Query Expansion Based on NLP and Word Embeddings

Billel Aklouche[1,2,3], Ibrahim Bounhas[1,2,4] and Yahya Slimani[1,2,5]

[1] JARIR: Joint group for Artificial Reasoning and Information Retrieval (www.jarir.tn)
[2] Laboratory of Computer Science for Industrial System (LISI), Carthage University, Tunisia
[3] National School of Computer Science (ENSI), La Manouba University, Tunisia
billel.aklouche@ensi-uma.tn
[4] Higher Institute of Documentation (ISD), La Manouba University, Tunisia
bounhas.ibrahim@gmail.com
[5] Higher Institute of Multimedia Arts of Manouba (ISAMM), La Manouba University, Tunisia
yahya.slimani@gmail.com

Abstract. Query Expansion is an important process in information retrieval, which consists in adding new related terms to the original query in order to better identify relevant documents. In this paper, we discuss the participation of the JARIR research group to the TREC 2018 Common Core Track. We present different Query Expansion methods, which are based on Natural Language Pre-processing (NLP) tools and Word2Vec embedding models. Using the title of TREC topics, we select semantically related terms to the query. Our approach is composed of four steps: (1) Data Pre-processing, (2) Model Training, (3) Query Expansion and (4) Documents Ranking. For our best runs, results show that most of our topics scores are above the published median scores with some topics having the best scores.

Keywords: Ad Hoc Information Retrieval, Query Expansion, NLP Tools, Word2Vec, BM25.

1 Introduction

The TREC 2018 Common Core Track[1] is a traditional ad hoc retrieval task with the aim of exploring new collection construction methodologies and providing the IR community with a solid test collection. The track provides participants with a new dataset composed of the Washington Post Corpus and 50 topics. The TREC Washington Post Corpus contains news articles and blog posts written and published by Washington Post[2] from January 2012 through August 2017. Two sets of topics are provided: (a) 25 topics from the 2017 Common Core Track and (b) 25 new topics developed by the NIST assessors for this track. Each team could submit up to 10 runs total, a run contains a ranked list of top *n* documents retrieved for each topic. The track received a total of 72 runs submitted by 12 participating teams. Across all 72 runs, 7 were manual runs,

[1] https://trec-core.github.io/2018/
[2] https://www.washingtonpost.com/

31 used existing relevance judgments if available and 34 were automatic runs. The track 50 topics are judged by the NIST assessors.

Our participation consists in using Query Expansion in Information Retrieval. Query Expansion refers to the techniques that expand search terms to include other related terms that enhance original queries, in order to provide richer and more relevant results. For several years, great effort has been devoted to the development of new Query Expansion approaches [1]. In this paper, we present different Query Expansion methods that are based on NLP tools and Word2Vec embedding models [5]. Indeed, there are two models in Word2Vec for producing word vectors: Skip-Gram and Continuous Bag Of Words (CBOW). We compare and investigate the use of both models in obtaining new semantically related terms to the original queries. The two models were trained on the TREC Washington Post Corpus. In all proposed methods, a common pre-processing step is applied before getting in the training step. In addition, we investigate the impact of query reweighting and terms selection strategy on retrieval effectiveness.

The remainder of the paper is organized as follows. Section 2 outlines our approach and experiments. We evaluate and discuss our results in Section 3. Finally, Section 4 concludes the work.

2 Approach And Experiments

Our submission to the TREC 2018 Common Core Track consists of six automatic runs. Using only the title of TREC topics, which contains few keywords, our goal is to investigate a real IR scenario where no prior judgment is available and no manual intervention is performed. Also, in accordance with our goal, we aimed to present a Query Expansion approach that combines the advantages of NLP techniques and Word2Vec embedding models, in order to get an efficient retrieval system.

