A Comparison of Deep Learning Based Query Expansion with Pseudo-Relevance Feedback and Mutual Information

Mohannad Almasri, Catherine Berrut, Jean-Pierre Chevallet

A Comparison of Deep Learning Based Query Expansion with Pseudo-Relevance Feedback and Mutual Information

Mohannad ALMasri, Catherine Berrut, and Jean-Pierre Chevallet

Université de Grenoble Alpes
{mohannad.almasri,catherine.berrut,jean-pierre.chevallet}@imag.fr
LIG laboratory, MRIM group, Grenoble, France

Abstract. Automatic query expansion techniques are widely applied for improving text retrieval performance, using a variety of approaches that exploit several data sources for finding expansion terms. Selecting expansion terms is challenging and requires a framework capable of extracting term relationships. Recently, several Natural Language Processing methods, based on Deep Learning, are proposed for learning high quality vector representations of terms from large amounts of unstructured text data with billions of words. These high quality vector representations capture a large number of term relationships. In this paper, we experimentally compare several expansion methods with expansion using these term vector representations. We use language models for information retrieval to evaluate expansion methods. The experiments are conducted on four CLEF collections show a statistically significant improvement over the language models and other expansion models.

1 Introduction

User queries are usually too short to describe the information need accurately. Important terms can be missing inside the query, leading to a poor coverage of the relevant documents. To solve this problem, automatic query expansion techniques, using a variety of approaches exist, leveraging on several data sources and employ different methods for finding expansion terms [2]. Selecting such expansion terms is challenging and requires a framework capable of adding interesting terms to the query. Different approaches have been proposed for selecting expansion terms.

Pseudo-relevance feedback (PRF) assumes that the top-ranked documents returned for the initial query are relevant, and uses a sub set of the terms extracted from those documents for expansion. PRF has been proven to be effective in improving retrieval performance [4].

Corpus-specific approaches analyze the content of the whole document collection. Corpus-specific approaches generate correlation between pairs of terms by co-occurrence [6], mutual information [3], etc. Mutual information (MI) is a good measure to assess how much two terms are related [3]. Mutual information

analyzes the entire collection in order to extract the association between terms. For each query term, every term that has a high mutual information score with it is used to expand the user query.

Many approaches exploit knowledge bases or thesauruses for query expansion, among them: WordNet [12], UMLS Meta thesaurus [13], Wikipedia [11], etc. The nature of these resources varies: linguistic like WordNet, domain specific like UMLS in the medical domain, or knowledge about named entities like Wikipedia.

Other approaches like semantic vectors and neural probabilistic language models, propose a rich representation for terms in order to capture the similarity between them. In these approaches, a term is represented by a mathematical object in a high dimensional semantic space which is equipped with a metric. The metric can naturally encode similarities between the corresponding terms. A typical instantiation of these approaches is to represent each term by a vector and use a cosine or distance between term vectors in order to measure term similarity [7][10][1].

Recently, several efficient Natural Language Processing methods, based on Deep Learning, are proposed to learn high quality vector representations of terms from large amounts of unstructured text data with billions of words [5]. This high quality vector representations capture a large number of term relationships. In this paper, we propose to investigate these term vector representations in query expansion in order to experimentally compare these approaches with two other expansion approaches: pseudo-relevance feedback and mutual information.

Our experiments are conducted on four CLEF medical collections. We use a language modeling framework to evaluate expanded queries. The experimental results show that the retrieval effectiveness can be improved significantly over the ordinary language models and pseudo-relevance feedback.

This paper is organized as follows. In Section 2, we present the query expansion method we use. Our experimental set-up and results are presented in section 3. Finally, section 4 concludes the paper.

2 Query Expansion Method

We propose to investigate term vector representations in query expansion. In this section, we first present the source of these term vectors. Then, we describe how we use these term vectors for query expansion.

