoring method (Mikolov et al., 2013b). From the processed data, uni-, bi-, and tri-grams are jointly trained in deriving our custom word embeddings. Previous work suggests that multi-gram can improve on the quality of the obtained word embeddings (Gupta et al., 2019). In addition to word embeddings, we also derive document embeddings based on the cleaned text using pre-trained contextual multilingual universal sentence encoders. The embeddings are indexed for later ranking by relevance of the returning documents using FAISS (Johnson et al., 2017). The outputs of this pipeline is structured/processed data (along with the indexed version) and the trained word embeddings, which are used by our search engine in the front-end for query expansions.

2.2 Word embedding monitoring on production

We intend to monitor the stability of the word embeddings on production (Figure 1B). We capitalize on the intrinsic variability of the model from run to run (on the same data) to estimate its background variability. We can then determine if the extra variability between an old vs new model is different in comparison to this background variability. More specifically, for two given embeddings *A* and *B*, we first sample *M* words common to both embeddings. We derive *M-1* cosine similarity values (e.g. between

[1]https://spacy.io/
[2]https://radimrehurek.com/gensim/

Pre-trained general-purpose		Custom domain-specific
Word2Vec (GoogleNews)	GloVe (Wikipedia)	Word2Vec (domain-specific)
(confidentiality, 0.65)	(policy, 0.62)	(confidentiality, 0.82)
(security, 0.61)	(confidentiality, 0.56)	(datum_protection, 0.81)
(secrecy, 0.60)	(disclaimer, 0.54)	(disclosure, 0.73)
(anonymity, 0.59)	(policies, 0.51)	(banking_secrecy, 0.72)
(rights, 0.59)	(security, 0.49)	(bank_secrecy, 0.69)
(protection, 0.58)	(rights, 0.47)	(confidential_information, 0.64)
(oversight, 0.57)	(disclosure, 0.47)	(confidential, 0.63)
(identity, 0.57)	(reserved, 0.46)	(personal_datum, 0.63)

Table 1: Pre-trained and custom language models with similar terms retrieved for the word *privacy*. Cosine similarity values are shown in parentheses.

1st and 2nd word, 2nd and 3rd word, etc) based on A and again on B. We calculate the Spearman's correlation (ρ) of the cosine similarity values between A and B, binned based on A into 10 bins (from -1 to 1). This procedure is repeated P times to generate a bootstrapped distribution of ρ values (per bin). A Gaussian kernel density estimation is then fitted to each distribution. We perform this bootstrapping method on several pairs of embeddings. For embeddings generated from different runs on the same data, the resulting distribution constitutes the null distribution. For embeddings generated from old and new input data (during monitoring), the resulting distribution constitutes the test distribution. Comparison between the two distributions, as assessed by Jensen-Shannon divergence or Kolmogorov-Smirnov test statistic and with pre-defined thresholds enable monitoring of any substantial changes to the embeddings.

3 Experiments

We collect around 160K records in the financial sector. To compare across the different word embeddings, we use both pre-trained and custom-trained models. Pre-trained models have been previously trained on a large corpus such as the Wikipedia, Common Crawl, or Google News. Popular models include ones based on Word2Vec (Mikolov et al., 2013a) or GloVe (Pennington et al., 2014). While using these general-purpose language models saves training time and the need for custom preprocessing of data, they potentially lack domain-specific word semantics. Therefore, we also train custom Word2Vec Skip-gram word embeddings based on our datasets for English, French, and Dutch languages. We evaluate our models directly with the anonymized data as anonymization is required for compliance and regulatory reasons. For extrinsic word embedding evaluations, we build a gated recurrent unit (GRU)-based model with an embedding layer, followed by a 64-unit GRU, a 16-unit fully connected layer with ReLU activation, and an output layer with sigmoid activation. The model is trained with Adam optimizer and binary cross entropy loss function.

We also evaluate the quality of the information retrieval, viz. the precision achieved by relevance ranking based on the different sentence embedding models. Several models are examined, including TF-IDF, average of the Word2Vec (Google News), Word2Vec (domain-specific), or BERT (Devlin et al., 2019), CLS token of BERT, sentence BERT (sBERT) (Reimers and Gurevych, 2019), LASER (Artetxe and Schwenk, 2019), and different versions of universal sentence encoders (USE) (Cer et al., 2018) and multilingual universal sentence encoders (MUSE) (Yang et al., 2020). With the exception of TF-IDF and domain-specific Word2Vec, the other models are all taken as pre-trained models.

