Focal Elements of Neural Information Retrieval Models. An Outlook through a Reproducibility Study

Stefano Marchesin, Alberto Purpura, Gianmaria Silvello

Department of Information Engineering, University of Padua, Italy.

Abstract

This paper analyzes two state-of-the-art *Neural Information Retrieval (NeuIR)* models: the *Deep Relevance Matching Model (DRMM)* and the *Neural Vector Space Model (NVSM)*. Our contributions include: (i) a reproducibility study of two state-of-the-art supervised and unsupervised NeuIR models, where we present the issues we encountered during their reproducibility; (ii) a performance comparison with other lexical, semantic and state-of-the-art models, showing that traditional lexical models are still highly competitive with DRMM and NVSM; (iii) an application of DRMM and NVSM on collections from heterogeneous search domains and in different languages, which helped us to analyze the cases where DRMM and NVSM can be recommended; (iv) an evaluation of the impact of varying word embedding models on DRMM, showing how relevance-based representations generally outperform semantic-based ones; (v) a topic-by-topic evaluation of the selected NeuIR approaches, comparing their performance to the well-known BM25 lexical model, where we perform an in-depth analysis of the different cases where DRMM and NVSM outperform the BM25 model or fail to do so. We run an extensive experimental evaluation to check if the improvements of NeuIR models, if any, over the selected baselines are statistically significant.

Keywords: neural information retrieval, reproducibility, neural vectors, deep learning

1. Introduction

Recently, *Neural Information Retrieval (NeuIR)* has attracted a great deal of attention from the research community. A dedicated workshop series was held at the *ACM SIGIR Conference on Research & Development in Information Retrieval (SIGIR)* [17, 19], an in depth monograph [56] and a special issue in the *Information Retrieval Journal* (IRJ) [18] have been published in 2018. Moreover, at SIGIR, papers employing deep learning are increasing at a fast pace – i.e., from one article published in 2015 to eleven articles published in 2017 [2].

Nevertheless, the burst of enthusiasm contrasts the concerns about the actual efficiency and effectiveness of NeuIR methods [50]. To this end, Wei et al. in [82] critically examined the advances in NeuIR regarding the reproducibility of the systems, the data on which they are tested and the improvements over robust and well-tuned baselines.

The issue of reproducibility of search algorithms regards the IR field as a whole, not the only NeuIR. Reproducibility is now a central research topic for the IR community with dedicated workshops [6, 15, 25] which raised awareness and proposed a reproducibility model – i.e., PRIMAD [32]; with a specific track at the *European Conference on Information Retrieval (ECIR)* since 2015; and, with dedicated journal special issues [26, 27]. Reproducibility efforts focused on several core topics in IR ranging from reproducing baselines [51, 84] and core system components [69] to evaluation [30, 43] and advanced applications [40]. Some works have focused on reproducing neural architectures for question answering [23], reproducing and

Email addresses: stefano.marchesin@unipd.it (Stefano Marchesin), purpuraa@dei.unipd.it (Alberto Purpura), silvello@dei.unipd.it (Gianmaria Silvello)

generalizing the linear transformation of word embeddings [86], or replicating neural search models on *ad-hoc* test collections [28]. However, there has been no specific effort towards the reproducibility and generalization of NeuIR systems.

A NeuIR system is an ecosystem of components. Its reproducibility is quite challenging, even when the source code is available. Such systems often include text processing methods, lexical ranking models, word embeddings, optimizers, query expansion methods, and other traditional *Information Retrieval (IR)* and *Natural Language Processing (NLP)* components. These are used to feed a shallow or deep neural network. Every single component has a sizable impact on the performances of the system, but its interactions have not been accurately investigated. For instance, how documents are pre-processed has implications on the creation of term embeddings. In turn, they affect the optimizer and parameter selection. Even though this domino effect holds true for almost all advanced IR systems, it is particularly accentuated in NeuIR systems – where it is hard (or even impossible) to understand why we get a specific output and thus how to detect the component which may be not working correctly. To reproduce the results achieved by NeuIR models, each system component needs to be finely tuned. Describing a neural network architecture in detail or providing the source code is usually not sufficient to reproduce the system successfully. When generalizing the application of NeuIR models on collections from different domains or in different languages, the problem is even more significant, as it needs to adapt and optimize many components on a different setting: i.e. from English news to German news, Web pages or medical documents.

Hence, with the increasing popularity of NeuIR models, the analysis of these approaches, especially through reproducibility studies, becomes crucial for in-depth understanding. In fact, the more we understand single system components and their interactions, the more we can generalize the approach and successfully transfer its performances to different domains.

However, the complexity of NeuIR models is not the only obstacle for their reproducibility. In many cases, NeuIR approaches work and are tested only on big sets of interaction proprietary data [9, 57, 88] which may not be easily accessible or available at all. On the other hand, to reproduce an experiment we require the original dataset or a reasonable approximation of it. Luckily, there are also several NeuIR approaches working on shared TREC collections [11, 35, 74, 90], especially in an *ad-hoc* retrieval setting.

In this paper, we reproduce and thoroughly evaluate two NeuIR systems: *Deep Relevance Matching Model (DRMM)* and *Neural Vector Space Model (NVSM)*. DRMM [35] was presented at the *25th ACM International Conference on Information and Knowledge Management* (CIKM 2016). DRMM achieved competitive results on re-ranking tasks in *ad-hoc* retrieval and it is still one of the reference NeuIR approaches. NVSM was published in the *ACM Transactions of Information Systems* in 2018 [74] and evaluated on shared TREC test collections, yielding competitive results in *ad-hoc* retrieval. NVSM is one of the very few completely unsupervised existing NeuIR models, therefore it has excellent potential for generalization since it does not need any interaction data nor labeled data, which is a scarce resource in a typical IR experimental setting.

We replicate the experimental results in the paper presenting DRMM, leveraging on the source code shared by the authors. Whereas, we re-implement NVSM from scratch in Python, relying on widely-used and consolidated libraries like TensorFlow. [1] This choice enables a straightforward comparison of NVSM with many other NeuIR models available in public repositories. [2] We reproduce the results of the original paper on the test collections used by the authors not only for NVSM, but also for the main baselines considered. Our aim is to check if we could reproduce the results achieved both with NVSM and the proposed baselines.

We consider four different perspectives in the analysis of DRMM and NVSM. First of all, we perform an in-depth evaluation of DRMM and NVSM and compare them with the most widely adopted lexical IR models such as TF-IDF, BM25, Query Likelihood Model (QLM), Divergence from Randomness (DFR), and other basic semantic models based on Word2Vec. The goal is to evaluate the potential of NeuIR approaches compared to a few widely-used and not necessarily heavily tuned IR models. Understanding NeuIR strengths and weaknesses can enhance their integration into full-stack IR systems, which employ a variety of pre- and post-retrieval components such as query expansion and relevance feedback.

[1] `https://www.tensorflow.org/`.

[2] `https://github.com/NTMC-Community/MatchZoo`.

Secondly, we test DRMM and NVSM on new search domains for which they were not initially designed: (i) the multilingual domain, where we consider Italian, German and Farsi [1, 22] news document collections from *Conference and Labs of the Evaluation Forum (CLEF)*; (ii) the medical domain, where we consider the OHSUMED collection [42] composed of references/documents from MEDLINE (the online life sciences/biomedicine information database); (iii) the Web domain, where we consider a small Web collection, namely the *Text REtrieval Conference (TREC)* WT2g [41]. Since NVSM does not scale to large document collections due to memory and time constraints, we chose collections of the same order of magnitude as those adopted in the original paper.

Thirdly, we perform an analysis of the impact of different word vector representations (i.e., word embedding models) on DRMM – considering other embedding models such as FastText [8], Word2Vec [55] and the word embeddings computed by NVSM [74]. Regarding NVSM, we cannot perform the same analysis as the joint learning of word and document embeddings is an inherent part of the model.

Finally, we perform a topic-by-topic analysis and comparison of selected NeuIR systems to the well-known BM25 lexical retrieval model. We highlight the performance differences among systems, describing the topics where NeuIR approaches are performing better than lexical models, and vice versa. The aim is to understand whether the two approaches are orthogonal, if they share common features, when it is more convenient to use a lexical model or a neural model.

Contributions. The contributions of this paper include: (i) a reproducibility study of DRMM and NVSM on the original experimental collections; (ii) a comparison of DRMM and NVSM to widely-used lexical and semantic IR models; (iii) an application of specific NeuIR models in heterogeneous search domains; (iv) an evaluation of the impact of different word embedding models on DRMM; and, (v) a topic-by-topic evaluation of selected NeuIR approaches, based on a comparison of NVSM, DRMM and the BM25 retrieval model. The source code and the settings we used for this study are available at the following URL: `https://github.com/giansilv/NeuralIR`.

Outline. The rest of the paper is organized as follows: in Section 2, we describe the experimental setup employed in our tests; in Sections 3 and 4, we describe the DRMM and NVSM models respectively, along with the results of our reproducibility experiments; in Section 5, we report a comparison of the selected NeuIR systems with traditional lexical approaches and other state-of-the-art models; in Section 6, the robustness of DRMM and NVSM is evaluated over a set of collections in different languages and from distinct domains; Section 7 assesses the impact of different word embedding models on DRMM; in Section 8, we perform an in-depth topic-by-topic analysis of the characteristics of the selected NeuIR models; and finally, in Section 10, conclusions are drawn with a discussion on the lessons learnt.

2. Experimental setup

To evaluate DRMM and NVSM, we use eleven experimental collections in different languages from various domains. The collections characteristics are indicated in Tables 1 and 2. For the retrieval experiments, we consider three combinations of the topic fields: the *title* (T) field only, the *description* (D) field only, or both the *title* and *description* (TD) fields. We compare the selected NeuIR models to other well-known semantic and lexical approaches for retrieval. The semantic models include Word2Vec and *Latent Dirichlet Allocation (LDA)*, while the lexical ones refer to QLM, BM25, TF-IDF, and the *Poisson estimation for randomness using Laplace succession for normalisation (PL2)* model [5] (hereafter DFR). In particular, for Word2Vec we consider the approaches originally proposed in [79], which define a document representation as the weighted sum of its word embeddings. We consider the unweighted sum, referred to as Word2Vec (add), and the sum weighted by the term's self-information, [3] referred to as Word2Vec (si). Whereas, for QLM, we consider the approaches with Jelinek-Mercer smoothing, referred to as QLM (jm), and with Dirichlet smoothing, referred to as QLM (dir) [89].

[3] Self-information is a term specificity measure similar to IDF [16].

Reproducibility experiments. In the reproducibility experiments of DRMM, we employ the Robust04 collection [38], which is one of the two collections where the model was evaluated in the reference paper [35].

In the reproducibility experiments of NVSM, we use the same set of 6 newswire article collections from TREC – see Table 1 – adopted in the reference paper [74]. Four of the collections considered herein are subsets of the TIPSTER corpus [38]: *Associated Press 88-89* (AP 88-89), *Financial Times* (FT), *LA Times* (LA) and *Wall Street Journal* (WSJ) [39]. The remaining two collections are the Robust04 collection – based on TIPSTER Disk 4&5 minus CR – and the *New York Times* (NY) collection – which consists of articles written and published by the *New York Times* between 1987 and 2007.

	AP88-89	FT	LA	NY	Robust04	WSJ
Vocabulary	247,725	437,511	197,024	1,062,137	760,467	184,717
Document Count	164,597	210,158	131,896	1,855,658	528,155	173,252
Query Count	149	144	143	50	249	150

Table 1: Statistics of the AP88-89, FT, LA, WSJ, Robust04 and NY collections.

Comparison with other lexical and semantic models. In this set of experiments, we compare DRMM and NVSM to other lexical (i.e. BM25, TF-IDF, DFR, QLM) and semantic models (i.e. Word2Vec (si)) on the Robust04 and NY collections. The aim is to compare the performance of NeuIR models considered to lexical and semantic matching techniques and to state-of-the-art retrieval approaches. The performances of other competitive approaches as the BM25+RM3 model in Anserini [84] and the one presented in [85] using BERT [21] to perform retrieval, are also considered for reference. We also report the performances of the best TREC systems when available.

Collection-based evaluation. In this set of experiments, we evaluate the impact of the domain and/or language of different collections on the performance of selected NeuIR models. The DRMM and NVSM models are evaluated on five collections from different domains and languages:

- WT2g: is a collection of documents crawled from the Web, used in TREC 1999 Web Track [41];
- OHSUMED: consists of a set of 348,566 references/documents from MEDLINE, an on-line life sciences/biomedicine information database, consisting of titles and/or abstracts from many published medical journals. [4] OHSUMED contains 106 topics, divided into 63 official topics and 43 pre-test topics that were rejected from official TREC-9 runs for a variety of reasons, often because they had too few relevance judgments. Topic fields are: title (patient description) and description (information need) [42].
- CLEF: document sets in different languages form the CLEF corpora but with common features: the same genre and period [1, 22]. The Italian and German corpora are composed of newspaper articles from 1994 to 1995 and the Persian corpus (i.e., Farsi) from 1996 to 2002. The German and Italian news agency dispatches are all gathered from the Swiss news agency and comprise of the same corpus translated in different languages.

The detailed characteristics of the collections are reported in Table 2. For reference, we also report – whenever possible – the performance of the best TREC and CLEF systems on the experimental collections relative to the evaluation measures we consider.

Embedding-based evaluation. In this set of experiments, we evaluate DRMM on the Robust04, NY, WT2g and OSHUMED collections using word embeddings generated with different techniques. Indeed, different word representations can have a sizeable impact on the performance of neural models. This is also shown in [52], where the authors evaluate BERT [21] and ELMo [64] representations for retrieval. We consider four different types of word embeddings:

[4] `https://www.nlm.nih.gov/bsd/medline.html`.

	WT2g	OHSUMED	CLEF-DE	CLEF-FA	CLEF-IT
Vocabulary	1,049,056	265,923	739,053	399,185	232,335
Document Count	247,491	348,566	223,132	166,774	157,558
Query Count	50	97	95	100	90

Table 2: Statistics of the WT2G, OHSUMED, CLEF-DE, CLEF-FA and CLEF-IT collections.

W2V: Word2Vec embeddings trained on each collection used to perform retrieval with Gensim [67], following the instruction in the DRMM reference paper [35]. To train this model, we consider a context window size of 10, an embeddings size of 300, 10 negative samples, a subsampling of frequent words of 10^{-4}, and we discard all terms appearing less than 10 times. We empirically select 10 as the best number of training epochs for the model after evaluating the system performance with a model trained for 5, 15 and 20 epochs. We also decided to apply punctuation and stopwords removal (using the INQUERY [13] stoplist), stemming (Krovetz [47]) and to remove all terms containing one or more digits or shorter than three characters as a pre-processing step on the document corpus before training the model;

FastText: FastText [8] embeddings trained with Gensim on each collection used to perform retrieval. FastText is an extension of Word2Vec to compute word representations based on character n-grams. In this case, we consider an embeddings size of 300, a context window of size 10, a subsampling of frequent words of 10^{-4}, 10 negative samples, and we discard words appearing in the collection less than 10 times. We train this model for 10 epochs on each of the selected experimental collections;

NVSM WE: the set of word embeddings learned and used by the NVSM model. The NVSM retrieval model works by learning both word and document representations from scratch in an unsupervised fashion and then using them to perform retrieval. Details on the training and retrieval processes are described in Section 4. Since NVSM embeddings are learned – like Word2Vec embeddings – based on the cosine similarity metric, we can use them in the DRMM model without any modifications. As done for the other embedding models, we train a new model on each of the considered collections;

W2V Google: Word2Vec embeddings trained by Google on part of Google News dataset (about 100 billion words). The model contains 300-dimensional vectors for 3 million words and phrases. [5]

Topic-based evaluation. In this set of experiments, we conduct a topic-by-topic analysis on the performance of the selected NeuIR models and the BM25 model. The aim is to perform an in-depth comparison between lexical and semantic retrieval models. We consider Robust04, NY, WT2g, OHSUMED and CLEF collections. Each plot in Section 8 represents the *Average Precision at Rank 1000 (AP)* of BM25 (x-axis) and NVSM or DRMM (y-axis).

We also employ the *Kernel Density Estimation (KDE)* [81] to estimate the *Probability Density Function (PDF)* of the AP of BM25, NVSM and DRMM models for all the topics. Then, we compute the *Kullback-Leibler Divergence (KLD)* [48] between these PDFs to get an estimate of the difference in AP distribution values for each model.

3. DRMM: Reproducibility of a supervised neural model

Model description. The Deep Relevance Matching Model (DRMM) [35] is a supervised system for *ad-hoc* retrieval which implements the following strategies for relevance matching:

- it considers exact matching signals between query and document terms;
- it enables a different importance weight of query terms;

[5] The model is available at: `https://code.google.com/archive/p/word2vec/`.

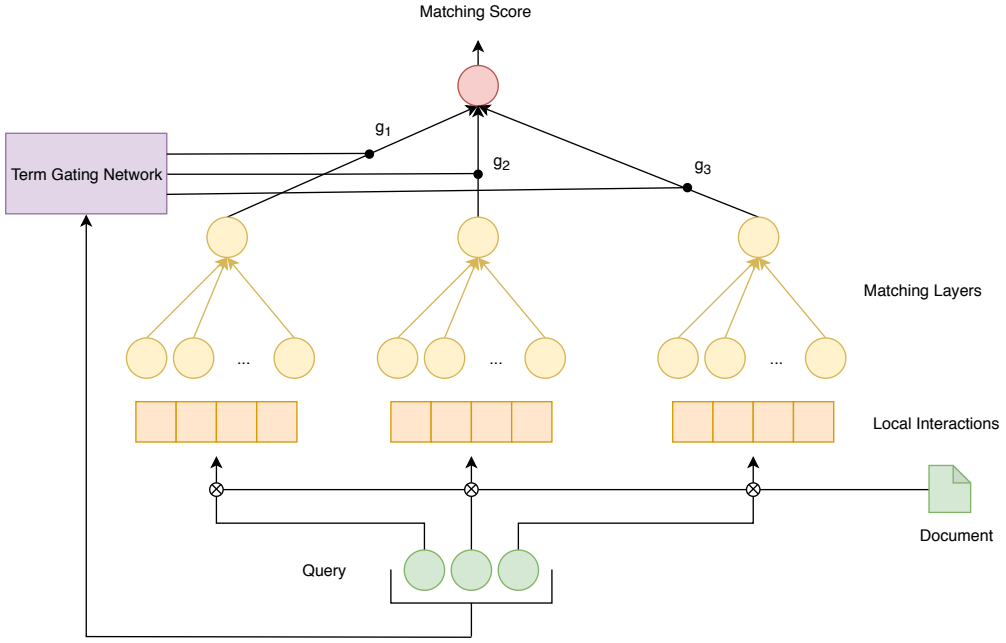


Figure 1: Architecture of Deep Relevance Matching Model (DRMM).