2.1 Expansion Terms

In this step, learning takes place from a large amount of unstructured text data, term vector representations are learned using Deep Learning. The resulting vectors carry relationships between terms, such as a city and the country it belongs to, e.g. France is to Paris as Germany is to Berlin [5]. Therefore, each term t is represented by a vector of a predefined dimension v_t[1]. In the rest of paper, we call this vector *Deep Learning Vector*. The similarity between two terms t_1 and

[1] A real-valued vector of a predefined dimension, 600 dimensions for exemple.

t_2 is measured by the normalized cosine between their two vectors: v_{t_1} and v_{t_2}.

$$SIM(t_1, t_2) = \widetilde{cos}(v_{t_1}, v_{t_2}) \tag{1}$$

where $\widetilde{cos}(v_{t_1}, v_{t_2}) \in [0, 1]$ is the normalized cosine between the two term vectors v_{t_1} and v_{t_2}. Based on this normalized cosine similarity between terms, we now define the function that returns the k-most similar terms to a term t, $top_k(t)$:

$$top_k : V \rightarrow 2^V \tag{2}$$

where V is the set of all terms t.

2.2 Building Expanded Query

Let q be a user query represented by a bag of terms, $q = [t_1, t_2, ..., t_{|q|}]$. Each term in the query has a frequency $\#(t, q)$. In order to expand a query q, we follow these steps:

- For each $t \in q$, collect the k-most similar terms to t using the function $top_k(t)$, eq.2. The expanded query q' is defined as follows: $q' = q \bigcup_{t \in q} top_k(t)$.
- The frequency of each $t \in q$ still the same in the expanded query q':

$$\#(t, q') = \#(t, q) \tag{3}$$

- The frequency of each expansion term $t' \in top_k(t)$ in the expanded query q' is given as follows:

$$\#(t', q') = \alpha \times \#(t, q') \times \widetilde{cos}(v_t, v_{t'}) \tag{4}$$

  Where $\alpha \in [0, 1]$ is a tuning parameter that determines the importance of expansion terms.
  In the rest of paper, the expansion method based on deep learning vectors is denoted by VEXP.

3 Experiments

The first goal of our experiments is to analyze the effect of the number of expansion terms k on the retrieval performance using deep learning vectors. The second goal is to compare between the proposed expansion based on deep learning vectors (VEXP) with two existing expansion approaches: pseudo-relevance feedback (PRF) [4], and mutual information (MI) [3], which both have been proven to be effective in improving retrieval performance. In order to achieve the comparison between VEXP, PRF, and MI, we use a language model with no expansion as a baseline (NEXP).

Documents are retrieved using Indri search engine [9], and two smoothing methods of language models: Jelinek-Mercer and Dirichlet.

The optimization of the free parameter α (eq.4) for controlling expansion terms importance is done using 4-fold cross-validation with Mean Average Precision (MAP) as the target metric. We vary α values between $[0.1, 1]$ with 0.1 as an interval. The best values of the tuning parameter α that indicate the importance of expansion terms are between $[0.2, 0.4]$.

In our experiments, the statistical significance is determined using Fisher's randomization test with $p < 0.05$ [8].

3.1 Evaluation Data

Four medical corpora from CLEF[2] are used.

- Image2010, Image2011, Image2012: contain short documents and queries.
- Case2011: contains long documents and queries.

Table 1 shows some statistics about them, *avdl* and *avql* are average length of documents and queries, respectively. These medical collections provide a huge amount of medical text that we need in the training phase, i.e. hundreds of millions of words for extracting high quality deep learning vectors.

Table 1. Training and testing collections.