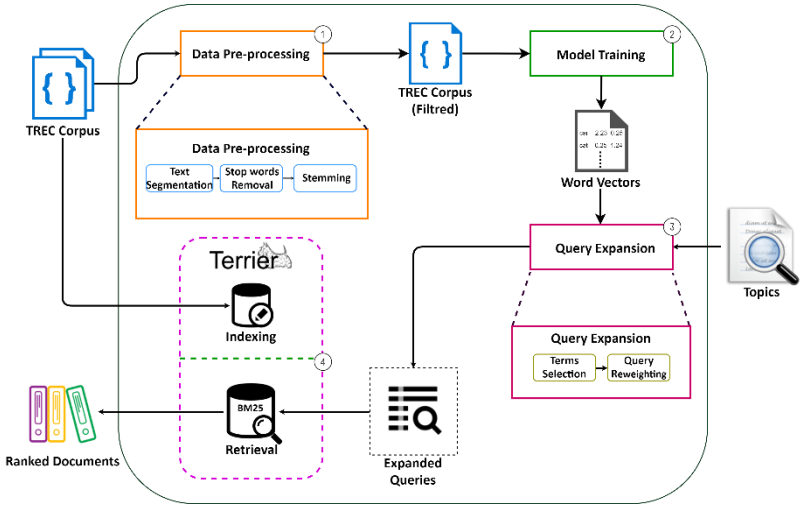

Fig. 1. General architecture of the proposed approach

Figure 1 illustrates the general architecture of the proposed approach. First, we exploit NLP techniques to extract a clean textual content from the TREC Washington Post Corpus. The filtered corpus is then used to train Word2Vec Skip-Gram or CBOW models; the training step consists in learning word vector representations that are good at capturing semantic relations between terms [5]. Afterward, the word vectors of the trained model are used to select top *n* similar terms to the original TREC topic titles; the new terms are selected based on their similarity to the entire query or their similarity to its individual terms. Furthermore, a term reweighting might be performed in order to update the new query. Finally, the new expanded queries are used to retrieve a set of top ranked 10,000 documents for each topic according to the state-of-the-art weighting scheme Okapi BM25 [8]. Each step of our approach is described in more details in the following subsections.

2.1 Data Pre-processing

Data pre-processing is an important and necessary phase before dealing with text corpora. We need to convert the corpus into a clean and consistent format, which is easier to process. To this end, we used the Python library NLTK[3] for all the pre-processing steps:

- **Text segmentation:** we used NLTK's Tokenizers to split text into sentences and then split sentences into individual tokens.
- **Stop words removal:** in this step, all non-informative words are filtered out. We used a stop word list composed from Terrier and NLTK.
- **Stemming:** we used Porter Stemmer [7], which is implemented in NLTK to reduce all terms to their stems. In addition, all terms are lowercased.

2.2 Model Training

Word embeddings refers to techniques that are used to produce vector representations of words [4]. In this work, we used Word2Vec for computing word embeddings. Word2Vec can use either Skip-Gram or CBOW model to construct word vectors. Within a surrounding window, the Skip-Gram model predicts the context words of a given target word, while the CBOW model predicts a target word based on context words [5]. We used the filtered TREC Washington Post Corpus resulting from the previous step to train the models. The training parameters were set as follows. We used a window size equal to 5 and we set the vectors dimension to 300. We also removed any words that appear less than 5 times in the whole corpus.

[3] Natural Language ToolKit (https://www.nltk.org/)

2.3 Query Expansion

In this step, word vectors are used to find semantically related terms to the query. To this end, we used Euclidean Distance to calculate the similarity score between two vectors. Thus, an exhaustive search is performed to find all possible candidate vectors. As mentioned before in this section, expansion terms are either similar to the entire query or to its individual terms. Consequently, the target vector is the result of addition of vectors associated with original query terms in the first case, while in the second case, the target vector is the vector associated with an individual term. Afterward, the candidate terms are sorted in a decreasing order according to their similarity scores. The top *n* terms are chosen as the final expansion terms where *n* is the number of the original query terms. Moreover, a reweighting step might be applied to assign weights to the new query terms. To this end, we chose to attribute a weight of *w=1* to the original query terms and a weight of *w=0.5* to the new expansion terms. The new expanded queries are then used in documents retrieval and ranking step.

2.4 Documents Ranking

We used Terrier[4], a flexible and a high performance scalable Information Retrieval platform [6], to index the collection and form our runs. We opted to use the well-established Okapi BM25 probabilistic weighting scheme with default parameters in order to perform the documents ranking. For each run, the top ranked 10,000 documents retrieved for each topic were submitted, in descending order by score.