- it complies with different matching requirements, i.e., *verbosity* and *scope* hypotheses. The *verbosity* hypothesis assumes that a long document is like a short document, covering a similar scope with more words. Vice versa, the *scope* hypothesis assumes that, in longer documents, only portions of the content may be relevant, the whole document is therefore not needed to be relevant for a query.

Based on word embeddings, DRMM considers local interactions between each pair of terms in a query and a document. For each query term, it maps the variable-length local interactions into a fixed-length matching histogram. Then, these matching histograms are fed into a feed-forward neural network that outputs a matching score for each pair of query/document terms. Finally, a weighted sum of these scores for each term pair is computed in order to produce a global matching score for each query. The scheme of this architecture is depicted in Figure 1.

The local interactions between each query and document term are obtained computing the cosine similarity between each term vector. Scores are then aggregated in matching histograms which discretize the interval $[-1, 1]$ into a set of k bins. For instance, if we take 0.5 as the bin size, each bin will contain the cosine similarity scores respectively in the intervals: $[-1, -0.5)$, $[-0.5, 0.5)$, $[0.5, 1)$. The interval $[1, 1]$, considers the exact match scores in a separate bin. The authors propose three different ways to map the values in the matching histograms:

- Count-based Histogram (CH): considers the count of local interactions in each bin as the histogram value;
- Normalized Histogram (NH): normalizes the count value in each bin by the total count;
- LogCount-based Histogram (LCH): applies logarithm over the count value in each bin.

In our experiments and in the reference paper, the number of bins considered is 30, and we only evaluate the results of the best performing matching-histograms configuration which is LCH.

The feed-forward network, in its best-performing configuration, has two layers, the first one of 5 nodes and the second one of a single node, both using *tanh* as the activation function and a bias term.

The matching scores returned by the network are then weighted with coefficients computed by a term gating network. The term gating network optimizes the function reported below in order to estimate the best values of the coefficients g_i for the aggregation of the matching scores at the query level:

$$g_i = \frac{exp(w_g x_i^{(q)})}{\sum_{j=1}^{M} exp(w_g x_j^{(q)})},\ i = 1, ..., M, \tag{1}$$

where w_g is the weight vector of the term network, and $x_i^{(q)},\ i = 1, ..., M$ denotes the i-th query term. The authors developed different types of weighting functions which require different input values:

- Term Vector (TV): in this case, $x_i^{(q)}$ denotes the i-th query term vector, and w_g is a weight vector of the same size of the term vectors;
- Inverse Document Frequency (IDF): in this case, $x_i^{(q)}$ denotes the inverse document frequency of the i-th query term, and w_g is a coefficient with a single parameter.

From the experiments in [35], the best performing approach to compute these weights is the second one. For this reason, we employ the IDF weighting scheme in our tests.

Finally, the training of the system is performed minimizing the following hinge loss:

$$\mathcal{L}_\Theta(q, d^+, d^-) = max(0, 1 - s(q, d^+) + s(q, d^-)), \tag{2}$$

where d^+ is a document ranked higher than d^- given a query q, Θ represent the system parameters to be optimized and $s(\cdot)$ is the function that computes the matching scores.

Evaluation Measures. For this reproducibility experiment we consider the same evaluation measures adopted in the DRMM reference paper [35]: *Mean Average Precision at Rank 1000 (MAP)*, *Normalized Discounted Cumulated Gain at Cutoff 20 (nDCG@20)* and *Precision at Cutoff 20 (P@20)*.

Model configuration. The authors of DRMM share the input data and the implementation of the system in a public repository. However, in order to replicate the original results and to generalize DRMM to other collections, we defined a new document pre-processing pipeline and developed a new training script for the word embeddings required by the retrieval model. We considered only the best configuration of the model with the LCH matching histograms with 30 bins and the IDF coefficients in the term gating network. Finally, we set the first feed-forward layer size to 5. This is the default configuration of the network as described in the reference paper [35].

Moreover, since DRMM performs a re-ranking of a pre-existing run, we compute it using a modified version of Terrier v.4.1 [53] [6] with the *Query Likelihood Model* (QLM) with Dirichlet smoothing [89], the INQUERY stop list [13] and Krovetz stemmer [47] as done in the reference paper [35]. We do not perform any optimization of the parameters of the model and keep the Terrier default smoothing value $\mu = 2500$. We also keep this experimental setup for all the experiments with DRMM throughout the paper.

The most critical element among the inputs of DRMM is the set of word embeddings of terms in the corpus. The authors share a pre-trained Word2Vec model [55] which contains the required word embeddings trained on the collections used in their original experiments. However, to replicate their experiments on the Robust04 collection – and to use DRMM on other collections – we train a new Word2Vec model following their instructions with the Word2Vec implementation available in Gensim. The model was trained as described in Section 2, in the embedding based evaluation paragraph. The Word2Vec hyperparameters are the same as the ones indicated in the reference paper. However, differently from what stated in [35], we experimented with different number of training epochs, and employed the gradient decay option available in the Word2Vec Gensim implementation. The corpus on which we evaluate the model is the Robust04 collection, where we consider the complete set of 249 topics. Finally, we evaluate DRMM considering the *title* (T) and the *description* (D) fields as done in the experiments of the reference paper [35].

[6] The modified version of Terrier we used for our experiments is available at the url: `https://github.com/gridofpoints`.

	Robust04 (T)			Robust04 (D)		
	MAP	nDCG@20	P@20	MAP	nDCG@20	P@20
QLM (original)	0.253	0.415	0.369	0.246	0.391	0.334
QLM (reproduced)	0.248 (-0.005)	0.415 (0.000)	0.355 (-0.014)	0.246 (0.000)	0.392 (+0.001)	0.326 (-0.008)
DRMM (original)	**0.279**	0.431	**0.382**	**0.275**	**0.437**	**0.371**
DRMM (Gensim WE)	0.268 (-0.011)	0.441 (+0.010)	0.376 (-0.006)	0.249 (-0.026)	0.411 (-0.026)	0.343 (0.028)
DRMM (original WE)	0.270 (-0.009)	**0.442** (+0.011)	0.377 (-0.005)	0.252 (-0.023)	0.415 (-0.022)	0.347 (-0.024)

Table 3: Results of our reproducibility experiments of DRMM. **Bold** values represent the highest scores among the models. We report between the parentheses the difference between the reproduced results and those reported in the DRMM reference paper – QLM (original) and DRMM (original).

Experimental Results. In Table 3, we report the experimental results we obtain using the shared DRMM code with our document pre-processing pipeline. In this experiment we also used (i) the Word2Vec embeddings trained with Gensim on the Robust04 collection (Gensim WE), and (ii) the word embeddings model shared in the official DRMM repository (Original WE). [7] We also compare our performance to those of the QLM model implementation in Terrier and the shared results in the DRMM reference paper [35]. From the results reported in Table 3, we observe that our document pre-processing pipeline and the strategy we adopted for the training of the word embeddings leads to a similar performance to the one reported in the DRMM reference paper [35].

In general, we obtain slightly lower MAP, nDCG@20 and P@20 values than the ones reported in [35] when considering only the *title* (T) or *description* (D) fields alone, with a larger difference in the second case. Interestingly, if we repeat the experiments considering both the title and description (TD) fields of the topics, we obtain better rankings than the ones computed using only the title or description – in this case, we obtain MAP, nDCG@20 and P@20 values respectively of 0.279, 0.451 and 0.386.

Given our experimental results, we consider our document pre-processing pipeline and word embedding training strategy as reliable enough to reproduce the DRMM performance.

Discussion. From the results we obtained in these reproducibility experiments, we found a sizeable impact of the word embeddings quality on the performance of the whole system. We highlight that [35] lacks a sufficiently detailed description of the embedding training process for the replication of the results. In fact, we had to perform numerous experiments with different combinations of parameters in order to train a word embedding model comparable to the one in [35]. We first performed the training of the word embedding model with the official Word2Vec package shared by Google and the hyperparameter configuration indicated in Section 2. In this case, we obtained a MAP value of 0.249 on Robust04 (T). Afterwards, we tried the Word2Vec implementation available in Gensim – the one we later decided to adopt for all of our experiments – that with the same hyperparameters configuration led to a MAP of 0.268 in the same experiment.

For these reasons, we conclude that sharing the training script for the word embeddings model and specifying the library used are important requirements to reproduce the results of a NeuIR system.

Another issue we encountered in the reproducibility process concerns the preparation of the other input data required by DRMM. The shared implementation of the algorithm requires seven files: a run in TREC format to be re-ranked; a word embedding model to be used by the system; a file containing the document and corpus frequency for each term in the collection; a file containing each document of the corpus with its identifier (the same used in the run to rerank), its length, and the frequency of each term in it; a file with the ideal discounted cumulative gain value for each considered topic; a file with the list of terms for each topic along with the topic identifier (the same used in the run to re-rank); the relevance judgments in TREC format for the given topics and the documents in the collection. However, the authors do not share a tool or describe, with enough detail, the process employed to compute the input in this format. Therefore, a few assumptions about the preprocessing steps to be applied to the collection were necessary. For instance, simply whitespace tokenizing, applying stemming and removing stopwords from the collection, as indicated in the reference paper, leads to the addition of great noise to the input of the system. For this reason, we first extract the text from the TREC-formatted documents, remove the tags, remove all punctuation, and finally

[7] `https://github.com/faneshion/DRMM`.

perform the tokenization, stemming and stopwords removal. Nonetheless, we obtain about $0.9M$ distinct terms compared to the $0.7M$ terms present in the shared input file. Therefore, applying more aggressive preprocessing steps on the collection might lead to further performance improvements of DRMM.

In conclusion, we believe that for the authors of NeuIR research papers it would be a a good practice to share the scripts and the libraries used during document pre-processing, and, when required, also for the training of word embeddings models. This would allow a better and more straightforward reproducibility of their experiments. In fact, the pre-processing step is often underspecified, despite its sizeable impact on the global performance of NeuIR models.

4. NVSM: Reproducibility of an unsupervised neural model

Model description. The Neural Vector Space Model (NVSM) [74] extends two unsupervised representation learning models for expert [75] and product [72] search to *ad-hoc* retrieval. NVSM jointly learns distinct word and document representations by optimizing an unsupervised loss function which minimizes the distance between sequences of n words (i.e. n-grams) and the documents containing them. Such optimization objective imposes that n-grams extracted from a document should be predictive of that document. Unlike the models from which it derives, NVSM integrates a notion of *term specificity* [68, 70] in the learning process of word and document representations. In fact, while optimizing the n-gram representations to be close to the corresponding documents, words that are discriminative for the target documents learn to contribute more to the n-gram representations. Therefore, words associated with many documents will be neglected due to low predictive power. After training, the learned word and document representations are used to perform retrieval. Queries are seen as n-grams and matched against documents in the feature space. Documents are then ranked in decreasing order of the cosine similarity computed between query and document representations. [8] A detailed description of NVSM components follows.

Given a document collection D and a word vocabulary V, the model considers the vector representations $\vec{w_i} \in \mathbb{R}^{k_w}$ and $\vec{d_j} \in \mathbb{R}^{k_d}$ for $w_i \in V$ and $d_j \in D$, respectively, where k_w and k_d denote the dimensionality of word and document representations. Due to the different dimensionality of word representations $\vec{w_i}$ and document representations $\vec{d_j}$, the model requires a transformation $f : \mathbb{R}^{k_w} \rightarrow \mathbb{R}^{k_d}$ from the word feature space to the document feature space. The considered transformation is linear:

$$f(\vec{x}) = W\vec{x}, \tag{3}$$

where $\vec{x}$ is a k_w-dimensional vector and W is a $k_d \times k_w$ parameter matrix that is learned using gradient descent. A sequence of n words (i.e. an n-gram) $(w_{j,i})_{i=1}^{n}$ extracted from a document d_j is obtained by averaging its constituent word representations as follows:

$$g\left((w_{j,i})_{i=1}^{n}\right) = \frac{1}{n}\sum_{i=1}^{n} \vec{w_{j,i}}. \tag{4}$$

Representations of words and documents are learned using mini-batches B of m n-gram/document pairs such that an n-gram representation is projected close to the document that contains it.

During training, an auxiliary function that L2-normalizes a vector of arbitrary dimensionality is further introduced:

$$norm(\vec{x}) = \frac{\vec{x}}{||\vec{x}||} \tag{5}$$

Therefore, the projection of an n-gram into the k_d-dimensional document feature space can be written as the following composition function:

$$\tilde{T}((w_{j,i})_{i=1}^{n}) = (f \circ norm \circ g)((w_{j,i})_{i=1}^{n}). \tag{6}$$

[8] Note that NVSM performs retrieval and then ranking on the whole document collection.

By estimating the per-feature sample mean and variance over batch B, the standardized projection of the n-gram representation is obtained as follows:

$$T((w_{j,i})_{i=1}^{n}) = \text{hard-tanh}\left(\frac{\tilde{T}((w_{j,i})_{i=1}^{n}) - \hat{\mathbb{E}}[\tilde{T}((w_{j,i})_{i=1}^{n})]}{\sqrt{\hat{\mathbb{V}}[\tilde{T}((w_{j,i})_{i=1}^{n})]}} + \beta\right). \tag{7}$$

The n-gram representation is optimized to be close to the corresponding document. The composition function g in combination with the L2-normalization *norm* causes words to compete for contributing to the resulting n-gram representation. Therefore, words that are discriminative for the target document learn to contribute more to the n-gram representation, and consequently, the L2-norm of the representations of discriminative words is larger than the L2-norm of non-discriminative words. This incorporates a notion of term specificity into the model. Moreover, standardization forces n-gram representations to distinguish themselves solely in the dimensions that matter for matching.

The similarity of two representations in latent vector space is defined as:

$$P(\mathcal{S}|d_j, (w_{j,i})_{i=1}^{n}) = \sigma(\vec{d_j} \cdot T((w_{j,i})_{i=1}^{n})), \tag{8}$$

where $\sigma(t) = \frac{1}{1+\exp(-t)}$ denotes the sigmoid function and $\mathcal{S}$ is a binary indicator that states whether the representation of document d_j is similar to the projection of its n-gram $(w_{j,i})_{i=1}^{n}$ or not. The probability of a document d_j, given its n-gram $(w_{j,i})_{i=1}^{n}$, is then approximated by uniformly sampling z contrastive examples [37]:

$$\log \tilde{P}(d_j|(w_{j,i})_{i=1}^{n})) = \frac{z+1}{2z}\left(z \log P(\mathcal{S}|d_j, (w_{j,i})_{i=1}^{n})) + \sum_{\substack{k=1,\\ d_k \sim U(D)}}^{z} \log(1.0 - P(\mathcal{S}|d_k, (w_{j,i})_{i=1}^{n})))\right), \tag{9}$$

where $U(D)$ represents the uniform distribution over documents D used to obtain contrastive examples. Then, the loss function used to optimize the model, averaged over the instances in batch B, is:

$$L(\theta|B) = -\frac{1}{m}\sum_{j=1}^{m} \log \tilde{P}(d_j|(w_{j,i})_{i=1}^{n}) + \frac{\lambda}{2m}\Big(\sum_{i=1}^{|V|} ||\vec{w_i}||_2^2 + \sum_{j=1}^{|D|} ||\vec{d_j}||_2^2 + ||W||_F^2\Big), \tag{10}$$

where θ is the set of parameters $\{\vec{w_i}\}_{i=1}^{|V|}$, $\{\vec{d_j}\}_{j=1}^{|D|}$, W, β and λ is a weight regularization hyperparameter.

After training, a query q is projected into the document feature space by the composition of f and g: $(f \circ g)(q) = h(q)$. The matching score between a document d_j and a query q is given by the cosine similarity between their representations in document feature space:

$$score(q, d_j) = \frac{h(q)^\intercal \cdot \vec{d_j}}{||h(q)||_2 \, ||\vec{d_j}||_2}. \tag{11}$$

Documents are ranked in decreasing order of $score(q, d_j)$ for a query q.

Evaluation measures. In this reproducibility experiment, we employ the same measures reported in [74] to evaluate retrieval effectiveness: MAP, *Normalized Discounted Cumulated Gain at Cutoff 100 (nDCG@100)*, and *Precision at Cutoff 10 (P@10)*. To test statistical significance, we perform a two-tailed paired Student's t-test between Word2Vec (si) and NVSM as in [74].

Model configuration. We re-implement NVSM from scratch in Python, relying on widely-used and consolidated libraries. We employ Whoosh, [9] a fast Python search engine library, to index document collections.

[9] `https://whoosh.readthedocs.io/en/latest/`.

Whoosh allows easy access to the underlying tokenized document collections, providing the same functionalities as `pyndri` [76] which was used in the reference paper. During indexing and querying, stopwords are removed using the Indri stoplist [10] and no stemming is performed. We implement the NVSM architecture using TensorFlow.

One of the biggest challenges we found to reproduce the results presented in [74] lies in the choice of the model's parameters and hyperparameters. The reference paper and the associated GitHub repository [11] lack a comprehensive description of all the parameters and hyperparameters used by NVSM, which are crucial to reproduce results. Therefore, to select the parameters and hyperparameters used in this study, we relied on the reference paper [74], the authors' GitHub repository and an additional paper [72] on which [74] is based on. [12] For each setting, the reference source(s) are reported below.

Word vocabulary:

- Vocabulary size is limited to 2^{16} words (GitHub repository: /scripts/functions.sh, [72]), or to 60,000 words (NVSM reference paper [74], gihub repository: /cpp/main.cu);
- Words containing numbers are not considered (GitHub repository: /cpp/main.cu, [72]);
- Words with a document frequency lower than 2 and greater than 50% are not considered (GitHub repository: /cpp/main.cu).