4 Results

4.1 Word embedding evaluations

We first want to evaluate if our custom domain-specific language models embed different word semantics compared to the pre-trained general-purpose models. When we examine for example the word *privacy*,

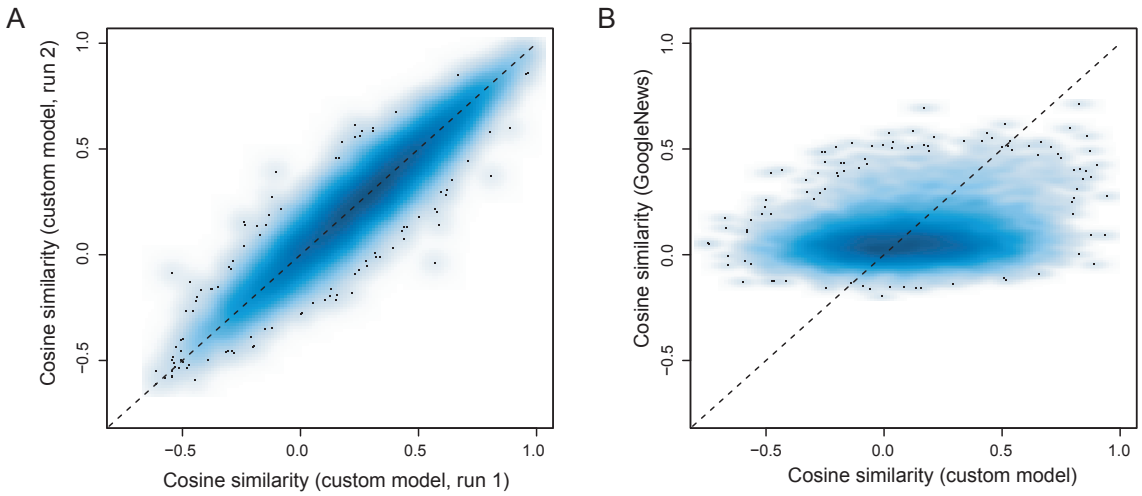

Figure 2: Sampling of 15,000 words from different embeddings, to show that the custom models retrieve relations that are vastly different from that compared to GoogleNews word relations.

we observe that GoogleNews- and Wikipedia-based embeddings associate *privacy* with many words that tend to have different semantics (Table 1). However, the custom word embeddings noticeably give a more consistent set of similar words and with higher similarity scores. Furthermore, the similar terms retrieved contain word phrases (bigrams) instead of just unigrams, and in many cases contain the word *bank*. As *privacy* is much more prevalent as a topic in our dataset related to the banking sector, these observations support a more relevant retrieval of similar words.

To assess whether the word relations are globally different between these models, we sample a large number of word pairs common in both custom and GoogleNews Word2Vec models, and evaluate their cosine similarity values based on either models. While we observe that the custom models from run to run maintain high concordance among the word relations (Figure 2A), they are vastly different from those based on GoogleNews models (Figure 2B).

In addition to our custom word embeddings encoding domain-specific word relations, we want to ensure they are good language models for other tasks. We perform extrinsic evaluations on the embeddings by building classifiers to assess whether the most common risk types can be predicted. Based on GRU-based neural network models, we observe that our custom embedding perform just as well as the much larger time-intensive pre-trained GoogleNews embedding (weighted F1 scores of 0.8 and 0.78, respectively). Therefore, our custom embedding encode language semantics on-par with pre-existing models, while retain domain-specificity for similar words extraction.

4.2 Word embedding monitoring

Our custom Word2Vec embeddings are rebuilt periodically with updated records to ensure the model is up-to-date and continues to capture the relevant vocabularies and semantics. For monitoring on production, we apply a statistical approach that measures the word relations and

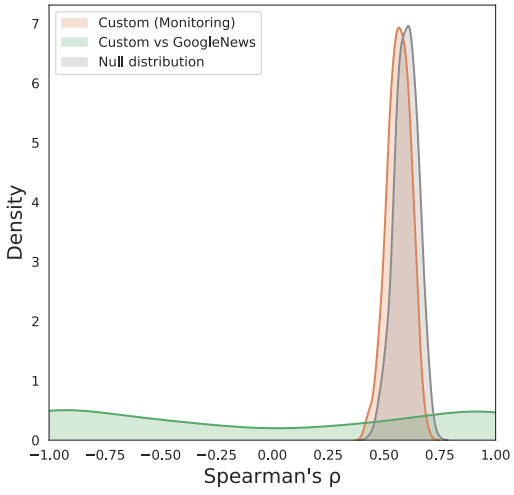

Figure 3: Monitoring word embeddings on production - illustrating that there are no substantial changes to the custom models being monitored.

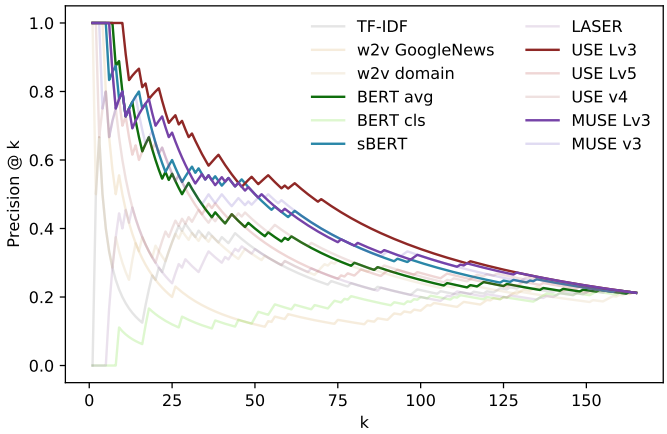

Figure 4: Precision at k for the query *statistical model*, with selected models highlighted.

