Names of Editors (Hrsg.): 47. Jahrestagung der Gesellschaft für Informatik e.V. (GI),
Lecture Notes in Informatics (LNI), Gesellschaft für Informatik, Bonn 2017 1

Evaluating the Impact of Word Embeddings on Similarity Scoring in Practical Information Retrieval

Lukas Galke[1,2], Ahmed Saleh[1,2], Ansgar Scherp[1,2]

Abstract: We assess the suitability of word embeddings for practical information retrieval scenarios. Thus, we assume that users issue ad-hoc short queries where we return the first twenty retrieved documents after applying a boolean matching operation between the query and the documents. We compare the performance of several techniques that leverage word embeddings in the retrieval models to compute the similarity between the query and the documents, namely word centroid similarity, paragraph vectors, Word Mover's distance, as well as our novel inverse document frequency (IDF) re-weighted word centroid similarity. We evaluate the performance using the ranking metrics mean average precision, mean reciprocal rank, and normalized discounted cumulative gain. Additionally, we inspect the retrieval models' sensitivity to document length by using either only the title or the full-text of the documents for the retrieval task. We conclude that word centroid similarity is the best competitor to state-of-the-art retrieval models. It can be further improved by re-weighting the word frequencies with IDF before aggregating the respective word vectors of the embedding. The proposed cosine similarity of IDF re-weighted word vectors is competitive to the TF-IDF baseline and even outperforms it in case of the news domain with a relative percentage of 15%.

Keywords: Word embeddings; Document representation; Information retrieval

1 Introduction

Word embeddings have become the default representation for text in many neural network architectures and text processing pipelines [BCV13; Be03; Go16]. In contrast to the typical bag-of-words representations, word embeddings are capable of capturing semantic and syntactic relations between the words [Mi13; PSM14]. So far, they have been successfully employed in various natural language processing tasks such as word analogies, clustering, and classification [BA16; Ku15; Mi13; PSM14]. Word embeddings are recognised as the main reason for natural language processing (NLP) breakout in the last few years [Go16]. Word vectors can be considered a latent semantic representation for heterogeneous textual data.

A word embedding is a distributed vector representation for words [Mi13]. Each word is represented by a low-dimensional (compared to the vocabulary size) dense vector, which

[1] ZBW – Leibniz Information Centre for Economics, Kiel and Hamburg, Germany, {L.Galke, A.Saleh, A.Scherp}@zbw.eu
[2] Knowledge Discovery, Department of Computer Science, Kiel University, Kiel, Germany

is learned from raw text data. In several natural language processing architectures such as neural networks these representations serve as first layer for the conversion from raw tokens (words) to a more useful representation. The property that semantically related terms are clustered close to each other in the representation space proves the usefulness of this approach for classification and other NLP tasks. However, transferring the success of word embeddings to the ad-hoc Information Retrieval (IR) task is currently an active research topic. While embedding-based retrieval models could tackle the vocabulary mismatch problem by making use of the embedding's inherent similarity between distinct words, most of them struggle to compete with the prevalent strong baselines such as TF-IDF [SB88] and BM25 [Ro92].

The majority of practical information retrieval systems rely on an extended boolean model [MRS08; SFW83]. Extended boolean models generalize both standard boolean models and vector space models. These extended boolean models are highly efficient, since the documents can be stored in an inverted index [MRS08]. Thus, the IR system stays responsive even if a huge amount of documents is indexed. Those practical IR systems employ a binary matching operation on the inverted index to reduce the set of documents, to which the similarity of the query is computed (see Figure 1). We consider a practical ad-hoc IR task which is composed of two steps, matching and scoring [MRS08]. In the matching step, documents of the corpus are matched against a query. Typically, this is conducted by (binary) term co-occurrence, i. e., either the document contains at least one term of the query or not (boolean `OR` query). In the scoring step, the matched documents are ranked according to their relevance to the query. As these core IR tasks are different from other NLP tasks, the incorporation of word embeddings is challenging. Since we evaluate the suitability of embedding-based retrieval models in a practical context, we keep the matching operation fixed for all experiments and concentrate on investigating the impact of the similarity scoring operation. Additionally, we restrict ourselves to purely unsupervised models, i. e., we do not employ any relevance information. Please note that every retrieval model can in principle be improved by using query-relevance information. We also do not employ pseudo-relevance feedback since it is typically not applied in a practical IR setting [1].

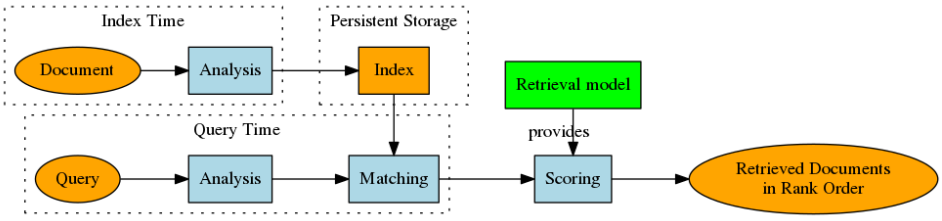

Fig. 1: A simplified information retrieval system.

In this paper, we compare and evaluate several similarity metrics for query-document pairs using word embeddings and assess their suitability in a practical IR setting. The considered

[1] Pseudo-relevance feedback is not included in Apache Lucene, thus SOLR and Elasticsearch

approaches are word centroid similarity, a novel IDF re-weighted variant of word centroid similarity, Word Mover's distance [Ku15], and paragraph vectors [LM14]. Practical IR systems allow treating the fields (`title`, `full-text`, `date`, ...) of a document differently. Thus, we analyze whether the performance of the embedding-based techniques depends on document length. In summary, we will answer the following research questions: *(1) Which embedding-based techniques are suitable for practical information retrieval? (2) How does their performance depend on the document's field, i. e. title vs full-text?*

The remainder is organized as follows: Subsequently, we discuss the related work. We describe our novel IDF re-weighted aggregation of word vectors and the associated word centroid similarity in Section 3. In Section 4, we describe the experiments and report the results in Section 5. We discuss our experimental results in Section 6, before we conclude.