Model parameters:

- Pseudo-random number generator seed equal to 0 (GitHub repository: /cpp/main.cu);
- The number of batches for a single epoch is computed as $\lceil \frac{1}{m} \sum_{d \in D} (|d| - n + 1) \rceil$, where m is the batch size (NVSM reference paper [74]);
- Word representations, document representations and the parameter matrix W are uniformly sampled in the range $[-\sqrt{\frac{6.0}{m+n}}, \sqrt{\frac{6.0}{m+n}}]$ for an $m \times n$ matrix – following the initialization scheme presented in [34] (GitHub repository, [72]).

Model hyperparameters:

- Word representation dimension $k_w = 300$ (NVSM reference paper [74]);
- Document representation dimension $k_d \in \{64, 128, 256\}$ (NVSM reference paper [74]);
- n-gram size $n \in \{4, 6, 8, 10, 12, 16, 24, 32\}$ (NVSM reference paper [74]);
- Batch size $m = 51200$ (NVSM reference paper [74]);
- The number of epochs to train the model is 15 (NVSM reference paper [74]);
- Number of negative examples $z = 10$ (NVSM reference paper [74]);
- Adam optimizer with parameters $\alpha = 0.001, \beta_1 = 0.9, \beta_2 = 0.999$ (NVSM reference paper [74] reports only α, [72] reports also β_1 and β_2);
- Regularization $\lambda = 0.01$ (NVSM reference paper [74]).

For each collection considered (see Section 2), the given set of topics is split into validation and test sets [13] and only the *title* (T) field is used. The size of document representations $k_d \in \{64, 128, 256\}$ and of the n-grams $n \in \{4, 6, 8, 10, 12, 16, 24, 32\}$ is optimized on the validation set and then employed on the test set. The optimal hyperparameters combinations, are not reported in the reference paper.

In our experiments, we set the vocabulary size to 2^{16} and we select $k_d = 256$ according to the results reported in Fig. 3 of the reference paper [74]. Similarly, for each considered collection, we chose the n-gram size that provides the best scores in terms of MAP. The rest of the parameters and hyperparameters are kept as above. We train the model for 15 epochs and we select the model iteration that performs best in terms of MAP on the validation set. The best model is then evaluated on the test set. Table 4 shows the comparison between the results obtained with the NVSM original version and our reproduced version.

[10] `http://www.lemurproject.org/stopwords/stoplist.dft`.

[11] `https://github.com/cvangysel/cuNVSM/`.

[12] It is worth mentioning that relevant information is scattered across all these sources and none of them provide an off-the-shelf description of all the required settings.

[13] Splits can be found at: `https://github.com/cvangysel/cuNVSM/tree/master/resources/adhoc-splits`.

		AP88-89 (T)			FT (T)			LA (T)		
		MAP	nDCG@100	P@10	MAP	nDCG@100	P@10	MAP	nDCG@100	P@10
QLM(jm)	original	0.199	0.346	0.365	0.218	0.356	0.283	0.182	0.331	0.221
	reproduced	0.199	0.346	0.364	0.209	0.337	0.258	0.178	0.319	0.214
	diff.	0.000	0.000	+ 0.001	+ 0.009	+ 0.019	*+ 0.025*	+ 0.004	+ 0.012	+ 0.007
QLM(dir)	original	0.216	0.370	0.392	**0.240**	**0.381**	**0.296**	**0.198**	**0.348**	**0.239**
	reproduced	0.217	0.368	0.397	**0.230**	**0.362**	**0.270**	**0.198**	**0.341**	**0.233**
	diff.	- 0.001	+ 0.002	- 0.005	+ 0.01	+ 0.019	*+ 0.026*	0.000	+ 0.007	+ 0.006
LDA	original	0.039	0.077	0.078	0.009	0.028	0.013	0.004	0.015	0.010
	reproduced	0.052	0.091	0.077	0.013	0.026	0.015	0.007	0.028	0.015
	diff.	- 0.013	- 0.014	+ 0.001	- 0.004	+ 0.002	- 0.002	- 0.003	- 0.013	- 0.005
W2V(add)	original	0.216	0.370	0.393	0.125	0.230	0.195	0.105	0.212	0.159
	reproduced	0.234	0.395	0.416	0.140	0.252	0.214	0.075	0.165	0.116
	diff.	- 0.018	*- 0.025*	*- 0.023*	- 0.015	*- 0.022*	- 0.019	*+ 0.030*	*+ 0.047*	*+ 0.043*
W2V(si)	original	0.230	0.383	0.418	0.141	0.250	0.204	0.131	0.242	0.179
	reproduced	0.240	0.400	0.419	0.148	0.261	0.226	0.109	0.215	0.172
	diff.	- 0.010	- 0.017	- 0.001	- 0.007	- 0.011	*- 0.022*	*+ 0.022*	*+ 0.027*	+ 0.007
NVSM	original	**0.257****	**0.418****	**0.425**	0.172**	0.302***	0.239*	0.166**	0.300***	0.209*
	reproduced	**0.257**	**0.414**	**0.429**	0.175**	0.304***	0.220	0.180***	0.316***	0.208**
	diff.	0.000	+ 0.004	- 0.004	- 0.002	- 0.002	+ 0.019	- 0.014	- 0.016	+ 0.001
		NY (T)			Robust04 (T)			WSJ (T)		
		MAP	nDCG@100	P@10	MAP	nDCG@100	P@10	MAP	nDCG@100	P@10
QLM(jm)	original	0.158	0.270	0.376	0.201	0.359	0.369	0.175	0.315	0.345
	reproduced	0.180	0.292	0.382	0.199	0.351	0.358	0.178	0.319	0.347
	diff.	*- 0.022*	*- 0.22*	- 0.006	+ 0.002	+ 0.008	+ 0.011	- 0.003	- 0.004	- 0.002
QLM(dir)	original	**0.188**	**0.318**	**0.486**	**0.224**	**0.388**	**0.415**	0.204	**0.351**	**0.398**
	reproduced	**0.213**	**0.343**	**0.500**	**0.222**	**0.376**	**0.411**	0.205	0.355	0.391
	diff.	*- 0.025*	*- 0.025*	- 0.014	+ 0.002	+ 0.012	+ 0.004	- 0.001	- 0.004	+ 0.007
LDA	original	0.009	0.027	0.022	0.003	0.010	0.009	0.038	0.082	0.076
	reproduced	0.024	0.053	0.064	0.004	0.015	0.014	0.041	0.074	0.060
	diff.	- 0.015	*- 0.026*	*- 0.042*	- 0.001	- 0.005	- 0.005	- 0.003	+ 0.008	+ 0.016
W2V(add)	original	0.081	0.160	0.216	0.075	0.177	0.194	0.175	0.322	0.372
	reproduced	0.100	0.196	0.252	0.065	0.158	0.184	0.181	0.326	0.393
	diff.	- 0.019	*- 0.036*	*- 0.036*	+ 0.010	+ 0.019	+ 0.010	- 0.006	- 0.004	*- 0.021*
W2V(si)	original	0.092	0.173	0.220	0.093	0.208	0.234	0.185	0.330	0.391
	reproduced	0.113	0.209	0.284	0.083	0.192	0.214	0.190	0.336	0.393
	diff.	*- 0.021*	*- 0.036*	*- 0.064*	+ 0.010	+ 0.016	*+ 0.020*	- 0.005	- 0.006	- 0.002
NVSM	original	0.117	0.208	0.296*	0.150***	0.287***	0.298***	**0.208****	**0.351**	0.370
	reproduced	0.110	0.205	0.290	0.138***	0.270***	0.289***	**0.213*****	**0.359*****	**0.408**
	diff.	+ 0.007	+ 0.003	+ 0.006	+ 0.012	+ 0.017	+ 0.009	- 0.005	- 0.008	*- 0.038*

Table 4: Result comparison between original versions of QLM, Word2Vec, and NVSM (as in [74]) and their reproduced versions. For each experimental collection, the first row reports the scores of the original model version on MAP, nDCG@100 and P@10, the second row reports the scores of the reproduced version and the third row reports the difference between original and reproduced versions; a negative difference indicates that the reproduced baselines are stronger than those employed in the reference paper. **Bold** values represent the best model (original and reproduced), whereas *italic* values represent differences greater than 0.02. A two-tailed paired Student's t-test is computed between Word2Vec (si) and NVSM. Statistical significance is marked as * for $p < 0.1$, ** for $p < 0.05$ and *** for $p < 0.01$.

Baseline configuration. As semantic baselines, we reproduce Word2Vec (add), Word2Vec (si) and LDA. Following [74], we rely on Gensim to implement Word2Vec (skip-gram architecture) and LDA. For Word2Vec approaches, we adopted the same choices made for NVSM regarding word vocabulary, seed value, negative examples, one-sided window size (i.e., n-gram size $n/2$) and number of epochs. The embedding size is set to 256 to be consistent with NVSM. Documents are ranked in decreasing order of cosine similarity between the document representation and the average of the word embeddings in the query. Once again, we selected the model iteration that performs best in terms of MAP on the validation set and we evaluated it on the test set. For LDA, we set the number of topics $K = 256$ and $\alpha = \beta = 0.1$. The model is trained until topic convergence is achieved. At query time, documents are ranked in decreasing order of the cosine similarity between the query topic distribution and the document topic distribution.

Regarding lexical baselines, we have reproduced QLM (jm) and QLM (dir). As in [74], we relied on Indri [71] to index and query the collections considered. To be consistent with semantic models, stopwords are removed using the Indri stoplist and no stemming is performed. Smoothing hyperparameters $\lambda \in \{x | k \in \mathbb{N}^+, k \leq 20, x = k/20\}$ and $\mu \in \{125, 250, 500, 750, 1000, 2000, 3000, 4000, 5000\}$ for QLM (jm) and QLM

(dir), respectively, are optimized on the validation set. The comparison between the results obtained with the original baselines and our reproduced versions is also shown in Table 4.

Rank Fusion configuration. We also reproduce the fusion of three individual rankers, that is QLM (dir), Word2Vec (si) and NVSM, that provide a mixture of lexical and semantic matching. In [74], the combination of individual rankers is performed through a grid search on the weights of a linear combination using 20-fold cross validation on the topic test sets. Feature weights are swept between 0.0 and 1.0 with increments of 0.0125 on the fold training set. Individual features are normalized per query so that their values lie between 0 and 1. The coefficients configuration that achieves the highest MAP on the fold training set is selected and used to score the fold test set. During scoring of the fold test set, the pool of the top-1k documents ranked by the individual rankers is used as candidate set.

Due to the extensive memory/time requirements demanded by such approach, [14] we have to limit the number of documents in each fold of the training set to the pool of the top-1k documents produced by the individual rankers. In addition, feature weights are swept between 0.0 and 1.0 with increments of 0.1, which is the minimum step we could support on our machine. Along with the supervised approach presented in [74], we also employ three classic, fast rank fusion methods, first proposed in [31]: CombSUM, CombMNZ and CombANZ.

In Table 6, we evaluate the results obtained with our version of the supervised approach and the three classic methods on the Robust04 collection. The best approach in terms of MAP is then employed on all the considered collections and compared to the supervised approach of the reference paper, as shown in Table 7.

Experimental Results. In Table 4, we present a comparison between the versions of QLM, LDA, Word2Vec and NVSM described in the reference paper and our reproduced versions. For each collection, we report the results from the reference paper, the results obtained with our reproduced versions and the difference among them.

If we consider the lexical baselines, we see that the original and the reproduced versions present similar values, for both QLM (jm) and QLM (dir), on AP88-89, LA, Robust04 and WSJ – where an absolute difference greater than 0.01 is achieved by QLM (dir) only in the nDCG@100 on Robust04, and by QLM (jm) in the nDCG@100 and P@10 on LA and Robust04, respectively. A larger difference between the two versions of QLM (jm) and QLM (dir) can be observed on the FT and NY collections. The results obtained on FT show a marked difference between the original and the reproduced versions of QLM (jm) and QLM (dir) for nDCG@100 and P@10. The original version of QLM (jm) outperforms the reproduced version by an absolute difference of 0.019 in nDCG@100 and of 0.025 in P@10, whereas the original version of QLM (dir) outperforms the reproduced version by an absolute difference of 0.019 in the nDCG@100 and of 0.026 in the P@10. An opposite behavior is observed on the NY collection, where the reproduced versions of QLM (jm) and QLM (dir) outperform the original versions in all measures, with an absolute difference greater than 0.02 for the MAP and nDCG@100. Overall, QLM (dir) achieves the best results for all measures on FT, LA, NY, Robust04 and, only for the original version, for the nDCG@100 and P@10 on WSJ.

Regarding semantic baselines, the reproduced version of LDA performs similarly to the original version on all the collections considered. The most notable exceptions are the nDCG@100 and P@10 values on the NY collection, where the reproduced LDA model outperforms the original one with an absolute difference of 0.026 and 0.042, respectively. Concerning Word2Vec-based models, the reproduced version of Word2Vec (add) outperforms the original version on AP88-89, FT, WSJ, and NY according to all of the performance measures, whereas the original version achieves better results on LA and Robust04. In relation to performance difference, a more marked gap is found between the two versions only on the LA and NY collections, with differences of +0.030 in MAP, +0.047 in nDCG@100 and +0.043 in P@10 on LA, and of −0.036 in nDCG@100 and −0.036 in P@10 on NY. A similar trend can also be observed for Word2Vec (si), where the reproduced version outperforms the original version on AP88-89, FT, WSJ, and NY, while the original

[14] The approach could not finish the first training fold, due to OOM, when considering the entire document collection and an incremental step of 0.0125. The machine used to run the experiments is a 2018 Alienware Area-51 with 36 cores and 64Gb of RAM.

version achieves better results on LA and Robust04. The absolute differences between the two versions are lower than or close to 0.02 on all collections, except for the NY and LA. In particular, the absolute differences are higher than 0.02 for all measures on the NY collection, with a difference of −0.064 for P@10 – which is the highest (absolute) difference amongst all measures and collections. It is also worth mentioning that, by achieving a score of 0.284 for P@10 on the NY collection, the reproduced version of Word2Vec (si) closes the gap with NVSM and results in a competitive neural baseline.

Regarding NVSM, the results obtained with the reproduced version are close to those reported in the reference paper. Considering the performance difference, the only measure that presents an absolute difference greater than 0.02 is P@10 on the WSJ collection (−0.038). On AP88-89 and NY, the absolute differences are lower than 0.01 for all the measures. NVSM outperforms the semantic baselines on all the considered collections except on the NY, where Word2Vec (si) shows better performance in MAP and nDCG@100. Furthermore, NVSM achieves the best results overall on AP88-89 and WSJ.

The results reported in Table 4 indicate that we successfully reproduced NVSM and that the baselines adopted in the original paper are generally aligned with their reproduced versions. The optimal n-gram size and the epoch at which we obtain the best model for NVSM are indicated in Table 5, for each collection considered.

In Table 6, we present the results obtained with the reproduced version of the supervised approach and the three classic methods: CombSUM, CombMNZ and CombANZ, applied on the Robust04 collection. For each combination of QLM (dir) with Word2Vec (si) and NVSM, we can observe that the reproduced version of the supervised approach performs consistently worse than its original version. Since the two versions of QLM (dir) perform similarly, it indicates that we could not successfully reproduce the supervised rank fusion approach presented in [74]. Overall, the best method is CombSUM and the worst is CombANZ. Therefore, we employ CombSUM on all the considered collections and compare its performance to the original supervised rank fusion approach in Table 7.

In Table 7, we can observe that AP88-89 is the only collection where CombSUM achieves similar results as those of the supervised approach – with differences between the two versions of QLM(dir)+Word2Vec(si)+ NVSM of +0.015 for MAP, +0.005 for nDCG@100 and +0.005 for P@10. Overall, CombSUM shows positive performance gains in all the combinations of the individual rankers on AP88-89, FT and WSJ. The performance gains on LA are positive for QLM(dir)+NVSM and QLM(dir)+Word2Vec(si)+NVSM, whereas they are negative for QLM(dir)+Word2Vec(si). In particular, the CombSUM version of QLM(dir)+NVSM consistently outperforms the original supervised approach on all the measures. On Robust04, the only combination that improves over the baseline is QLM(dir)+NVSM. However, when compared to the original supervised approach, the CombSUM version of QLM(dir)+NVSM presents a performance gap of about 0.020 on all the measures. Regarding the NY collection, each combination of individual rankers using CombSUM achieves lower performances than the baseline. In particular, the difference between the original supervised approach and the CombSUM version of QLM(dir)+Word2Vec(si)+NVSM is +0.100 for P@10.

	n-gram size	Best epoch
AP88-89	16	13
FT	16	11
LA	10	10
WSJ	16	14
Robust04	16	11
NY	16	1

Table 5: NVSM optimal n-gram size and best epoch for each collection.

Discussion. The results reported in Table 4 illustrate that the reproduced and original versions of NVSM perform similarly. Therefore, based on the parameter and hyperparameter choices presented in the model configuration, we can reproduce the results of the reference paper. However, we have had to look into different sources to get appropriate values, since the configuration data are scattered across the reference paper, the GitHub repository and a previous paper [72]. Furthermore, the lack of information regarding which are the best hyperparameters to use, for the results reported in Table 2 of the reference paper, has led

		Robust04 (T)		
		MAP	nDCG@100	P@10
QLM(dir)	original	0.224	0.388	0.415
	reproduced	0.222	0.376	0.411
QLM(dir)+W2V(si)	original	0.232	0.399	0.428
	reproduced	0.190	0.344	0.346
	CombSUM	**0.199**	**0.358**	**0.378**
	CombMNZ	0.191	0.353	0.366
	CombANZ	0.179	0.325	0328
QLM(dir)+NVSM	original	0.247	0.411	0.448
	reproduced	0.204	0.357	0.375
	CombSUM	**0.230**	0.391	**0.424**
	CombMNZ	0.229	**0.392**	0.419
	CombANZ	0.199	0.352	0.373
QLM(dir)+W2V(si)+NVSM	original	0.247	0.412	0.446
	reproduced	0.1836	0.336	0.344
	CombSUM	**0.206**	**0.368**	**0.385**
	CombMNZ	0.202	0.363	0.373
	CombANZ	0.175	0.323	0.338

Table 6: Result comparison on the Robust04 collection between our version of the supervised rank fusion approach, presented in [74], and the three classic methods originally proposed in [31]: CombSUM, CombMNZ and CombANZ. The first two rows report, for reference, the scores of the original and the reproduced versions of QLM (dir) for MAP, nDCG@100 and P@10. Then, for each combination of QLM (dir) with Word2Vec (si) and NVSM, the first two rows report the scores of the original and the reproduced versions of the supervised approach. Subsequent rows report the scores of CombSUM, CombMNZ and CombANZ, respectively. **Bold** values represent the best method amongst the supervised approach (reproduced) and the three classic methods.

us to identify such hyperparameters differently. We relied on Fig. 3 from the reference paper, which allowed us to identify the optimal choices and to obtain results with the reproduced version of NVSM that are close to those reported in [74].