Corpus	**#d**	**#q**	**avdl**	**avql**
Image2009	74901	25	62.16	3.36
Image2010	77495	16	62.12	3.81
Image2011	230088	30	44.83	4.0
Image2012	306530	22	47.16	3.55
Case2011	55634	10	2594.5	19.7
Case2012	74654	26	2570.72	24.35

3.2 Learning Data and Tools

We use word2vec[3] to generate deep learning vectors. The word2vec tool takes a text corpus as input and produces the term vectors as output. It first constructs a vocabulary from the training text data and then learns the vector representation of terms. We build our training corpus using three different CLEF medical collection: Image2009, Case2011, Case2012. Our training corpus consists of about 400 millions words. The vocabulary size for this training corpus is about 350 thousands different terms. We have used the recommended setting for this training tool like the term vector dimension and the learning context window size.

3.3 Number of Expansion Terms Analysis

We first analyze the effect of number of expansion terms k on the retrieval performance of VEXP. Each query term is expanded by $k \in \{1, 2, 3, ..., 10\}$ terms. Stop words are not considered in the expansion. The optimal k value for the number of expansion terms vary depending on the test collections. All tested k values are given in Table 2. The best performance is presented in bold.

Similarly, we analyzed the best number of expansion terms for the two other approaches: PRF and MI:

- For PRF, we have tested several configurations for $k \in \{5, 10, ..., 50\}$ and the number of feedback documents $\#fbdocs \in \{5, 10, , ..., 50\}$.
- For MI, we have also tested several configurations for $k \in \{1, 2, ..., 25\}$.

Table 3 gives the best configurations for VEXP, PRF, and MI.

[2] www.clef-initiative.eu

[3] An efficient implementation of the continuous bag-of-words and skip-gram architectures for computing vector representations of terms [5].

Table 2. VEXP performance using MAP on test collections. k is the number of expansion terms for each query term.

	Jelinek-Mercer				Dirichlet			
k	Image2010	Image2011	Image2012	Case2011	Image2010	Image2011	Image2012	Case2011
1	0.3286	0.2258	0.1997	0.1373	0.3397	0.2173	0.1947	0.1288
2	0.3298	0.2325	0.1988	0.1431	0.3361	0.2204	0.1890	0.1345
3	0.3395	0.2330	0.1996	0.1440	0.3411	0.2192	0.1902	0.1366
4	0.3399	0.2338	0.2002	0.1413	0.3561	0.2175	0.1909	0.1384
5	0.3323	0.2340	0.1909	**0.1634**	0.3519	0.2187	0.1787	0.1410
6	**0.3402**	0.2324	0.1909	0.1432	**0.3603**	0.2163	0.1798	**0.1451**
7	0.3397	0.2333	0.1881	0.1446	0.3599	0.2184	0.1778	0.1431
8	0.3397	**0.2353**	0.1895	0.1414	0.3584	0.2200	0.1813	0.1416
9	0.3365	0.2230	0.2004	0.1387	0.3544	**0.2221**	0.1953	0.1379
10	0.3362	0.2233	**0.2036**	0.1343	0.3510	0.2215	**0.1990**	0.1357

Table 3. Best configurations for VEXP, PRF, and MI.

		Jelinek-Mercer				Dirichlet			
		Image2010	Image2011	Image2012	Case2011	Image2010	Image2011	Image2012	Case2011
PRF	k	15	10	20	10	15	10	10	10
	#fbdocs	10	10	20	10	10	10	10	10
MI	k	10	8	6	10	10	7	6	10
VEXP	k	6	4	10	4	5	9	10	5

3.4 Performance Comparison

In this section, we compare between three expansion methods: VEXP, PRF, and MI, using a language model with no expansion as a baseline (NEXP). We use two tests for statistical significance: † indicates a statistical significant improvement over NEXP, and $*$ indicates a statistical significant improvement over PRF. Results are given in Table 4. We first observe that VEXP is always statistically better than NEXP for the four test collection, which is not the case for PRF and MI. VEXP shows a statistically significant improvement over PRF in five cases.

Table 4. Performance comparison using MAP on test collections. † indicates statistically significant improvement over NEXP. * indicates statistically significant improvement over PRF, $p < 0.05$.