In Table 1, we summarize the method adopted for each submission.

Table 1. Summary of the six submitted runs to TREC

Run Tag	Data Pre-Processing	Word2Vec Model	Expansion Terms Similarity	Query Re-weighting
jarir_skipgram	✓	Skip-Gram	Whole query	No
jarir_cbow	✓	CBOW	Whole query	No
jarir_sg_re	✓	Skip-Gram	Whole query	Yes
jarir_cb_re	✓	CBOW	Whole query	Yes
jarir_sg_ind	✓	Skip-Gram	Individual terms	No
jarir_cb_ind	✓	CBOW	Individual terms	No

3 Results and Discussion

As mentioned before, the TREC 2018 Common Core Track received a total of 72 runs from 12 participating teams. Across all 72 runs, 7 were manual runs, 31 used existing relevance judgments if available and 34 were automatic runs. The judgment sets for the

[4] http://terrier.org/

50 topics were created by NIST assessors. We submitted 6 automatic runs for this year's track.

Table 2 demonstrates the mean performance of each run over the 50 NIST topics for the evaluation metrics MAP, NDCG and P@10 [3]. To position our results with results of other teams, we included the average of the "Median" scores calculated across all the submitted automatic runs.

Table 2. Overall Performance of submitted runs, compared to the median results of all automatic runs

Run Tag	MAP	NDCG	P@10
jarir_skipgram	0.1769	0.4576	0.2980
jarir_cbow	0.1289	0.3841	0.2460
jarir_sg_re	**0.2040**	**0.4861**	**0.3700**
jarir_cb_re	0.1896	0.4595	0.3200
jarir_sg_ind	0.1471	0.4130	0.2720
jarir_cb_ind	0.1317	0.3868	0.2600
TREC_Median	0.1744	0.4482	0,3340

Half of our runs reached a score above the TREC median according to all measures. Figure 2 illustrates the mean performance of our best run *jarir_sg_re* measured by MAP compared with *TREC_Median*.

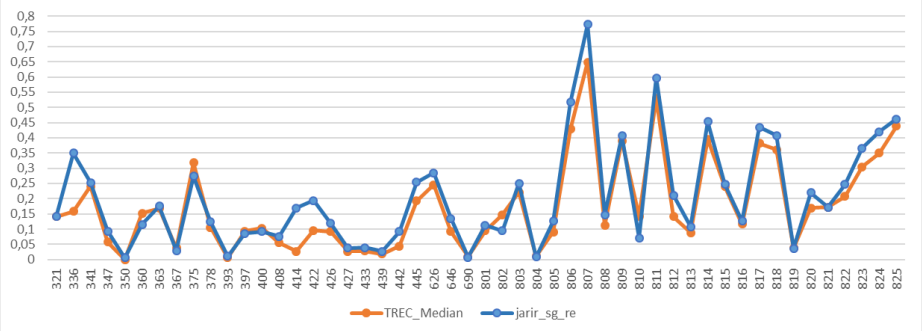

Fig. 2. Performance of *jarir_sg_re* run measured by MAP compared with *TREC_Median*.

We can observe that most of our results are above the median scores. More precisely, 40 out of 50 topics have a result above the median including 2 topics having the best score. Besides, we also achieved the best P@10 value on 3 topics and the best NDCG value on 2 topics. It is important to note that the *TREC_Median* scores were computed from runs that used existing relevance judgments. As stated before, our runs were produced without using any prior judgment. Furthermore, it is worth mentioning that 27

out of 50 topics achieved a score above the median over all TREC submitted runs, i.e., over the four categories of runs.

Among our runs, *jarir_sg_re* achieved the best retrieval effectiveness according to all measures. As described in Section 2, *jarir_sg_re* used Word2Vec Skip-Gram model for training, expansion terms that are similar to the entire query and query reweighting. Figure 3 shows a performance comparison between all submitted runs in terms of MAP, NDCG and P@10.