Another critical aspect when reproducing NVSM is the lexicon. For example, when considering the Robust04 collection, NVSM does not retrieve any document for the following four topics: topic 312 "Hydroponics"; topic 316 "Polygamy Polyandry Polygyny"; topic 348 "Agoraphobia" and topic 379 "mainstreaming". If we analyze the content of such topics, we see that three out of four topics are composed of a single word. Therefore, since the word vocabulary does not contain any of those query terms, NVSM cannot retrieve any document for them. To faithfully reproduce results, it is pivotal to know the exact lexicon used in the original paper. Otherwise, we cannot identify whether the differences between the original and the reproduced versions are related to implementation nuances or different preprocessing steps.

Regarding lexical baselines, we relied on the same search engine, i.e. Indri, and performed the same operations reported in [74] for indexing, querying and hyperparameters optimization. The differences between the original and the reproduced versions of QLM (jm) and QLM (dir) might lie on the different tokenization process applied to topics. In fact, we relied on Indri for both indexing and querying, whereas in [74] `pyndri` is used to perform querying.

Regarding semantic baselines, for LDA, we followed the same configuration presented in [74]. However, the Gensim implementation of LDA presents many more parameters than those mentioned in the reference paper. Thus, not knowing which values to assign to these parameters prevents the reproducibility of the results. For what concerns Word2Vec-based retrieval approaches, we adopt the same hyperparameter values of NVSM on every collection. The resulting versions of Word2Vec (si) and Word2Vec (add) present sizable differences compared to the original ones. Most likely, our hyperparameter choices are different than those used in [74], especially if we consider the results obtained with the reproduced versions on the NY collection. However, in the reference paper, there is no description of the optimal choices found for Word2Vec, nor any

		AP88-89 (T)			FT (T)		
		MAP	nDCG@100	P@10	MAP	nDCG@100	P@10
QLM(dir)	original	0.216	0.370	0.392	0.240	0.381	0.296
	reproduced	0.217	0.368	0.397	0.230	0.362	0.270
QLM(dir)+W2V(si)	original	0.279 (+29%)	0.437 (+18%)	0.450 (+14%)	0.251 (+4%)	0.393 (+3%)	0.313 (+6%)
	CombSUM	0.275 (+27%)	0.441 (+20%)	0.446 (+12%)	0.242 (+5%)	0.381 (+5%)	0.293 (+9%)
	diff.	+ 0.004	- 0.004	+ 0.004	+ 0.009	+ 0.012	*+ 0.020*
QLM(dir)+NVSM	original	0.289 (+33%)	0.444 (+20%)	0.473 (+20%)	0.251 (+4%)	0.401 (+5%)	0.322 (+9%)
	CombSUM	0.269 (+24%)	0.428 (+16%)	0.456 (+15%)	0.233 (+1%)	0.378 (+4%)	0.286 (+6%)
	diff.	*+ 0.020*	+ 0.016	+ 0.017	+ 0.018	*+ 0.023*	*+ 0.036*
QLM(dir)+W2V(si)+ NVSM	original	**0.307** (+42%)	**0.466** (+26%)	**0.498** (+27%)	**0.258** (+7%)	**0.406** (+6%)	**0.322** (+9%)
	CombSUM	**0.292** (+35%)	**0.461** (+25%)	**0.493** (+24%)	**0.244** (+6%)	**0.386** (+7%)	**0.297** (+10%)
	diff.	+ 0.015	+ 0.005	+ 0.005	+ 0.014	*+ 0.020*	*+ 0.025*
		LA (T)			NY (T)		
		MAP	nDCG@100	P@10	MAP	nDCG@100	P@10
QLM(dir)	original	0.198	0.348	0.239	0.188	0.318	0.486
	reproduced	0.198	0.341	0.233	**0.213**	**0.343**	**0.500**
QLM(dir)+W2V(si)	original	0.212 (+7%)	0.360 (+3%)	0.236 (-1%)	0.206 (+9%)	0.333 (+4%)	0.494 (+1%)
	CombSUM	0.191 (-4%)	0.326 (-4%)	0.229 (-2%)	0.194 (-9%)	0.325 (-5%)	0.436 (-13%)
	diff.	*+ 0.021*	*+ 0.034*	+ 0.007	+ 0.012	+ 0.008	*+ 0.058*
QLM(dir)+NVSM	original	0.220 (+11%)	0.376 (+7%)	0.244 (+1%)	**0.222** (+18%)	**0.355** (+11%)	0.520 (+6%)
	CombSUM	**0.232** (+17%)	**0.381** (+12%)	**0.255** (+9%)	0.198 (-7%)	0.333 (-3%)	0.476 (-5%)
	diff.	- 0.012	- 0.005	- 0.011	*+ 0.024*	*+ 0.022*	*+ 0.044*
QLM (dir)+W2V(si)+ NVSM	original	**0.226** (+14%)	**0.378** (+8%)	**0.250** (+4%)	**0.222** (+18%)	0.353 (+10%)	**0.526** (+8%)
	CombSUM	0.214 (+8%)	0.366 (+7%)	0.251 (+7%)	0.182 (-14%)	0.320 (-7%)	0.426 (-15%)
	diff.	+ 0.012	+ 0.012	- 0.001	*+ 0.040*	*+ 0.033*	*+ 0.100*
		Robust04 (T)			WSJ (T)		
		MAP	nDCG@100	P@10	MAP	nDCG@100	P@10
QLM(dir)	original	0.224	0.388	0.415	0.204	0.351	0.398
	reproduced	0.222	0.376	0.411	0.205	0.355	0.391
QLM(dir)+W2V(si)	original	0.232 (+3%)	0.399 (+2%)	0.428 (+2%)	0.254 (+24%)	0.410 (+16%)	0.454 (+13%)
	CombSUM	0.199 (-10%)	0.358 (-5%)	0.378 (-8%)	0.238 (+16%)	0.399 (+12%)	0.449 (+15%)
	diff.	*+ 0.033*	*+ 0.041*	*+ 0.050*	+ 0.016	+ 0.011	+ 0.005
QLM(dir)+NVSM	original	**0.247** (+10%)	0.411 (+6%)	**0.448** (+7%)	0.248 (+21%)	0.396 (+12%)	0.425 (+6%)
	CombSUM	**0.230** (+4%)	**0.391** (+4%)	**0.424** (+3%)	0.244 (+19%)	0.403 (+14%)	0.443 (+13%)
	diff.	+ 0.017	*+ 0.020*	*+ 0.024*	+ 0.004	- 0.007	- 0.018
QLM(dir)+W2V(si)+ NVSM	original	**0.247** (+10%)	**0.412** (+6%)	0.446 (+7%)	**0.271** (+32%)	**0.426** (+21%)	**0.456** (+14%)
	CombSUM	0.206 (-7%)	0.369 (-2%)	0.385 (-6%)	**0.251** (+22%)	**0.416** (+17%)	**0.455** (+16%)
	diff.	*+ 0.041*	*+ 0.043*	*+ 0.061*	*+ 0.020*	+ 0.010	+ 0.001

Table 7: Result comparison between CombSUM rank fusion [31] and the supervised rank fusion proposed in [74]. For each experimental collection, the scores of the original and the reproduced versions of QLM (dir) on MAP, nDCG@100 and P@10 are reported for reference. For each of the three combinations of QLM (dir) with Word2Vec (si) and NVSM, the first row reports the original scores of the supervised rank fusion while the second row reports the scores of the CombSUM rank fusion. The third row illustrates the difference between the original scores of the supervised approach and the scores obtained with CombSUM; a negative difference indicates that CombSUM achieves higher scores than those of the supervised approach used in the reference paper. **Bold** values represent the best method (original and reproduced), whereas *italic* values represent differences greater than 0.02. The percentage gain (or loss) over the baseline method is reported next to each rank fusion approach.

figure that allows us to identify a subset of candidate choices. Moreover, the same considerations that are true for NVSM about the word vocabulary are also true for Word2Vec methods. Nevertheless, the results of the two-tailed paired Student's t-tests between the reproduced versions of Word2Vec (si) and NVSM are consistent with those presented in [74]. The only notable variations are in AP88-89 – where no statistical difference is found between the reproduced versions of Word2Vec (si) and NVSM – and in WSJ – where for nDCG@100 there is a statistical difference between the two reproduced versions.

Regarding rank fusion, the main issues relate to the extensive memory/time requirements of the supervised rank fusion approach. The choices we made to reproduce it were insufficient to obtain comparable results as those presented in [74]. On the other hand, classic and fast rank fusion techniques like CombSUM produced mixed results, which tended to even worsen the performance of the QLM (dir) baseline on some collections. Nevertheless, the trade-off between effectiveness and efficiency provided by CombSUM shows that far less expensive fusion methods can be employed to improve performance, achieving, on some collections, performance gains similar to those reported in [74].

5. Comparison with other lexical and semantic retrieval models

In this section, we compare the performance of NVSM and DRMM (re-ranking over QLM (dir)) to other lexical and semantic models considering the Robust04 (T) and NY (T) collections. Other retrieval models we consider in this comparison are: QLM (dir), DFR, BM25, TF-IDF (all with Krovetz stemmer) and Word2Vec (add and self-information versions). We also employ NVSM to re-rank the same QLM (dir) runs used by DRMM in its experiments where the objective is to investigate how well a neural unsupervised model like NVSM performs within a re-ranking scenario. Finally, we perform a statistical significance analysis with Tukey's T test to assess the statistical significance of performance differences of the retrieval models. The evaluation measures used in this set of experiments are: MAP, nDCG@100 and P@10.

Experimental results. The results in Table 8 show that the lexical models considered perform better than all the Word2Vec-based approaches or the NeuIR models on the NY collection. On the other hand, we also notice that DRMM outperforms both the lexical models and NVSM on the Robust04 collection. Instead, NVSM always performs better than the semantic matching models, but is never competitive with other lexical models nor with DRMM. In the re-ranking task, NVSM outperforms DRMM on the NY collection, but not on the Robust04. Furthermore, we observe that performing re-ranking with NVSM on QLM (dir) always leads to a performance improvement over NVSM alone. However, the re-ranking produced by NVSM significantly worsens the performance of the QLM (dir) baseline on both collections.

	NY (T)			Robust04 (T)		
	MAP	nDCG	P@10	MAP	nDCG	P@10
QLM(dir)	0.232	0.370	**0.522**	0.248	0.411	0.424
BM25	**0.234**	0.347	0.468	0.242	0.404	0.431
TF-IDF	0.228	**0.499**	0.472	0.242	0.405	0.431
DFR	0.225	0.350	0.478	0.227	0.390	0.425
Word2Vec (add)	0.100	0.196	0.252	0.062	0.150	0.175
Word2Vec (si)	0.113	0.209	0.284	0.081	0.185	0.205
DRMM	0.141	0.211	0.274	**0.268**	**0.435**	**0.455**
NVSM	0.110	0.205	0.290	0.139	0.269	0.279
QLM(dir)/NVSM	0.152	0.252	0.328	0.157	0.289	0.287

Table 8: Results comparison between reproduced versions of DRMM, NVSM and QLM(dir)/NVSM and other lexical and neural baselines: DRM, BM25, TF-IDF, QLM and Word2Vec. For each experimental collection used, the scores on MAP, nDCG at rank 100 AND P@10 are reported for each model. **Bold** values represent the highest scores among the models.

For reference, we report the best values for MAP and P@10 obtained in the TREC 2004 Robust Retrieval Track [78] on the Robust04 collection, and the best value for MAP in the TREC 2017 Common Core Track [4, 77] on the NY collection. On Robust04 (T), the best value for MAP is 0.333 and for P@10 is 0.513. On NY, the best value for MAP is 0.538.

Significance tests. In Figure 2, we report the results of Tukey's T test on the runs produced by the systems in Table 8. Firstly, we notice that on the NY collection there is no statistical difference between considered lexical models (i.e. QLM (dir), DFR, TF-IDF and BM25). Additionally, pure lexical models always perform statistically better than semantic models (i.e. Word2Vec (add), Word2Vec (si), QLM(dir)/NVSM, DRMM and NVSM). If we consider the tests on the Robust04 collection, we observe that DRMM belongs to the top group and statistically outperforms the other semantic matching models, including the re-ranking of the QLM (dir) run performed by NVSM – i.e., when NVSM performs the same task as DRMM. However, it is worth mentioning that QLM(dir)/NVSM is an unsupervised re-ranking model, whereas DRMM is a supervised one. Therefore, while NVSM performs re-ranking relying only on word and document representations learned from the collection, DRMM exploits relevance judgments to learn how to rank documents given a query. These observations apply to all the performance measures considered – i.e. MAP, nDCG@100, and P@10.

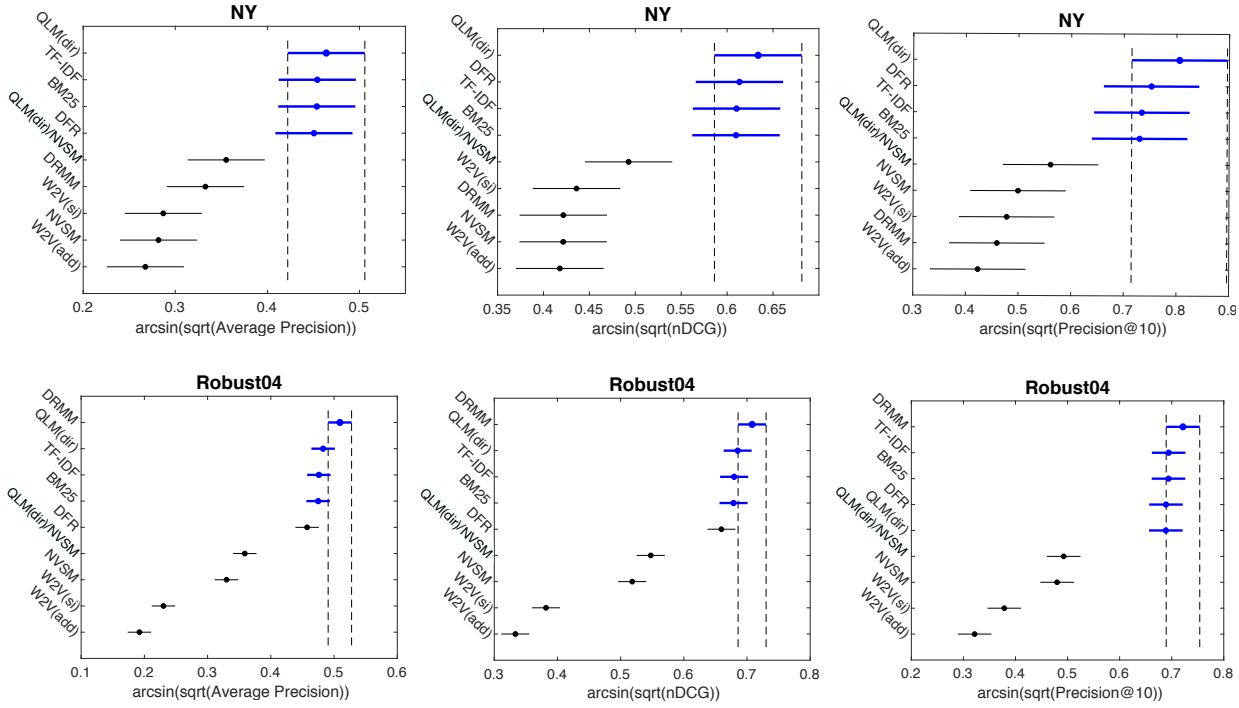


Figure 2: Significance tests for the results reported in Table 8 (top group highlighted). All pairwise comparisons are calculated with Tukey's HSD confidence intervals and a significance level $\alpha = 0.05$. Each row depicts the comparisons made for MAP, NDCG at rank 100 and P@10 for a specific collection.

Discussion. From the results reported in Table 8, our first observation has to do with the performance of DRMM which differs in the two considered collections. In fact, DRMM improves the ranking of QLM (dir) and it is the best system on the Robust04 collection; but, it worsens the ranking computed by the QLM (dir) baseline, provided in input on the NY collection. This result reflects one of the weaknesses of supervised NeuIR models: i.e., their inability to generalize when trained on a limited number of topics. Compared to Robust04, which presents 249 topics, the NY collection has only 50 topics – about one fifth of the topics of Robust04. Therefore, on the NY collection DRMM suffers from the lack of topics to learn from, compared to the large size of the collection. In fact, the large number of documents – combined with the intrinsic characteristics of the collection and the queries – generates a wide variety of matching signals to be interpreted by the model. This makes it harder for DRMM to learn how to discriminate between relevant and non-relevant documents without seeing larger parts of the collection during training. This claim is also confirmed by our experiments on the WT2g collection in Section 6. Indeed, the WT2g collection has the same number of topics of the NY collection, but contains seven times less documents. This difference in size reduces the effect of the lack of training topics on DRMM which performs similarly to other lexical models on this collection. Conversely, on the Robust04 collection, where the model is trained on a larger number of topics, DRMM outperforms all the other baselines. On the other hand, the re-ranking performed by NVSM always worsens the performance of the QLM (dir) baseline. This suggests that NVSM does not effectively re-rank the results provided by a lexical model like QLM (dir). As shown in Table 7, a rank fusion approach that combines lexical and semantic signals together is better suited for NVSM than a simple re-ranking approach.

When considering other state-of-the-art approaches that use different ranking strategies – such as [85], an application of BERT to IR that achieves a MAP of 0.328 on the Robust04, or the BM25+RM3 model

in Anserini [84] that achieves a MAP of 0.2903 [15] – we have observed that DRMM and NVSM are not competitive with such approaches. However, these two NeuIR models remain relevant as they are unsupervised (NVSM) and supervised (DRMM) innovative neural architectures for document retrieval which do not rely on any auxiliary data source or component, like relevance feedback or query expansion. Indeed, the advancement proposed by DRMM and NVSM is more methodological rather than performance oriented. Nevertheless, the combination of DRMM and NVSM with pre-trained NLP models, such as BERT, or other IR techniques, such as RM3, may still lead to better performance than state-of-the-art.