	Jelinek-Mercer				Dirichlet			
	Image2010	Image2011	Image2012	Case2011	Image2010	Image2011	Image2012	Case2011
NEXP	0.3016	0.2113	0.1862	0.1128	0.3171	0.2033	0.1681	0.1134
PRF	0.3090	0.2136	0.1920	0.1256	0.3219	0.2126	0.1766	0.1267
MI	0.3239	0.2116	0.1974	0.1360	0.3338	0.2110	0.1775	0.1327
VEXP	0.3402†*****	0.2340†	0.2036†	0.1634†*****	0.3603†*****	0.2221†	0.1990†*****	0.1451†*****

Deep learning vectors are a promising source for query expansion because they are learned from hundreds of millions of words, in contrast to pseud relevance feedback which is obtained from top retrieved document and mutual information which is calculated on the collection itself. Deep learning vectors are not only useful for collections that were used in the training phase, but also

for other collections which contain similar documents. In our case, training and testing collections dealing with medical cases.

There are two architectures of neural networks for obtaining deep learning vectors: skip-gram and bag-of-words [5]. We only present the results obtained using the skip-gram architecture in our experiments. We have also evaluated the bag-of-words architecture, but there was no big difference in retrieval performance between the two architectures.

4 Conclusions

We explored the use of the relationships extracted from deep learning vectors for query expansion. We showed that deep learning vectors are a promising source for query expansion by comparing it with two effective methods for query expansion: pseudo-relevance feedback and mutual information. Our experiments on four CLEF collections showed that using this expansion source gives a statistically significant improvement over baseline language models with no expansion and pseudo-relevance feedback. In addition, it is better than the expansion method using mutual information.

References

1. Yoshua Bengio, Holger Schwenk, Jean-Sbastien Sencal, Frderic Morin, and Jean-Luc Gauvain. Neural probabilistic language models. volume 194 of *Studies in Fuzziness and Soft Computing*, pages 137–186. Springer Berlin Heidelberg, 2006.
2. Claudio Carpineto and Giovanni Romano. A survey of automatic query expansion in information retrieval. *ACM Comput. Surv.*, 44(1):1:1–1:50, January 2012.
3. Jiani Hu, Weihong Deng, and Jun Guo. Improving retrieval performance by global analysis. In *ICPR 2006.*, pages 703–706, 2006.
4. Victor Lavrenko and W. Bruce Croft. Relevance based language models. SIGIR '01, pages 120–127, New York, NY, USA, 2001. ACM.
5. T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. *CoRR*, 2013.
6. Helen J. Peat and Peter Willett. The limitations of term co-occurrence data for query expansion in document retrieval systems. *J. Am. Soc. Inf. Sci*, 1991.
7. Midori Serizawa and Ichiro Kobayashi. A study on query expansion based on topic distributions of retrieved documents. volume 7817 of *LNCS*, pages 369–379. 2013.
8. Mark D. Smucker, James Allan, and Ben Carterette. A comparison of statistical significance tests for information retrieval evaluation. CIKM '07. ACM, 2007.
9. T. Strohman, D. Metzler, H. Turtle, and W. B. Croft. Indri: A language model-based search engine for complex queries. 2004.
10. D. Widdows and T. Cohen. The semantic vectors package: New algorithms and public tools for distributional semantics. In *ICSC*, pages 9–15, 2010.
11. Yang Xu, Gareth J.F. Jones, and Bin Wang. Query dependent pseudo-relevance feedback based on wikipedia. SIGIR '09, pages 59–66, Boston, MA, USA, 2009.
12. Jiuling Zhang, Beixing Deng, and Xing Li. Concept based query expansion using wordnet. AST '09, pages 52–55. IEEE Computer Society, 2009.
13. W. Zhu, Xuheng Xu, Xiaohua Hu, I.-Y. Song, and R.B. Allen. Using umls-based re-weighting terms as a query expansion strategy. pages 217–222, May 2006.