Text Mining for Information Retrieval on WikIR

Project for Machine Learning course

Jakub Bartczuk

Winter 2020

1 Motivation

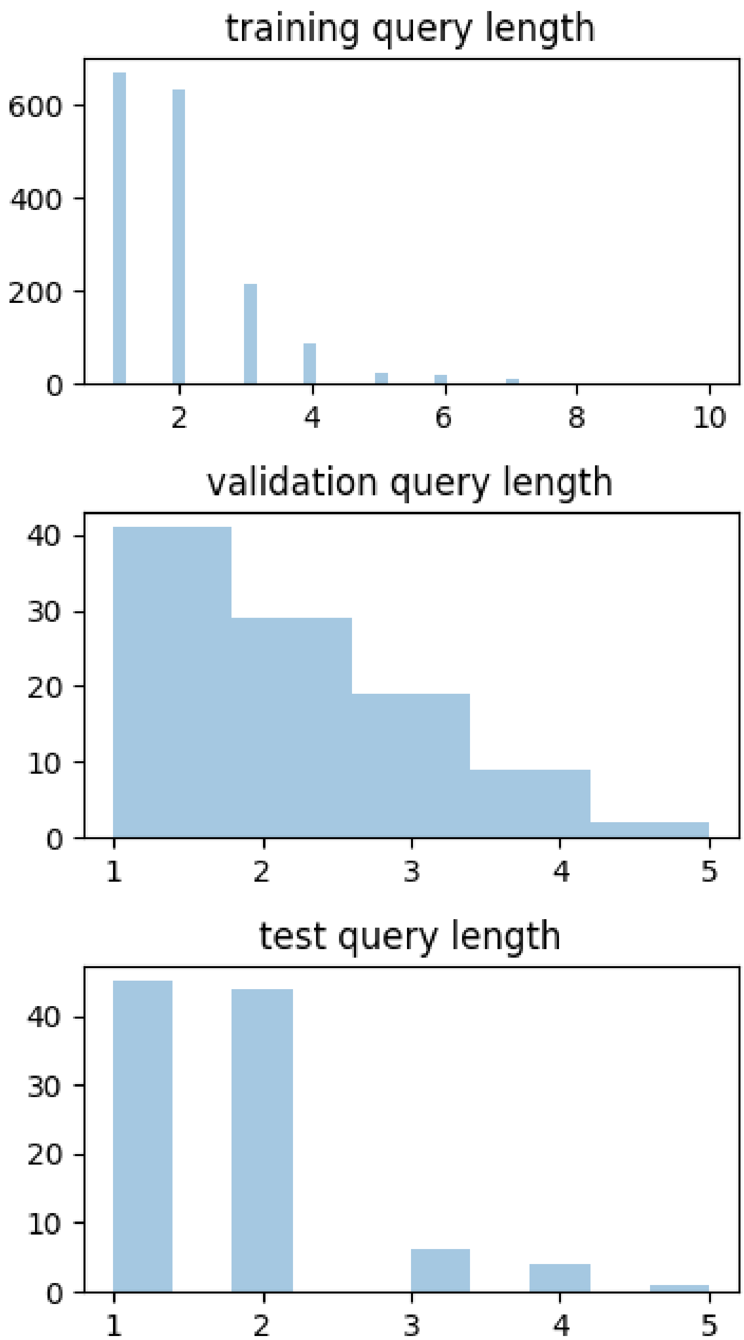
One of the first things that data scientists do when facing a new problem is searching for related problems and software on GitHub. GitHub's search capabilities are very limited (at the time of writing this report) - there are several features like project descriptions, tags and readmes that could be helpful, but its search engine doesn't allow for searching them simultaneously. The effect is that sometimes reformulating query might result in drastically different search results. This project aims at evaluating in a controlled experiment several simple text mining methods that can be useful for expanding traditional bag-of-words retrieval model.

2 Abstract

WikIR [2] is a recently proposed dataset for evaluating Information Retrieval. So far it was used only to benchmark standard Information Retrieval method (BM25)[5] versus deep learning-based text matching. In this project several machine learning methods for text, using classification, word embeddings and matrix decomposition were used to improve searching results.