6. Collection-based evaluation

In this set of experiments, we evaluate the ability of DRMM and NVSM models to generalize in the presence of different domains and tasks. The considered collections and topic fields include: CLEF-DE (TD), CLEF-FA (TD), CLEF-IT (TD), WT2G (TD), and OHSUMED (D). We select the above combinations of topic fields, since they are the ones that lead to the best retrieval performance on each collection and the most widely used. Additionally, we also compare the performance of DRMM and NVSM to those of lexical IR models, that is BM25, DFR, TF-IDF and QLM (dir). To perform retrieval on CLEF multilingual collections, all the considered models rely on publicly available stop lists specifically defined for the target language. [16] The evaluation measures we consider for this set of experiments are: MAP, nDCG@100, and P@10. Furthermore, we rely on post-hoc Tukey's T test to assess statistical significance.

Hyperparameter tuning. The performance measures of DRMM are obtained according to the steps described in Section 2, where a re-ranking of the documents in each QLM (dir) run is performed. The main hyperparameters of DRMM are the number of bins in the matching histograms and the size of the hidden layer. The number of bins in the matching histograms should be high enough to allow the representation of different degrees of word similarity. At the same time, it should be small enough to maintain the generalization power of the model and keep computational complexity low.

We performed hyperparameter optimization on the OHSUMED and WT2g collections – which are the corpora that differ the most from the ones used in the DRMM reference paper – to assess whether the changes to the mentioned hyperparameters could improve the performance of the model. We considered all the combinations of the following hyperparameters: number of bins in the matching histograms, in the range {5, 10, 15, 20, 25, 30, 35} and hidden layer size, in the range {5, 10, 15, 20, 25}. In our experiments, we noticed a performance improvement – an increase of 0.02 or higher in MAP, nDCG@100 and P@10 – only on the WT2g collection with 20 bins in the matching histograms, and 15 units in the hidden layer. We will keep this configuration of DRMM also for the other experiments on this collection. On the other hand, we did not notice any sizable performance increase on the OHSUMED collection changing the hyperparameters of the DRMM model.

Regarding NVSM, we adopted the same experimental setup described in Section 4. For every collection, we optimize the following hyperparameters:

- Vocabulary size $|V| \in \{2^{16}, 2^{17}\}$;
- Document representation dimension $k_d \in \{128, 256\}$;
- n-gram size $n \in \{4, 6, 8, 10, 12, 16, 24, 32\}$;
- Batch size $m \in \{12800, 25600, 51200\}$;
- Regularization $\lambda \in \{0.01, 0.1, 1.0\}$.

The rest of the hyperparameters are kept as in Section 4. Due to the prohibitive time required to perform grid search over the hyperparameters, we first optimize the n-gram size by keeping the default values for $|V| = 2^{16}$, $k_d = 256$, $m = 51200$, and $\lambda = 0.01$. Then, for each collection we keep the n-gram size that performs best in terms of MAP and we optimize the rest of the hyperparameters. From this optimization we did not detect any improvement related to different combinations of k_d, m, and λ. However, the impact

[15] `https://github.com/castorini/anserini/blob/master/docs/experiments-robust04.md`.
[16] `http://members.unine.ch/jacques.savoy/clef`.

of the vocabulary size has shown to be significant on two collections: WT2g and CLEF-DE. For WT2g, NVSM goes from MAP: 0.206, nDCG@100: 0.356, and P@10: 0.370, with $|V| = 2^{16}$, to MAP: 0.225, nDCG@100: 0.380, and P@10: 0.402, with $|V| = 2^{17}$. Similarly, for CLEF-DE the model goes from MAP: 0.194, nDCG@100: 0.322, and P@10: 0.281, with $|V| = 2^{16}$, to MAP: 0.211, nDCG@100: 0.343, and P@10: 0.301, with $|V| = 2^{17}$. Therefore, we adopt $|V| = 2^{17}$ for WT2g and CLEF-DE collections and we keep the rest of the hyperparameters as default. The optimal n-gram size, vocabulary size, and the epoch at which we obtain the best model for NVSM are reported in Table 9 for every collection.

Experimental results. Table 10, illustrates the retrieval results related to the WT2g and OHSUMED collections. In such case, we first observe that DRMM always improves the ranking of the runs produced with QLM (dir). We also employed DRMM to re-rank the runs obtained with BM25 and TF-IDF, but we found no substantial improvement over the baselines. Conversely, NVSM performances are the worst overall on WT2g and outperform only those of QLM (dir) on OHSUMED. In particular, the performance gap between NVSM and all the other models is more accentuated on WT2g than on OHSUMED. This may be due to the fact that NVSM derives from two approaches that are specifically tailored to product and expert search [72, 75]. Therefore, the model has a robust domain-specific nature and for heterogeneous collections like WT2g, where documents have different contents and scopes, it generalizes worse than for homogeneous collections like OHSUMED. Indeed, NVSM achieves better results than QLM (dir) on OHSUMED for all the evaluation measures considered.

For reference, we report the best value for MAP obtained in the TREC-8 Web Track [41] on the WT2g collection, and the best value for MAP obtained on the OHSUMED collection according to [66]. These are 0.383 and 0.450 on WT2g and OHSUMED, respectively.

In Table 11, we report the experimental results on CLEF collections. If we consider the performances of DRMM and compare them to QLM (dir), we do not observe the same trend for all the experimental collections. DRMM outperforms QLM (dir) on CLEF-IT and CLEF-FA collections, whereas it worsens the performance of QLM (dir) on CLEF-DE. Overall, the performance of DRMM and the lexical baselines follow the same trend of the previous set of experiments, where BM25 and TF-IDF are the two best baselines. The results of NVSM also represent a similar behavior as those reported for the two previous collections when compared to BM25, TF-IDF, and DFR. Nevertheless, NVSM shows competitive results with QLM (dir) on CLEF-IT and CLEF-DE, and it outperforms QLM (dir) on CLEF-FA. When compared to DRMM, NVSM achieves lower results on CLEF-IT and CLEF-FA, but it outperforms DRMM on CLEF-DE.

For reference, we report the best values for MAP obtained in the CLEF 2006 Ad Hoc Track [22] on the CLEF-IT and CLEF-DE collections, and the best value for MAP obtained in the CLEF 2009 Ad Hoc Track [29] on the CLEF-FA collection. On CLEF-IT (TD) and CLEF-DE (TD), the best values for MAP are 0.419 and 0.483, respectively. On CLEF-FA (TD), the best value for MAP is 0.494.

	n-gram size	Vocabulary	Best epoch
WT2g	16	131072	11
OHSUMED	16	65536	12
CLEF-IT	20	65536	14
CLEF-DE	8	131072	13
CLEF-FA	16	65536	15

Table 9: NVSM optimal n-gram size, vocabulary size, and best epoch for each collection.

Significance Tests. From Tukey's T tests conducted with a significance level $\alpha = 0.05$ reported in Figure 3, we observe a similar behavior as in the results reported in Tables 10 and 11. NVSM is never in the top group – except when we consider the P@10 on the CLEF-FA collection – while DRMM is in the top group for most of the collections (i.e. CLEF-FA, WT2g, and OHSUMED). Regarding lexical models, BM25, TF-IDF and DFR are almost always the best performing systems, while QLM (dir) is in general the worst performing one – often being out of the top group (e.g. in the CLEF-FA, CLEF-IT, and OHSUMED collections).

	WT2g (TD)			OHSUMED (D)		
	MAP	nDCG	P@10	MAP	nDCG	P@10
QLM(dir)	0.264	0.418	0.418	0.212	0.326	0.305
BM25	**0.296**	0.454	**0.484**	0.250	0.369	0.364
TF-IDF	0.234	0.389	0.419	0.250	0.370	0.364
DFR	0.267	0.426	0.452	0.236	0.354	0.344
DRMM	0.289	**0.455**	0.434	**0.272**	**0.407**	**0.375**
NVSM	0.225	0.380	0.402	0.214	0.335	0.319

Table 10: Generalization experiments results on the Web (WT2g) and medical (OHSUMED) domain. The highest score amongst the models is in **bold**. The considered evaluation measures are MAP, nDCG@100 and P@10.

	CLEF-IT (TD)			CLEF-DE (TD)			CLEF-FA (TD)		
	MAP	nDCG	P@10	MAP	nDCG	P@10	MAP	nDCG	P@10
QLM(dir)	0.334	0.510	0.319	0.209	0.350	0.324	0.200	0.340	0.405
BM25	0.437	0.627	0.402	**0.253**	**0.395**	**0.373**	0.408	0.550	0.594
TF-IDF	**0.438**	**0.628**	**0.408**	0.251	0.392	0.367	0.411	0.553	0.601
DFR	0.415	0.605	0.393	0.241	0.382	0.350	**0.421**	**0.565**	0.619
DRMM	0.357	0.557	0.349	0.188	0.329	0.316	0.375	0.551	**0.623**
NVSM	0.345	0.522	0.317	0.211	0.343	0.301	0.342	0.494	0.567

Table 11: Generalization experiments results on CLEF collections in Italian (IT), German (DE) and Persian (FA). The highest score amongst the models is in **bold**. The considered evaluation measures are MAP, nDCG@100 and P@10.

Discussion. From the experimental results described above, we can conclude that DRMM does not always statistically improve the ranking produced by the QLM (dir) baseline. The only collections where there is a statistically significant improvement of DRMM over QLM (dir) are CLEF-FA and OHSUMED. The same behavior can be found when DRMM is trained and used to re-rank runs produced by different lexical models (i.e., TF-IDF and BM25), where it often fails to produce a substantial improvement. For these reasons, it is unclear whether the improvement brought by DRMM in the re-ranking task is more attributable to DRMM itself, or to the lexical model used to compute the input run – which might be good at finding relevant documents but bad at ranking them. In fact, we observe that DRMM tends to achieve competitive performances only when those of QLM (dir) are particularly low compared to other lexical models, e.g. on the OHSUMED and CLEF-FA collections. On the other hand, if we consider WT2g or CLEF-DE, where QLM (dir) achieves performances closer to those of other lexical models, the re-ranking of DRMM is less effective or even detrimental. Indeed, DRMM worsens the QLM (dir) baseline in all measures on CLEF-DE. However, the overall results on CLEF-DE – compared to those obtained on the other CLEF collections – suggest that CLEF-DE is a more difficult collection to perform retrieval on. Nevertheless, we cannot exclude that, if more topics were provided in the training step, DRMM would improve the ranking of QLM (dir) also on this collection.

Regarding NVSM, we observe that it is not competitive with traditional lexical models. The only exception is QLM (dir), that achieves results that are comparable to or lower than NVSM on the OHSUMED and CLEF collections. Results from Tables 10 and 11 show that NVSM struggles to generalize when considering heterogeneous data. The reason why NVSM suffers more with the WT2g collection than on OHSUMED and CLEF collections may be related to its inherent domain-specific nature. Indeed, NVSM derives from [72, 75], which are two approaches targeting product and expert search, respectively.

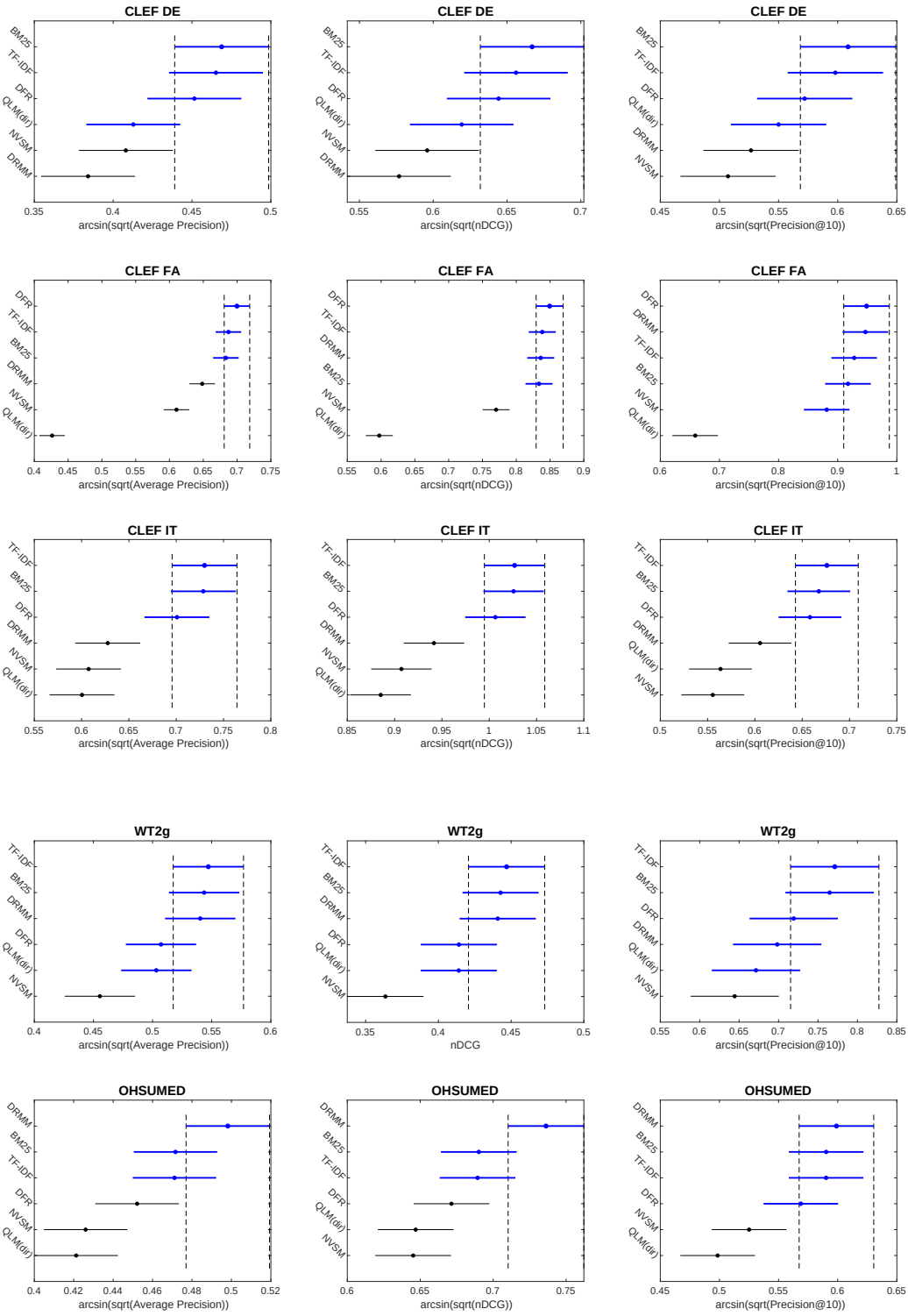


Figure 3: Significance tests for the results reported in Table 10 and Table 11 (top group highlighted). All pairwise comparisons are calculated with Tukey's HSD confidence intervals and a significance level $\alpha = 0.05$. Each row depicts the comparisons made for MAP, nDCG@100 and P@10 for a specific collection.

7. Embedding-based evaluation

This section, evaluates the effect of different word embedding models on DRMM. We performed the evaluation on the Robust04 (T), NY (T), WT2g (TD), and OHSUMED (D) collections.

In fact, DRMM allows the usage of different word embedding models, regardless of their characteristics. Differently from NVSM, which learns its own words and documents representation, DRMM does not learn any representation. But rather, it learns to interpret the interactions between each document and query term, given a certain word representation. For this reason, we have investigated here how sensitive DRMM can be to different word embedding models.

We consider the word embedding models described in Section 2: (i) Word2Vec word embeddings computed with the implementation available in Gensim (W2V), (ii) FastText word embeddings obtained again with the FastText implementation from Gensim (FastText), (iii) the word embeddings obtained with NVSM (NVSM Embeddings), (iv) a set of word embeddings trained by Google on a Google News corpus. All the models, except the last one, have been trained on the experimental collections on which we later performed the retrieval experiments. We also report the performance of NVSM on each collection as a baseline.

Experimental results. From the results reported in Table 12, we observed the robustness of DRMM when using different embedding models. Indeed, there is no sizeable difference between the Gensim Word2Vec and the FastText embeddings on the Robust04, NY and OHSUMED collections. However, on the NY and WT2g collections FastText embeddings lead to a slightly higher performance. In general, none of the embedding models considered outperforms the others on all the collections and for all the measures. However, we observe that the Word2Vec embeddings trained by Google lead to the lowest performances overall – despite the significantly larger corpus used to train such model. Conversely, the word embeddings generated by NVSM outperform all the others on Robust04 and OHSUMED in MAP and P@10.

	Robust04 (T)			NY (T)			WT2g (TD)			OHSUMED (D)		
	MAP	NDCG	P@10	MAP	NDCG	P@10	MAP	NDCG	P@10	MAP	NDCG	P@10
NVSM	0.139	0.269	0.279	0.141	0.211	0.274	0.225	0.380	0.402	0.214	0.335	0.319
DRMM (W2V)	0.268	**0.435**	0.455	0.141	0.211	0.274	0.289	0.455	0.434	0.272	**0.407**	0.375
DRMM (FastText)	0.262	0.427	0.456	**0.153**	**0.226**	**0.276**	**0.309**	**0.471**	**0.468**	0.268	0.402	0.368
DRMM (NVSM WE)	**0.270**	0.434	**0.458**	0.130	0.207	0.232	0.294	0.458	0.442	**0.273**	0.405	**0.397**
DRMM (W2V Google)	0.261	0.428	0.455	0.126	0.215	0.268	**0.309**	0.455	0.460	0.238	0.361	0.360

Table 12: Comparison between NVSM and DRMM using with different word embedding models: Word2Vec trained on the experimental collection (W2V), FastText trained on the experimental collection (FastText), NVSM word embeddings trained on the experimental collection (NVSM WE), pre-trained Word2Vec model by Google on Google News (W2V Google). The considered evaluation measures are MAP, nDCG@100 and P@10.

Discussion. The results in Table 12 suggest that DRMM has the tendency to learn to match documents leveraging more on strong exact matching signals or collection-based co-occurrence signals (NVSM WE) rather than latent semantic ones (W2V or FastText). Otherwise, DRMM would have achieved better retrieval results using a Word2Vec model trained on a very large corpus – like Google News – compared to an embedding model trained on each experimental collection. Indeed, we expect a model trained on a much larger corpus to be better at representing semantic relations between words.