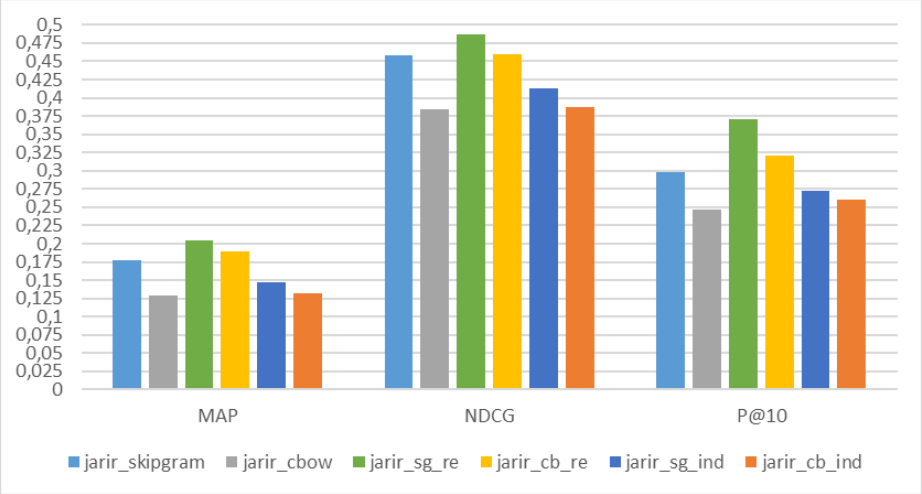

Fig. 3. Comparison of performance by MAP, NDCG and P@10 for all submitted runs.

As it can be observed, Word2Vec Skip-Gram model outperformed the CBOW model in all scenarios. Interestingly enough, using query reweighting yielded the best results and brought a significant improvement compared to other approaches. That is, retrieval effectiveness decreases when all terms of the expanded query have been assigned the same weight. Furthermore, using similarity to the whole query to get expansion terms yielded better results than using similarity to individual query terms, which is consistent with our expectation. However, what we did not expect was that the run *jarir_cbow* was ranked last after the run *jarir_cbow_ind*, which means that in the case of CBOW model, selecting expansion terms that are similar to individual query terms gave slightly better results.

4 Conclusion

In this paper, we proposed a Query Expansion approach based on Natural Language Pre-processing techniques and Word2Vec embedding models. Overall results are consistent with our expectations. The obtained scores show that for our best run, 80% of our topics are above the TREC median scores. In our experiments, we trained Word2Vec models over the entire TREC Washington Post Corpus. As a future work,

we plan to investigate local training of Word2Vec models in a query-specific manner as previous studies demonstrated promising results [2].

References

1. Carpineto, C., Romano, G.: A Survey of Automatic Query Expansion in Information Retrieval. ACM Computing Surveys (CSUR), 44(1): 1–50 (2012).
2. Diaz, F., Mitra, B., Craswell, N.: Query Expansion with Locally-Trained Word Embeddings. arXiv:1605.07891 ,2016.
3. Manning, C. D., Raghavan, P., Schütze, H.: Introduction to Information Retrieval. Cambridge University Press, UK, 2008.
4. Mikolov, T., Chen, K., Corrado, G., & Dean, J.: Efficient Estimation of Word Representations in Vector Space. arXiv:1301.3781, 2013.
5. Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., Dean, J.: Distributed Representations of Words and Phrases and their Compositionality. In: Advances in Neural Information Processing Systems, Proceedings of the 26th International Conference on Neural Information Processing Systems, Vol. 2, pp. 3111-3119, Lake Tahoe, Nevada, December 05 - 10, 2013.
6. Ounis, I., Amati, G., Plachouras, V., He, B., Macdonald, C., Johnson, D.: Terrier Information Retrieval Platform. In: Advances in Information Retrieval, Proceedings of the European Conference on Information Retrieval, pp. 517–519, March 26-29, 2005, Grenoble, France.
7. Porter, M. F.: An algorithm for suffix stripping. Program journal, 14(3): 130–137 (1980).
8. Robertson, S. E., Walker, S.: Some Simple Effective Approximations to the 2-Poisson Model for Probabilistic Weighted Retrieval. In: Proceedings of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 232–241. Springer-Verlag, Dublin, Ireland, July 03 - 06, 1994.