3 Dataset

The dataset consists of approx. 100k documents and queries split into training, validation and test sets (with 1k, 100 and 100 examples respectively). It is created from wikipedia articles and their titles: queries are the titles, whereas rest of article (with first paragraph removed) gets split into 'documents' to be retrieved. Document is deemed relevant for query if either the query is the title of document's article (relevance score 2) or its article is linked in retrieved document's article. Below query lengths are reported. Document lengths are less interesting, since author's method splits articles into documents of approx. 200 words.


training query length
0
200
400
600
2
4
6
8
10
validation query length
0
10
20
30
40
1
2
3
4
5
test query length
0
10
20
30
40
1
2
3
4
5

4 Evaluation

All methods apart from query expansion-based retriever first retrieve 100 records using BM25 relevance model and then perform reranking.

Results are evaluated using standard Information Retrieval metrics: Precision at k, nDCG at k and Mean Average Precision.

5 Data preparation

For our models we preprocess data according to original paper: english stop words are removed and the words are stemmed using Porter stemmer. These steps are performed using NLTK [4].

Models were tested with both stemmed and unstemmed texts. Also, keyword extraction was used as a step for some word embedding models (to alleviate problem of averaging larege documents). Keyword extraction was performed using gensim's [7] textrank model.

6 Models

1. Classification on BM25 features and tf-idf features [3]

2. Classification on using GloVe [6] and FastText [1] word embeddings

3. Scoring using similarity based on unsupervised representation (using Non-negative Matrix Factorization)

4. Query expansion using nearest words using word embeddings.

   Models 1-3 work by combining their score with relevance score. BM25 model is first used to retrieve 100 top documents. Model score results from running classification: top k are defined as positive and bottom k documents are defined as negative.

7 Results

8 Deliverables

Project's code can be found on its GitHub page[1].

9 Conclusion

BM25 already provides strong baseline for search results ranking. The difference noted in Lin's paper (metod beating BM25 by 2% MAP) doesn't happen for

[1] https://github.com/lambdaofgod/wikir$_{t}ext_{m}ining$

retriever_type	parameters	P_10	P_20	P_5	map	ndcg_cut_10	ndcg_cut_20	ndcg_cut_5
baseline	[retriever_type=baseline]	0.238	0.1720	0.322	0.198483	0.386128	0.394731	0.424872
classifier_retriever	[alpha=0.3, classification_scored_documents=20, retriever_type=classifier_retriever, text_col=keywords]	0.240	0.1700	0.328	0.198906	0.388802	0.394965	0.430794
query_expander	[retriever_type=query_expander]	0.126	0.0960	0.140	0.096759	0.197082	0.213800	0.195086
topic_model_retriever	[alpha=0.1, text_col=stemmed_text, topic_modeler_n_components=100]	0.206	0.1555	0.290	0.178164	0.348725	0.361182	0.390542
word_embedding_classifier_retriever	[alpha=0.4, retriever_type=word_embedding_classifier_retriever, text_col=keywords]	0.242	0.1660	0.330	0.206379	0.394558	0.393970	0.433004

WikIR, most likely because of the fact that WikIR queries are very short so retrieval is almost 'all or nothing'. It remains an open question whether the results can be improved by using simple machine learning models on top of features extracted using pretrained language models.

10 Further steps

- use word embedding features for global search - there are methods for building large scale approximate kNNs. These might alleviate query vocabulary mismatch problem.
- use language-model based features
- use Learning to Rank approach
- use dataset construction method on another source

References

[1] Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. Enriching word vectors with subword information. *arXiv preprint arXiv:1607.04606*, 2016.

[2] Jibril Frej, Didier Schwab, and Jean-Pierre Chevallet. Wikir: A python toolkit for building a large-scale wikipedia-based english information retrieval dataset, 2019.

[3] Jimmy Lin. The simplest thing that can possibly work: Pseudo-relevance feedback using text classification. *CoRR*, abs/1904.08861, 2019.

[4] Edward Loper and Steven Bird. Nltk: The natural language toolkit. In *In Proceedings of the ACL Workshop on Effective Tools and Methodologies for Teaching Natural Language Processing and Computational Linguistics. Philadelphia: Association for Computational Linguistics*, 2002.

[5] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. *Introduction to Information Retrieval.* Cambridge University Press, Cambridge, UK, 2008.

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

[7] Radim Řehůřek and Petr Sojka. Software Framework for Topic Modelling with Large Corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, pages 45–50, Valletta, Malta, May 2010. ELRA. `http://is.muni.cz/publication/884893/en`.