Interestingly, the results obtained with the embeddings generated by NVSM (NVSM WE) – trained to model term specificity along with co-occurrence relations between terms – are the best on Robust04 and OHSUMED. This result suggests that NVSM embeddings are better at modeling the collection-based co-occurrence relations between terms than Word2Vec-based approaches. This also leads to more useful matching signals that may be used to perform retrieval on the collection they are trained on. For DRMM, we can therefore conclude that matching signals based on collection-based co-occurrence relations have a more positive effect on performances than semantic ones.

Also, since there is no relevant performance difference between all the considered embedding models, we can conclude that DRMM learns to assign more importance to exact matching signals – associated to a

separate bin in the histograms received in input (see Section 3) – than the other ones. Indeed, as shown in [65], word embedding models such as Word2Vec or FastText, learn very different word representations and produce different matching scores for the same terms. Hence, since the component with less variability in the matching histograms is the one associated to exact matching signals and the performance of the model is very similar for different embedding models, we conclude that DRMM learns to rely more on this dimension of the input.

8. Topic-based evaluation

In this section, we perform a topic-by-topic analysis of the runs of NVSM and DRMM, comparing their per-topic AP to that obtained with BM25 on the Robust04 (T), NY (T), WT2g (TD), OHSUMED (D) and CLEF (TD) collections.

We also employ *Kernel Density Estimation (KDE)* [81] to estimate the PDF of the APs of BM25, NVSM and DRMM for all topics.

Discussion. In Figure 4, we present the scatter plots of the per-topic AP of NVSM, DRMM and BM25 on different collections. For each collection, we compare each model with the others.

We begin to consider the results on the Robust04 collection. If we compare DRMM and BM25, we observe that most of the points are close to the bisector. This indicates that, in general, the two models have a similar performance on most of the topics. On the other hand, the comparison between NVSM and BM25, reveals that the majority of the points is concentrated below the bisector, meaning that BM25 outperforms NVSM on most of the topics. If we analyze the results of the comparison between NVSM and DRMM, we observe a similar point distribution as in the previous chart, once again indicating, on this collection, DRMM's greater effectiveness over NVSM.

Considering the results on NY, DRMM reveals a worse performance than BM25 on most of the topics, and so does NVSM. However, in comparing the two NeuIR models, we see that DRMM outperforms NVSM on a large number of topics. The large performance difference between DRMM and BM25 is likely due to the fact that the NY collection only has 50 topics, which are not enough for DRMM to learn a good ranking model as done in other collections. However, since DRMM re-ranks on a set of 2000 documents previously retrieved with QLM (dir), it has an advantage over NVSM which is also confirmed by its performance difference. Examining the documents retrieved by NVSM in greater detail, and comparing them with the ones returned by BM25, we noted that NVSM can return relevant documents which do not contain any query term. For instance, on topic 442 ("heroic acts") NVSM returns a relevant document – with doc id "1036498" – which does not contain any query term and is not returned by BM25. In this case, terms like "heroism" and "sacrifices" are used in the documents instead of those of the query. Similar situations for other topics are observable in other collections (i.e. OHSUMED).

On CLEF collections, we observe similar relative performances as in the two previous collections. DRMM performed overall similarly to BM25 except for a few topics where it outperforms it, i.e., topic 200 in CLEF-DE, and topic 148 in CLEF-IT; or vice versa, where BM25 performs better by a large margin, i.e., topics 141, 149, and 161 on CLEF-DE, topic 628 on CLEF-FA and topics 161 and 44 on CLEF-IT. On the other hand, NVSM is in general outperformed by BM25 on all the collections for most of the topics. This time however, the difference is less marked than on the Robust04 and NY collections. Finally, if we compare DRMM and NVSM on the same set of collections, we observe a performance difference of the two models for a large number of topics mostly on the CLEF-IT collection – where the points in the scatterplot are in general further from the bisector – while we observe a similar performance on the other two collections in the majority of the topics with just a few outliers where NVSM outperforms DRMM by a larger margin on CLEF-DE.

On the WT2g collection, we observe that DRMM is performing once again in the majority of the topics in a very similar way to BM25 – with the exception of topics 423 and 410 where BM25 outperforms DRMM by a large margin. NVSM on the other hand is outperformed by the two other models on the vast majority of the topics. However, we also observe that in a handful of topics, DRMM and NVSM both manage to outperform BM25. This happens for topic 416: "Three Gorges Project What is the status of The Three

Gorges Project?". In this case, the documents containing the topic keywords are very long and BM25 does not recognize many of them as relevant.

Finally, on OHSUMED, we observe a similar relative performance of the models as on the Robust04 collection. Where DRMM is performing on all topics in a similar way as BM25, and NVSM is in general outperformed by the two other models in most of the topics. Here, as in the NY collection, there are some topics, for which NVSM returns relevant documents which do not contain any query term. This is the case for topic OHSU7 ("lactase deficiency therapy options") and document with doc id "91359745" which contains none of the query terms but only their synonyms or closely related terms (i.e. "lactose" and "intolerance").

Overall, we can conclude that DRMM and BM25 have a very similar performance across all topics, with a few outliers only on CLEF-DE and NY, while NVSM is usually outperformed by both BM25 and DRMM on most of the topics and collections. If we consider the differences between NVSM and the two other models we also observe that these are always larger than DRMM and BM25. This is likely due to the fact that DRMM performs a re-ranking on the top 2000 documents retrieved with QLM (dir) – a lexical model like BM25. For this reason, we expect the rankings of NVSM to contain a more diverse set of documents than the ones in the runs produced by DRMM or BM25 since NVSM relies solely on the neural model without explicitly considering exact term matches.

Considering the charts in Figure 5, we can assess the differences in the systems' behavior that is otherwise not identifiable when observing the average measures across all topics. In this case, we see that on the Robust04, NY, CLEF-DE, WT2g and OHSUMED collections, the MAP value is highly influenced by the low performances of the systems on a large number of topics – this situation is especially true for NVSM. In fact, in all these charts there is a high peak associated to low AP values. Conversely, the experiments on the CLEF-IT, CLEF-FA collections, highlight that the final MAP value is dominated by a large number of topics where the systems obtain an AP around 0.3 – except for NVSM which also has a high peak close to 0.1.

The quantitative distance in the distribution of AP values across different topics can be measured considering the KLD between the AP distributions reported in Figure 5. Table 13 illustrates the distances between the AP distribution associated to the BM25, NVSM and DRMM models.

	Robust04	NY	CLEF-DE	CLEF-FA	CLEF-IT	OHSUMED	WT2g
BM25 - DRMM	10.86	63.29	21.76	18.58	14.48	16.91	27.08
BM25 - NVSM	20.57	40.69	28.15	22.70	26.65	49.60	61.22
DRMM - NVSM	31.52	23.85	24.06	16.45	18.69	71.66	76.29

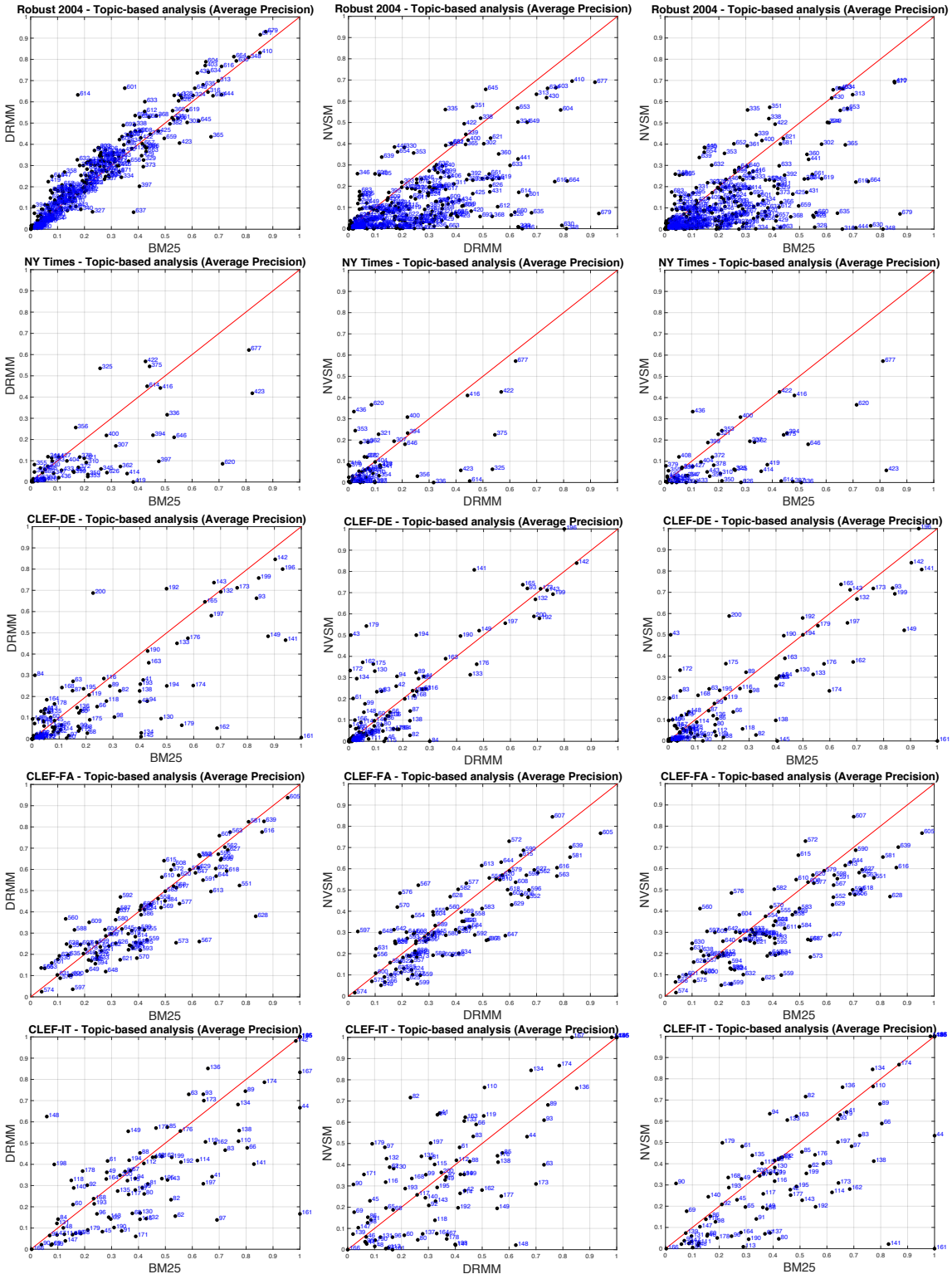
Table 13: KLD values between the PDF of the AP values obtained with BM25, NVSM and DRMM on the Robust04, NY, CLEF-DE, CLEF-FA, CLEF-IT, OHSUMED, and WT2g collections. KLD $\in [0, +\infty)$, it denotes the divergence between the two distributions [12]; therefore, 0 means that the two models behave in the same way for all the topics in the collection; $+\infty$ means that for no topic the two models behave in the same way.

Considering the distances among the distributions on the Robust04, CLEF-DE, CLEF-IT, OHSUMED and WT2g collections, we notice that DRMM and BM25 are often the closest systems. This indicates that DRMM and BM25 have a comparable number of topics where they obtain a similar AP. Worth noting that the set of topics with a similar AP might be different since we are only considering the distribution of the AP values, ignoring any corresponding topic ids. On the NY and CLEF-FA collections instead, DRMM and NVSM are the two closest systems, implying that they have a similar number of topics with comparable AP values. This also corresponds to the results reported in Figure 5, where both models show very high peaks associated to AP values around 0.1 and 0.2 on the NY and CLEF-FA collections, respectively.

In conclusion, the main advantage observed of the considered NeuIR models is the ability to retrieve documents which do not contain any query term. Moreover, in a few cases we observed that NeuIR models have the ability to rank higher than lexical models very long documents containing relatively short relevant passages related to a query. Nevertheless, we did not find enough empirical evidence to strongly support this claim.

Finally, NeuIR systems do not outperform in general traditional lexical approaches such as BM25. It

can however be noticed that DRMM is competitive with lexical models on certain collections (Robust04, WT2g and OHSUMED), and outperforms BM25 on a few topics.



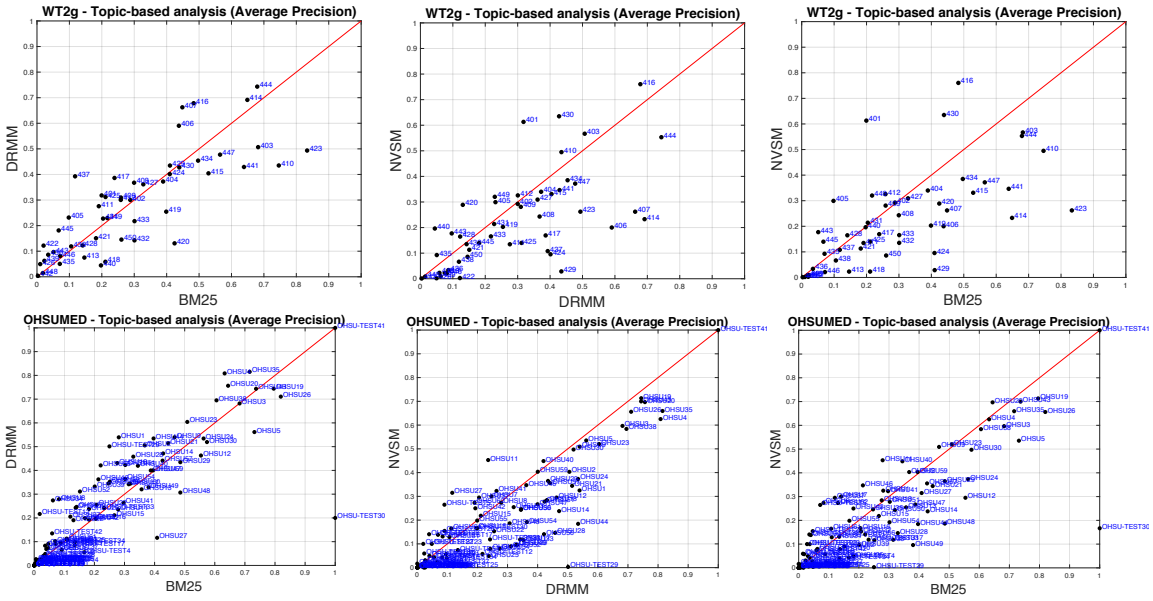


Figure 4: Scatter plots of the AP for each topic of Robust04, NY, CLEF-DE, CLEF-FA, CLEF-IT, WT2g and OHSUMED collections, obtained with DRMM, NVSM and BM25 retrieval models.

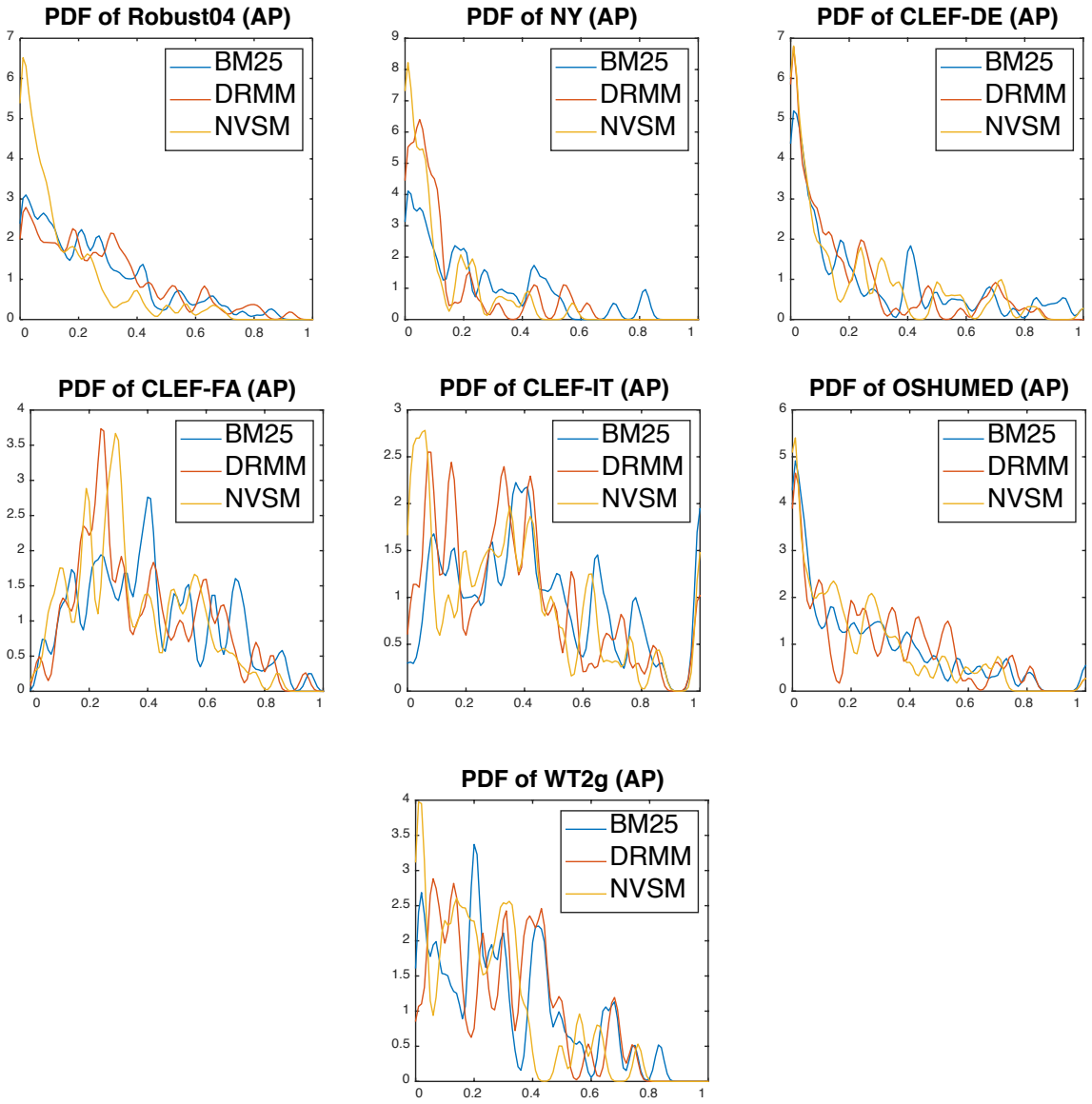


Figure 5: PDF of the AP relative to WT2g, OHSUMED, CLEF-IT, CLEF-DE, and CLEF-FA collections, of NVSM and BM25 retrieval models. Note that on the x-axis there are the AP values distributed into 100 bins (from 0 to 1 with 0.01 step) and on the y-axis the density estimation.