Model	Average precision
TF-IDF	0.29
Word2Vec (GoolgeNews)	0.29
Word2Vec (domain-specific)	0.33
BERT (average)	0.53
BERT (CLS)	0.18
sBERT	0.61
LASER	0.28
USE Lv3	0.75
USE Lv5	0.51
USE v4	0.44
MUSE Lv3	0.63
MUSE v3	0.58

Table 2: Average precision for relevance ranking based on sentence embedding models for the query *statistical model*.

their distributional differences between a given and reference embeddings. We observe that the variations in the updated embeddings are not substantially different from the null distribution based on intrinsic stochasticity on the same dataset (Figure 3). When we compare to a substantially different word embedding (i.e. Word2Vec based on GoogleNews), we observe a dramatic change and shift in the distribution. Quantitatively, relative to the null distribution, we observe the updated custom Word2vec embeddings and GoogleNews Word2vec have Jensen-Shannon divergence of 0.22 and 0.70, respectively. We evaluate this for the embeddings of all different languages and *n*-grams, and set empirical thresholds for triggers on production.

4.3 Information retrieval evaluations

We assess the ranking quality of the returning documents across different sentence embedding models and queries. Using the query *statistical model* as example, we observe that the performance vastly differs (Figure 4 and Table 2). TF-IDF, average of word embeddings, and LASER models are unable to understand context, resulting in many highly ranked irrelevant documents that mention the word *model*, but in other contexts (e.g. business model, device model, car model, etc). CLS token of BERT is also found to be ineffective. BERT-based models are too slow for practical usage. Most strikingly, pre-trained universal sentence encoders (USE and MUSE) achieve higher precision, albeit variability in performance are observed among the various versions - likely due to difference in datasets and tasks used for training. We also finetune the USE/MUSE models for next sentence prediction, but no difference in performance is observed. These, among other examples, support the use of USE/MUSE for document embedding and relevance ranking in our system.

5 Conclusions

We present an end-to-end IR system used in production as a semantic search engine with intelligent keyword expansion and continual model monitoring. It addresses several challenges in handling multilingual domain-specific unstructured text and privacy compliance, and simplifying the derivation of keywords within a semantic class. Our experiments show that the custom word embeddings are distinct from general-purpose models, can present more relevant search terms, and can be monitored on production based on a novel statistical approach. We also show that different sentence embeddings differ vastly in performance and some, based on experimentation, can be used to improve relevance ranking. Overall our proposed solution is applicable for use in other domains.

Acknowledgements

We thank the anonymous reviewers for their feedback and colleagues at ING for helpful discussions.

References

Mikel Artetxe and Holger Schwenk. 2019. Massively multilingual sentence embeddings for zero-shot cross-lingual transfer and beyond. *Transactions of the Association for Computational Linguistics*, 7:597–610, mar.

Daniel Cer, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St. John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, Brian Strope, and Ray Kurzweil. 2018. Universal sentence encoder for English. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 169–174, Brussels, Belgium, November. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota, June. Association for Computational Linguistics.

Prakhar Gupta, Matteo Pagliardini, and Martin Jaggi. 2019. Better word embeddings by disentangling contextual n-gram information. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 933–939, Minneapolis, Minnesota, June. Association for Computational Linguistics.

Zellig S. Harris. 1954. Distributional structure. *Word*, 10(2-3):146–162.

Jeff Johnson, Matthijs Douze, and Hervé Jégou. 2017. Billion-scale similarity search with gpus. *arXiv preprint arXiv:1702.08734*.

Jonathan Mamou, Oren Pereg, Moshe Wasserblat, Ido Dagan, Yoav Goldberg, Alon Eirew, Yael Green, Shira Guskin, Peter Izsak, and Daniel Korat. 2018. Term set expansion based on multi-context term embeddings: an end-to-end workflow. In *Proceedings of the 27th International Conference on Computational Linguistics: System Demonstrations*.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013a. Efficient estimation of word representations in vector space. In *Proceedings of Workshop at ICLR*.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013b. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.

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)*, pages 1532–1543, Doha, Qatar, October. Association for Computational Linguistics.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 11.

Ariel Schwartz and Marti Hearst. 2003. A simple algorithm for identifying abbreviation definitions in biomedical text. *Pacific Symposium on Biocomputing*, 4:451–62, 02.

Yinfei Yang, Daniel Cer, Amin Ahmad, Mandy Guo, Jax Law, Noah Constant, Gustavo Hernandez Abrego, Steve Yuan, Chris Tar, Yun-hsuan Sung, Brian Strope, and Ray Kurzweil. 2020. Multilingual universal sentence encoder for semantic retrieval. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 87–94, Online, July. Association for Computational Linguistics.

Yunyi Zhang, Jiaming Shen, Jingbo Shang, and Jiawei Han. 2020. Empower entity set expansion via language model probing. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8151–8160.