9. Related work

In the past few years, the increased availability of data and the success of deep neural networks in the NLP field, has promoted the diffusion of neural models even in the IR field. Existing NeuIR approaches can be classified into *representation-* and *interaction*-based models [59].

The aim of representation-based models is to learn how to represent a query and a document and then use it to estimate their similarity. Within this class, one of the earliest neural models is the *Deep Structured Semantic Model (DSSM)*. DSSM is trained by maximizing the conditional likelihood of clicked documents given a query using click-through data. The model employs word hashing, which allows DSSM to scale up and handle large vocabularies which are common in large-scale Web search applications. Architectural variants of this model are *Convolutional Latent Semantic Model (CLSM)* [87] and *LSTM Deep Structured Semantic Model (LSTM-DSSM)* [60]. On the other hand, recent advances in building unsupervised low-dimensional text representations [14, 49, 54, 63] has led the IR community to consider them for retrieval. We can identify two main lines of work: (i) approaches that incorporate features from neural language models [3, 33, 36, 90] and (ii) methods that learn representations of words and documents from scratch and use them directly for retrieval [72, 74, 75, 79]. Within (i), the effectiveness of Doc2Vec (PV-DBOW architecture) is evaluated for *ad-hoc* retrieval in [3]. The *Generalized Language Model (GLM)* is presented in [33] where the mutual independence between a pair of words no longer holds and word embeddings are used to derive the transformation probabilities between words. A similar idea is proposed in [90] where word embeddings are used to estimate probabilities in a translation model that is combined with traditional retrieval models. In [36], a semantic matching model based on the *Bag-of-Word-Embeddings (BoWE)* representation is introduced. The model represents every document as a matrix of the word embeddings occurring in it and the matching between queries and documents is seen as a non-linear word transportation problem. Within (ii), the approach presented in [79] proposes to compose document representations as the weighted sum of their word embeddings using the self-information [16] value of each word as its weighting operator. The idea is that IDF-inspired weights assign more importance to words bearing more information content during the compositional process. The works presented in [72, 75] introduce an unsupervised end-to-end representation learning model for product and expert search, respectively. The NVSM model [74] extends [72, 75] to *ad-hoc* retrieval by incorporating term specificity in the learned word representations. Finally, an extension of NVSM that integrates text matching and product substitutability for product search is presented in [73].

Differently from representation-based approaches, interaction-based models tackle the problem of predicting the relevance score between a query and a document computing the interactions between the query and document terms. DRMM is one of these approaches and the first that outperforms similar previous existing NLP techniques for text matching such as ARC-II [44], and MatchPyramid [61]. Other successful approaches which belong to this category are *Match-SRNN* [80], *HIerarchical Neural maTching model (HiNT)* [24], *Kernel based Neural Ranking Model (K-NRM)* [83], *Convolutional Kernel-based Neural Ranking Model (Conv-KNRM)* [20], *A Position-Aware Neural IR Model for Relevance Matching (PACRR)* [45] and *A Context-Aware Neural IR Model for Ad-hoc Retrieval (Co-PACRR)* [46]. The architecture of Match-SRNN models the interaction between two texts as a recursive process. This means that the interaction of two texts at each position can be considered as a combination of interactions between their prefixes and words at a given position. This approach is similar to the one employed in MatchPyramid [61], even though it uses a a *Spatial Recurrent Neural Network (SRNN)* instead of a regular *Convolutional Neural Network (CNN)*. HiNT focuses on the diverse relevance patterns in a document given a query. This means that a document may be completely or partially relevant to a query as long as it provides sufficient information for users needs. For this reason, HiNT allows relevance signals at different granularities to compete with each other for final relevance assessment through a hierarchy of matching layers. K-NRM uses a translation matrix that models word-level similarities via word embeddings, a kernel-pooling technique that uses kernels to extract multi-level soft match features, and a learning-to-rank layer that combines those features into the final ranking score. Conv-KNRM is similar to K-NRM but computes the similarity between word n-grams after computing their representation with a CNN on the word embeddings layer. PACRR and Co-PACRR compute the relevance score of a query-document pair from multiple word n-gram similarity

matrices processed first with a CNN and then with a *Recurrent Neural Network (RNN)*. In Co-PACRR, Hui et al. propose to employ – as an extension of the model proposed in PACRR – a context vector to enrich the matching signals, and replace the RNN with a simpler *Feed Forward Neural Network (FFNN)*.

Along with the growing importance of NeuIR models, their reproducibility is becoming a central topic in the IR community. As we mentioned above, Lin criticized the "neural hype" in a recent SIGIR Forum paper [50] and more recently Wui et al. [82] critically examined the baselines used for assessing NeuIR improvements pointing out that "weak baselines (still) pervade the [IR] literature". The comparison to weak baselines is a long-standing problem in the community [7] that needs to be seriously addressed to prevent unreliable claims and statistical analyses. For this reason, in this work we also focused on comparing some NeuIR models with generally good performing "classic" IR models. [82] shows that, currently, only one NeuIR system actually outperforms a well-tuned RM3 model on the Robust04 collection in a re-ranking task; in this paper we further investigate this aspect and, amongst other findings, we extend and confirm Wei et al. findings by showing that classic retrieval models like BM25 and TF-IDF are still highly competitive or better than NeuIR models both for retrieval and re-ranking tasks on a variety of collections.

10. Conclusion

In this work, we analyzed the key components and some relevant issues related to *Neural Information Retrieval (NeuIR)* models. The neural models we selected are prominent examples of the current NeuIR wave and they report competitive results for *ad-hoc* retrieval. *Deep Relevance Matching Model (DRMM)* is a supervised approach which performs a re-ranking of the documents retrieved by another retrieval model. This approach relies on a set of pre-trained word embeddings – obtained with Word2Vec – to extract semantic matching signals which, in turn, are used to perform the re-ranking of documents. The *Neural Vector Space Model (NVSM)*, on the other hand, is an unsupervised model which performs retrieval on the whole collection. NVSM extends two unsupervised representation learning models for expert [75] and product [72] search to *ad-hoc* retrieval. It integrates the notion of term specificity [68, 70] in the learning process of word and document representations. Both models have been tested on shared experimental collections, which play a fundamental role in enabling the reproducibility of the results.

First, we studied the most important factors for the reproducibility of DRMM and NVSM. From the experiments on DRMM, we noticed the importance of sharing the tool used for document pre-processing (i.e. the tool used to compute the input data for the model). This step is often overlooked in the description of the experimental setup of a new NeuIR model. However, it has a great impact on the model performance. An additional factor to consider while using a word embedding model which is not publicly available, is to share the script and tools used to train it. Small changes in the training parameters can lead to very different word embeddings and largely impact the retrieval phase. Regarding NVSM, we have drawn similar conclusions. The lack of information regarding the pre-processing steps – and especially the creation of the word vocabulary – hampers the reproduction of the results presented in [74].

Secondly, we compared DRMM and NVSM to several lexical retrieval models (i.e. BM25, TF-IDF, DFR, and QLM (dir)), semantic approaches (i.e. Word2Vec (add) and Word2Vec (si) [79]) and other state-of-the-art approaches (i.e. BM25+RM3 [84], as well as an application of BERT to IR [85]). From the comparison analysis, we observed that DRMM outperforms traditional lexical models on the Robust04 collection. Therefore, its combination with state-of-the-art retrieval methods, such as BM25+RM3 [84] or BERT-based applications for IR like [85], could further improve the retrieval effectiveness. On the other hand, the limited number of topics available on the NY collection combined with its large size and the differences between the documents hampers the learning process of DRMM, leading to poor retrieval performance which worsen the QLM (dir) baseline used for re-ranking. Regarding NVSM, the comparison analysis on the Robust04 showed that it outperforms Word2Vec (add) and Word2Vec (si) [79] semantic baselines. Nevertheless, the gap between NVSM and lexical models is still significant. On the NY collection, DRMM, NVSM, and Word2Vec (si) show comparable results – far from those obtained by lexical models. We also evaluated the effectiveness of NVSM when used to perform re-ranking. The results showed that NVSM significantly deteriorates the performance of the QLM (dir) baseline on which it performs re-ranking. Our intuition is that NVSM, by learning exclusively from the document collection and not relying on any

interaction or labeled data, does not exploit the lexical signals provided by the QLM (dir) baseline. In fact, unlike DRMM which considers exact matching signals between query and document terms to minimize a supervised loss function, NVSM performs a semantic matching between the latent representations of words and documents obtained by minimizing an unsupervised loss function. Thus, the unsupervised nature of NVSM suits better for rank fusion techniques (e.g. CombSUM [31]), where lexical and semantic signals are combined with promising results (see Table 7), or for providing additional features to supervised re-ranking models like DRMM (see Table 12) or Learning-to-Rank.

Thirdly, we evaluated the robustness of the NeuIR models on collections from different domains and in different languages. We considered three CLEF collections, i.e. CLEF-DE, CLEF-FA and CLEF-IT, respectively in German, Farsi and Italian; one Web collection, i.e. WT2g; and, one collection from the medical domain, i.e. OHSUMED. We performed hyperparameters tuning for both DRMM and NVSM, obtaining significant improvements over the default values on WT2g, for DRMM and NVSM, and on CLEF-DE, for NVSM. Regarding DRMM, we reduced the number of bins in the matching histogram and increased the hidden layer size. For NVSM instead, the performance improvement was obtained by tuning the vocabulary size. Our intuition is that the default vocabulary size of NVSM (i.e. 2^{16} words) is not sufficient to represent CLEF-DE and WT2g collections, which present bigger vocabularies compared to those of the other collections considered (see Table 2). Thus, by increasing NVSM's vocabulary size we managed to obtain higher results – even outperforming DRMM on CLEF-DE. However, improving effectiveness comes at the expense of efficiency since a higher number of vocabulary words means a higher number of word embeddings, which in turn means higher memory requirements. Therefore, it is necessary to find a trade-off between the NVSM's vocabulary size and the collection size. The same considerations also apply to the collections employed in the original NVSM paper [74], where Robust04 and NY present vocabulary sizes similar to those of CLEF-DE and WT2g, respectively (see Table 1). Additionally, we found that NVSM struggles more on heterogeneous collections, like WT2g, than in domain-specific ones, like OHSUMED. The reason might be related to the fact that NVSM derives from two approaches specifically tailored to product [72] and expert [75] search, thus presenting a strong domain-specific nature.

Fourthly, we evaluated the impact of different word embedding models (i.e. Word2Vec, FastText and NVSM embeddings) on DRMM. From this analysis, we conclude that collection-based co-occurrence representations of words such as the ones learned by NVSM lead to a performance improvement over traditional embedding models which are trained to represent semantic relations. Also, we did not detect any relevant performance difference when using different semantic-based embedding models (i.e. Word2Vec and FastText), concluding that DRMM learns to rely more on exact matching signals than semantic ones. We also experimented with different hyperparameter settings during the training of Word2Vec embeddings (i.e. learning rate decay, and number of training epochs). These often overlooked hyperparameters variations had a great influence on the overall performance of the model, showing the impact of small differences in word representations on the overall performance of DRMM.

Fifthly, we conducted an in-depth per topic analysis of the performance of DRMM and NVSM, analyzing the different cases where these models outperform the BM25 lexical model or vice-versa. The experiments have shown that DRMM performs better than BM25 on certain topics. In particular, a few relevant documents where the frequency of query terms is low have ranked higher than BM25. On the other hand, through semantic matching, NVSM can retrieve documents which do not contain any query term. This characteristic is however insufficient to outperform BM25 on average. In other words, when semantic matching is required, or there is the need to focus on a limited portion of the document containing relevant information, NeuIR models tend to outperform lexical ones. However, in the considered collections the number of topics where these characteristics are needed is quite limited. As a consequence, the effect of semantic matching has a minor impact on average performances.

We believe that understanding NeuIR strengths and weaknesses can enhance their integration into full-stack IR systems, which employ a variety of pre- and post-retrieval components such as query expansion and relevance feedback. We think that this work and the insights resulting from it can shed some light on NeuIR models and their potentialities, inspiring others to further investigate their differences with traditional lexical models and how their complementary nature can be leveraged to the best.

Acknowledgments

This work is partially supported by the Computational Data Citation (CDC-STARS) project of the University of Padua and by the ExaMode project, as part of the European Union Horizon 2020 program under Grant Agreement no. 825292. We thank the anonymous reviewers for their thorough work and useful suggestions that helped to improve this paper.

References

[1] Agirre, E., Di Nunzio, G. M., Ferro, N., Mandl, T., Peters, C., 2009. CLEF 2008: Ad Hoc Track Overview. In: Peters, C., Deselaers, T., Ferro, N., Gonzalo, J., Jones, G. J. F., Kurimo, M., Mandl, T., Peñas, A. (Eds.), Evaluating Systems for Multilingual and Multimodal Information Access: Ninth Workshop of the Cross–Language Evaluation Forum (CLEF 2008). Revised Selected Papers. Lecture Notes in Computer Science (LNCS) 5706, Springer, Heidelberg, Germany, pp. 15–37.

[2] Agosti, M., Fabris, E., Silvello, G., 2019. On Synergies Between Information Retrieval and Digital Libraries. In: Manghi, P., Candela, L., Silvello, G. (Eds.), Proc of the 15th Italian Research Conference on Digital Libraries, IRCDL 2019. Vol. 988 of Communications in Computer and Information Science. Springer, pp. 3–17.
URL https://doi.org/10.1007/978-3-030-11226-4

[3] Ai, Q., Yang, L., Guo, J., Croft, W. B., 2016. Improving language estimation with the paragraph vector model for ad-hoc retrieval. In: Proc. of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval (SIGIR 2016). ACM Press, New York, USA, pp. 869–872.

[4] Allan, J., Harman, D., Kanoulas, E., Li, D., Van Gysel, C., Voorhees, E., 2017. TREC 2017 common core track overview. In: Proc. of The Twenty-Sixth Text REtrieval Conference, TREC 2017. National Institute of Standards and Technology (NIST), Special Publication 500-324, Washington, USA.

[5] Amati, G., Van Rijsbergen, C. J., 2002. Probabilistic models of information retrieval based on measuring the divergence from randomness. ACM Transactions on Information Systems (TOIS) 20 (4), 357–389.

[6] Arguello, J., Crane, M., Diaz, F., Lin, J., Trotman, A., December 2015. Report on the SIGIR 2015 Workshop on Reproducibility, Inexplicability, and Generalizability of Results (RIGOR). SIGIR Forum 49 (2), 107–116.

[7] Armstrong, T. G., Moffat, A., Webber, W., Zobel, J., 2009. Improvements That Don't Add Up: Ad-Hoc Retrieval Results Since 1998. In: Proc. 18th International Conference on Information and Knowledge Management (CIKM 2009). ACM Press, New York, USA, pp. 601–610.

[8] Bojanowski, P., Grave, E., Joulin, A., Mikolov, T., 2016. Enriching word vectors with subword information. arXiv preprint arXiv:1607.04606.

[9] Borisov, A., Markov, I., de Rijke, M., Serdyukov, P., 2016. A Neural Click Model for Web Search. In: [10], pp. 531–541.
URL https://doi.org/10.1145/2872427.2883033

[10] Bourdeau, J., Hendler, J., Nkambou, R., Horrocks, I., Zhao, B. Y. (Eds.), 2016. Proc. of the 25th International Conference on World Wide Web, WWW 2016. ACM.
URL http://dl.acm.org/citation.cfm?id=2872427

[11] Boytsov, L., Novak, D., Malkov, Y., Nyberg, E., 2016. Off the Beaten Path: Let's Replace Term-Based Retrieval with k-NN Search. In: [58], pp. 1099–1108.
URL https://doi.org/10.1145/2983323.2983815

[12] Burnham, K. P., Anderson, D. R., 2002. Model Selection and Multimodel Inference. A Practical Information-Theoretic Approach, 2nd Edition. Springer-Verlag, Heidelberg, Germany.

[13] Callan, J. P., Croft, W. B., Broglio, J., 1995. TREC and TIPSTER experiments with INQUERY. Information Processing & Management 31 (3), 327–343.

[14] Chen, M., 2017. Efficient vector representation for documents through corruption. arXiv preprint arXiv:1707.02377.

[15] Clancy, R., Ferro, N., Hauff, C., Lin, J., Sakai, T., Wu, Z. Z., 2019. The SIGIR 2019 Open-Source IR Replicability Challenge (OSIRRC 2019). In: Proc. of the 42nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2019). ACM Press, pp. 1432–1434.

[16] Cover, T. M., Thomas, J. A., 2012. Elements of information theory. John Wiley & Sons.

[17] Craswell, N., Croft, W. B., de Rijke, M., Guo, J., Mitra, B., 2017. SIGIR 2017 Workshop on Neural Information Retrieval (Neu-IR'17). In: Proc. 40th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2017). ACM Press, New York, USA, pp. 1431–1432.
URL https://doi.org/10.1145/3077136.3084373

[18] Craswell, N., Croft, W. B., de Rijke, M., Guo, J., Mitra, B., 2018. Neural information retrieval: introduction to the special issue. Inf. Retr. Journal 21 (2-3), 107–110.

[19] Craswell, N., Croft, W. B., Guo, J., Mitra, B., de Rijke, M., 2016. Neu-IR: The SIGIR 2016 Workshop on Neural Information Retrieval. In: Perego, R., Sebastiani, F., Aslam, J. A., Ruthven, I., Zobel, J. (Eds.), Proc. 39th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2016). ACM Press, New York, USA, pp. 1245–1246.

[20] Dai, Z., Xiong, C., Callan, J., Liu, Z., 2018. Convolutional Neural Networks for Soft-Matching N-Grams in Ad-hoc Search. In: Proc. of the Eleventh ACM International Conference on Web Search and Data Mining, WSDM 2018. ACM Press, pp. 126–134.

[21] Devlin, J., Chang, M. ans Lee, K., Toutanova, K., 2018. BERT: pre-training of deep bidirectional transformers for language understanding. CoRR abs/1810.04805.

[22] Di Nunzio, G. M., Ferro, N., Mandl, T., Peters, C., 2007. CLEF 2006: Ad Hoc Track Overview. In: Peters, C., Clough, P., Gey, F. C., Karlgren, J., Magnini, B., Oard, D. W., de Rijke, M., Stempfhuber, M. (Eds.), Evaluation of Multilingual and Multi-modal Information Retrieval : Seventh Workshop of the Cross–Language Evaluation Forum (CLEF 2006). Revised Selected Papers. Lecture Notes in Computer Science (LNCS) 4730, Springer, Heidelberg, Germany, pp. 21–34.

[23] Dür, A., Rauber, A., Filzmoser, P., 2018. Reproducing a Neural Question Answering Architecture Applied to the SQuAD Benchmark Dataset: Challenges and Lessons Learned. In: [62], pp. 102–113.
URL https://doi.org/10.1007/978-3-319-76941-7_8

[24] Fan, T., Guo, J., Lan, Y., Xu, J., Zhai, C., Cheng, X., 2018. Modeling diverse relevance patterns in ad-hoc retrieval. In: The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval. ACM Press, New York, USA, pp. 375–384.

[25] Ferro, N., Fuhr, N., Järvelin, K., Kando, N., Lippold, M., Zobel, J., 2016. Increasing Reproducibility in IR: Findings from the Dagstuhl Seminar on "Reproducibility of Data-Oriented Experiments in e-Science". SIGIR Forum 50 (1), 68–82.
URL https://doi.org/10.1145/2964797.2964808

[26] Ferro, N., Fuhr, N., Rauber, A., 2018. Introduction to the Special Issue on Reproducibility in Information Retrieval: Evaluation Campaigns, Collections, and Analyses. J. Data and Information Quality 10 (3), 9:1–9:4.
URL https://doi.org/10.1145/3268408

[27] Ferro, N., Fuhr, N., Rauber, A., 2018. Introduction to the special issue on reproducibility in information retrieval: Tools and infrastructures. J. Data and Information Quality 10 (4), 14:1–14:4.
URL https://doi.org/10.1145/3268410

[28] Ferro, N., Marchesin, S., Purpura, A., Silvello, G., 2019. A Docker-Based Replicability Study of a Neural Information Retrieval Model. In: Proc. of the Open-Source IR Replicability Challenge co-located with 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, OSIRRC@SIGIR 2019. pp. 37–43.
URL http://ceur-ws.org/Vol-2409/docker05.pdf

[29] Ferro, N., Peters, C., 2009. CLEF 2009 Ad Hoc Track Overview: TEL & Persian Tasks. In: Borri, F., Nardi, A., Peters, C. (Eds.), Working Notes for the CLEF 2009 Workshop. Published Online.

[30] Ferro, N., Silvello, G., 2015. Rank-Biased Precision Reloaded: Reproducibility and Generalization. In: Hanbury, A., Kazai, G., Rauber, A., Fuhr, N. (Eds.), Advances in Information Retrieval. Proc. 37th European Conference on IR Research (ECIR 2015). Lecture Notes in Computer Science (LNCS) 9022, Springer, Heidelberg, Germany, pp. 768–780.

[31] Fox, E. A., Shaw, J., 1993. Combination of Multiple Searches. In: Harman, D. K. (Ed.), The Second Text REtrieval Conference (TREC-2). National Institute of Standards and Technology (NIST), Special Pubblication 500-215, Washington, USA., pp. 243–252.

[32] Fuhr, N., March 2016. The PRIMAD Model of Reproducibility. Dagstuhl Seminar 16111 - Rethinking Experimental Methods in Computing.

[33] Ganguly, D., Roy, D., Mitra, M., Jones, G. J. F., 2015. Word embedding based generalized language model for information retrieval. In: Proc. of the 38th international ACM SIGIR conference on research and development in information retrieval (SIGIR 2015). ACM Press, New York, USA, pp. 795–798.

[34] Glorot, X., Bengio, Y., 2010. Understanding the difficulty of training deep feedforward neural networks. In: Proc. of the 13th international conference on artificial intelligence and statistics (AISTATS). pp. 249–256.

[35] Guo, J., Fan, Y., Ai, Q., Croft, W. B., 2016. A Deep Relevance Matching Model for Ad-hoc Retrieval. In: [58], pp. 55–64.
URL https://doi.org/10.1145/2983323.2983769

[36] Guo, J., Fan, Y., Ai, Q., Croft, W. B., 2016. Semantic matching by non-linear word transportation for information retrieval. In: Proc. of the 25th ACM International on Conference on Information and Knowledge Management (CIKM 2016). ACM Press, New York, USA, pp. 701–710.

[37] Gutmann, M., Hyvärinen, A., 2010. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In: Proc. of the Thirteenth International Conference on Artificial Intelligence and Statistics. pp. 297–304.

[38] Harman, D., 1992. The darpa tipster project. In: ACM SIGIR Forum. Vol. 26. ACM, pp. 26–28.

[39] Harman, D., 1993. Document detection data preparation. In: TIPSTER TEXT PROGRAM: PHASE I: Proceedings of a Workshop held at Fredricksburg, Virginia, September 19-23, 1993.

[40] Hasibi, F., Balog, K., Bratsberg, S. E., 2016. Exploiting Entity Linking in Queries for Entity Retrieval. In: Carterette, B., Fang, H., Lalmas, M., Nie, J. (Eds.), Proc. of the 2016 ACM on International Conference on the Theory of Information Retrieval, ICTIR 2016. ACM Press, New York, USA, pp. 209–218.
URL http://doi.acm.org/10.1145/2970398.2970406

[41] Hawking, D., Voorhees, E. M., Craswell, N., Bailey, P., 1999. Overview of the TREC-8 Web Track. In: Voorhees, E. M., Harman, D. K. (Eds.), The Eighth Text REtrieval Conference (TREC-8). National Institute of Standards and Technology (NIST), Special Publication 500-246, Washington, USA.
URL http://trec.nist.gov/pubs/trec8/papers/web_overview.pdf

[42] Hersh, W. R., Buckley, C., Leone, T. J., Hickam, D., 1994. Ohsumed: an interactive retrieval evaluation and new large test collection for research. In: Croft, W. B., van Rijsbergen, C. J. (Eds.), Proc. 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 1994). Springer, Springer-Verlag, New York, USA, pp. 192–201.

[43] Hopfgartner, F., Hanbury, A., Müller, H., Eggel, I., Balog, K., Brodt, t., Cormack, G. V., Lin, J., Kalpathy-Cramer, J., Kando, N., Kato, M. P., Krithara, A., Gollub, T., Potthast, M., Viegas, E., Mercer, S., 2018. Evaluation-as-a-Service for the Computational Sciences: Overview and Outlook. J. Data and Information Quality 10 (4), 15:1–15:32.

URL https://doi.org/10.1145/3239570
[44] Hu, B., Lu, Z., Li, H., Chen, Q., 2014. Convolutional Neural Network Architectures for Matching Natural Language Sentences. In: Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems (NIPS 2014). pp. 2042–2050.
[45] Hui, K., Yates, A., Berberich, K., de Melo, G., 2017. Pacrr: A position-aware neural ir model for relevance matching. arXiv preprint arXiv:1704.03940.
[46] Hui, K., Yates, A., Berberich, K., de Melo, G., 2018. Co-pacrr: A context-aware neural ir model for ad-hoc retrieval. In: Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining. WSDM '18. ACM, New York, NY, USA, pp. 279–287.
URL http://doi.acm.org/10.1145/3159652.3159689
[47] Krovetz, R., 1993. Viewing Morphology as an Inference Process. In: Korfhage, R., Rasmussen, E., Willett, P. (Eds.), Proc. 16th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 1993). ACM Press, New York, USA, pp. 191–202.
[48] Kullback, S., Leibler, R. A., March 1951. On Information and Sufficiency. The Annals of Mathematical Statistics 22 (1), 79–86.
[49] Le, Q., Mikolov, T., 2014. Distributed representations of sentences and documents. In: International conference on machine learning. pp. 1188–1196.
[50] Lin, J., Jan. 2019. The neural hype and comparisons against weak baselines. SIGIR Forum 52 (2), 40–51.
URL http://doi.acm.org/10.1145/3308774.3308781
[51] Lin, J., Crane, M., Trotman, A., Callan, J., Chattopadhyaya, I., Foley, J., Ingersoll, G., MacDonald, C., Vigna, S., 2016. Toward reproducible baselines: The open-source IR reproducibility challenge. In: Ferro, N., Crestani, F., Moens, M.-F., Mothe, J., Silvestri, F., Di Nunzio, G. M., Hauff, C., Silvello, G. (Eds.), Advances in Information Retrieval - 38th European Conference on IR Research, ECIR 2016, Padua, Italy, March 20-23, 2016. Proceedings. Lecture Notes in Computer Science (LNCS) 9626, Springer, Heidelberg, Germany, pp. 408–420.
URL https://doi.org/10.1007/978-3-319-30671-1_30
[52] MacAvaney, S., Yates, A., Cohan, A., Goharian, N., 2019. Cedr: Contextualized embeddings for document ranking. arXiv preprint arXiv:1904.07094.
[53] Macdonald, C., McCreadie, R., Santos, R. L. T., Ounis, I., 2012. From Puppy to Maturity: Experiences in Developing Terrier. Proc. of OSIR at SIGIR, 60–63.
[54] Mikolov, T., Chen, K., Corrado, G., Dean, J., 2013. Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781.
[55] Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., Dean, J., 2013. Distributed representations of words and phrases and their compositionality. In: Advances in neural information processing systems. pp. 3111–3119.
[56] Mitra, B., Craswell, N., December 2018. An Introduction to Neural Information Retrieval. Foundations and Trends® in Information Retrieval 13 (1), 1–126.
[57] Mitra, B., Diaz, F., Craswell, N., 2017. Learning to Match using Local and Distributed Representations of Text for Web Search. ACM Press, New York, USA, pp. 1291–1299.
URL https://doi.org/10.1145/3038912.3052579
[58] Mukhopadhyay, S., Zhai, C., Bertino, E., Crestani, F., Mostafa, J., Tang, J., Si, L., Zhou, X., Chang, Y., Li, Y., Sondhi, P. (Eds.), 2016. Proc. of the 25th ACM International Conference on Information and Knowledge Management, CIKM 2016. ACM Press, New York, USA.
URL https://doi.org/10.1145/2983323
[59] Onal, K. D., Zhang, Y., Altingovde, I. S., Rahman, M. M., Karagoz, P., Braylan, A., Dang, B., Chang, H., Kim, H., McNamara, Q., Angert, A., Banner, E., Khetan, V., McDonnell, T., Nguyen, A. T., Xu, D., Wallace, B. C., De Rijke, M., Lease, M., 2018. Neural information retrieval: at the end of the early years. Inf. Retr. Journal 21 (2-3), 111–182.
URL https://doi.org/10.1007/s10791-017-9321-y
[60] Palangi, H., Deng, L., Shen, Y., Gao, J., He, X., Chen, J., Song, X., Ward, R., 2014. Semantic modelling with long-short-term memory for information retrieval. arXiv preprint arXiv:1412.6629.
[61] Pang, L., Lan, Y., Guo, J., Xu, J., Wan, S., Cheng, X., 2016. Text matching as image recognition. In: Proc. of the Thirtieth AAAI Conference on Artificial Intelligence. AAAI Press, pp. 2793–2799.
[62] Pasi, G., Piwowarski, B., Azzopardi, L., Hanbury, A. (Eds.), 2018. Proc. of the 40th European Conference on IR Research, ECIR 2018. Vol. 10772 of Lecture Notes in Computer Science. Springer, Heidelberg, Germany.
URL https://doi.org/10.1007/978-3-319-76941-7
[63] Pennington, J., Socher, R., Manning, C., 2014. Glove: Global vectors for word representation. In: Proc. of the 2014 conference on empirical methods in natural language processing (EMNLP). pp. 1532–1543.
[64] Peters, M. E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., Zettlemoyer, L., 2018. Deep contextualized word representations. arXiv preprint arXiv:1802.05365.
[65] Purpura, A., Maggipinto, M., Silvello, G., Susto, G. A., 2019. Probabilistic Word Embeddings in Neural IR: A Promising Model That Does Not Work as Expected (For Now). In: Proc. of the 9th International Conference on the Theory of Information Retrieval (ICTIR). ACM Press, New York, USA.
[66] Qin, T., Liu, T. Y., Xu, J., Li, H., 2010. LETOR: A benchmark collection for research on learning to rank for information retrieval. Inf. Retr. 13, 346–374.
[67] Řehůřek, R., Sojka, P., May 2010. Software Framework for Topic Modelling with Large Corpora. In: Proc. of the LREC 2010 Workshop on New Challenges for NLP Frameworks. ELRA, Valletta, Malta, pp. 45–50, http://is.muni.cz/publication/884893/en.

[68] Robertson, S., 2004. Understanding inverse document frequency: on theoretical arguments for idf. Journal of documentation 60 (5), 503–520.
[69] Silvello, G., Bucco, R., Busato, G., Fornari, G., Langeli, A., Purpura, A., Rocco, G., Tezza, A., Agosti, M., 2018. Statistical Stemmers: A Reproducibility Study. In: [62], pp. 385–397.
URL https://doi.org/10.1007/978-3-319-76941-7_29
[70] Sparck Jones, K., 1972. A statistical interpretation of term specificity and its application in retrieval. Journal of documentation 28 (1), 11–21.
[71] Strohman, T., Metzler, D., Turtle, H., Croft, W. B., 2005. Indri: A language model-based search engine for complex queries. In: Proceedings of the International Conference on Intelligent Analysis. Vol. 2. Citeseer, pp. 2–6.
[72] Van Gysel, C., de Rijke, M., Kanoulas, E., 2016. Learning latent vector spaces for product search. In: [58], pp. 165–174.
URL https://doi.org/10.1145/2983323
[73] Van Gysel, C., de Rijke, M., Kanoulas, E., 2018. Mix'n Match: Integrating Text Matching and Product Substitutability within Product Search. In: Proc. of the 27th ACM International Conference on Information and Knowledge Management (CIKM 2018). ACM Press, New York, USA, pp. 1373–1382.
[74] Van Gysel, C., de Rijke, M., Kanoulas, E., 2018. Neural Vector Spaces for Unsupervised Information Retrieval. ACM Trans. Inf. Syst. 36 (4), 38:1–38:25.
[75] Van Gysel, C., de Rijke, M., Worring, M., 2016. Unsupervised, efficient and semantic expertise retrieval. In: [10], pp. 1069–1079.
URL http://dl.acm.org/citation.cfm?id=2872427
[76] Van Gysel, C., Kanoulas, E., de Rijke, M., 2017. Pyndri: a Python interface to the indri search engine. In: Proc. of the 39th European Conference on IR Research, ECIR 2018. Springer, Springer, Heidelberg, Germany, pp. 744–748.
[77] Van Gysel, C., Li, D., Kanoulas, E., 2018. ILPS at TREC 2017 Common Core Track. arXiv preprint arXiv:1801.10603.
[78] Voorhees, E. M., 2004. Overview of the TREC 2004 Robust Track. In: Voorhees, E. M., Buckland, L. P. (Eds.), The Thirteenth Text REtrieval Conference Proceedings (TREC 2004). National Institute of Standards and Technology (NIST), Special Pubblication 500-261, Whasington, USA. http://trec.nist.gov/pubs/trec13/papers/ROBUST.OVERVIEW.pdf.
[79] Vulić, I., Moens, M. F., 2015. Monolingual and cross-lingual information retrieval models based on (bilingual) word embeddings. In: Baeza-Yates, R., Lalmas, M., Moffat, A., Ribeiro-Neto, B. (Eds.), Proc. 38th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2015). ACM, ACM Press, New York, USA, pp. 363–372.
[80] Wan, S., Lan, Y., Xu, J., Guo, J., Pang, L., Cheng, X., 2016. Match-srnn: Modeling the recursive matching structure with spatial RNN. arXiv preprint arXiv:1604.04378.
[81] Wand, M. P., Jones, M. C., 1995. Kernel Smoothing. Chapman and Hall/CRC, USA.
[82] Wei, J., Kuang, L., Y., P., Lin, J., 2019. Critically examining the" neural hype": Weak baselines and the additivity of effectiveness gains from neural ranking models. In: Proc. of the 42nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2019). ACM Press, pp. 1129–1132.
[83] Xiong, C., Dai, Z., Callan, J., Liu, Z., Power, R., 2017. End-to-end neural ad-hoc ranking with kernel pooling. In: Proceedings of the 40th International ACM SIGIR conference on research and development in information retrieval. ACM Press, New York, USA, pp. 55–64.
[84] Yang, P., Fang, H., Lin, J., 2018. Anserini: Reproducible ranking baselines using lucene. J. Data and Information Quality 10 (4), 16:1–16:20.
URL https://doi.org/10.1145/3239571
[85] Yang, W., Zhang, H., Lin, J., 2019. Simple applications of BERT for ad hoc document retrieval. CoRR abs/1903.10972.
[86] Yang, X., Ounis, I., McCreadie, R., Macdonald, C., Fang, A., 2018. On the Reproducibility and Generalisation of the Linear Transformation of Word Embeddings. In: [62], pp. 263–275.
URL https://doi.org/10.1007/978-3-319-76941-7_20
[87] Yelong, S., Xiaodong, H., Jianfeng, G., Li, D., Mesnil, G., 2014. A latent semantic model with convolutional-pooling structure for information retrieval. In: Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management. ACM Press, New York, USA, pp. 101–110.
[88] Zamani, H., Mitra, B., Song, X., Craswell, N., Tiwary, s., 2018. Neural Ranking Models with Multiple Document Fields. In: Chang, Y., Zhai, C., Liu, Y., Maarek, Y. (Eds.), Proc. of the Eleventh ACM International Conference on Web Search and Data Mining, WSDM 2018. ACM Press, New York, USA, pp. 700–708.
URL https://doi.org/10.1145/3159652.3159730
[89] Zhai, C., Lafferty, J., 2004. A study of smoothing methods for language models applied to information retrieval. ACM Transactions on Information Systems (TOIS) 22 (2), 179–214.
[90] Zuccon, G., Koopman, B., Bruza, P., Azzopardi, L., 2015. Integrating and Evaluating Neural Word Embeddings in Information Retrieval. In: Park, L. A. F., Karimi, S. (Eds.), Proc. of the 20th Australasian Document Computing Symposium, ADCS 2015. ACM Press, New York, USA, pp. 12:1–12:8.
URL https://doi.org/10.1145/2838931.2838936