2 Related Work

Extended boolean models such as TF-IDF [SB88] and Okapi BM25 [Ro92] rely on bag-of-words representations, re-weighted by inverse document frequency. While still considered as strong baselines, these models (along with others) struggle to deal with two typical difficulties of the IR task: *term dependencies* and *vocabulary mismatch* [MRS08]. The former means the independence assumption of terms does not hold in natural language, the latter describes the problem of disregarding semantically related terms, when exact matching fails. There are several probabilistic models that rely on language modeling. The documents are ranked either by each document language model's probability of generating the query or by the probability of generating the document, given the query language model [BBL99; Hi98; MLS99; PC98]. The divergence from randomness retrieval model was shown to outperform BM25 consistently on several TREC collections [AR02]. The idea of distributed representations for documents goes back to singular value decomposition of the term-document matrix. It was extended with a probabilistic variant by Hofmann [Ho99]. Finally, Blei et al. [BNJ03] proposed the probabilistic topic model Latent Dirichlet Allocation (LDA) in 2003. Bengio et al. [Be03] first introduced a statistical language model based on neural networks, so-called neural net language models. These language models form the basis for word embeddings learned by a neural network. Mikolov et al. [Mi13] proposed a neural network based word embedding (Word2Vec), in which the representations are learned by training to reconstruct each word's context (skip-gram model). The success of the Word2Vec model relies on skip-gram training with negative sampling, an efficient training algorithm (not involving dense matrix multiplication). Beside other word embeddings [CW08; MH08; TRB10], it is notable that a word embedding can also be computed by directly factorizing the global co-occurrence matrix as done with GloVe [PSM14]. Le and Mikolov [LM14] further extended the Word2Vec approach by adding representations of whole documents by paragraph vectors (Doc2Vec). Their experiments indicate that these distributed representations are useful for information retrieval tasks. However, the evaluation task is to find one relevant document out of three (given 80%

training data), which is not a classical ad-hoc query task. Clinchant and Perronnin [CP13] proposed a method for aggregating word vectors with the Fisher kernel to a document level. The authors applied their approach in ad-hoc retrieval outperforming Latent Semantic Indexing, but not TF-IDF or divergence from randomness. Zheng and Callan [ZC15] learn to re-weight word embeddings using BM25 in a supervised context. Kusner et al. [Ku15] proposed the Word Mover's distance, a similarity metric between documents based on word embeddings. Inspired by the Earth Mover's distance, the Word Mover's distances solves an optimization problem for the minimum cost of transportation between the words of two documents. The cost of moving from a single word to another is the cosine distance of their respective word vectors. Recently, Zamani and Croft [ZC16] proposed embedding based query language models, a dedicated retrieval technique based on word embeddings which thrives to tackle the vocabulary mismatch problem by incorporating word embeddings into query language models. They propose two methods for embedding-based query expansion as well as a method for embedding-based pseudo-relevance feedback.

3 IDF Re-weighted Aggregation of Word Vectors

In the following, we describe how word embeddings can be leveraged for information retrieval. The desired similarity score between the query and the documents can be obtained by aggregating the word vectors to their centroid and computing the cosine distance.

Word centroid similarity (WCS) Given the term occurrence matrix $\boldsymbol{X} \in \mathbb{R}^{n \times k}$ (n documents over a vocabulary of k words) and a word embedding $\boldsymbol{W} \in \mathbb{R}^{k \times h}$ with word vectors of size h. The value X_{ij} is the number of occurrences of word j in document i. The row j of matrix $\boldsymbol{W}$ is the word vector corresponding to column j of $\boldsymbol{X}$. To compute the centroids, we first normalize each row i of $\boldsymbol{X}$ to unit L2-norm (nBOW representation). Then, we obtain the word centroid representation of the documents by matrix multiplication $\boldsymbol{C} = \boldsymbol{X} \cdot \boldsymbol{W}, \boldsymbol{C} \in \mathbb{R}^{n \times h}$. Now, the cosine similarity of the query to the centroids provides a notion of similarity:

$$\mathrm{WCS}(q, i) = \frac{\left(\boldsymbol{q}^T \cdot \boldsymbol{W}\right) \cdot \boldsymbol{C}_i}{\left\|\boldsymbol{q}^T \cdot \boldsymbol{W}\right\| \cdot ||\boldsymbol{C}_i||}$$

The employed norm $||\cdot||$ is the L2-norm. Given a query, the documents are ranked by descending cosine similarity to the query. In case of length-normalized word frequency vectors, the resulting ranking of word centroid similarity is equivalent to the one of word centroid distance mentioned by Kusner et al. [Ku15].

As an example for the desired benefit of employing a word embedding, consider a document containing a high amount of occurrences of the word `automobile` and query consisting of the term `car`. The document would be scored by the TF-IDF retrieval model relatively low since the term `car` does not occur frequently in the document. WCS would score the

document higher because the vector representations for `car` and `automobile` are close to each other in the embedding space (cosine similarity of .58 in the considered Word2Vec model).

IDF re-weighted word centroid similarity (IWCS) In addition, we propose a novel variant of the WCS, where the documents' bags of words are re-weighted by inverse document frequency as in TF-IDF, before the centroids are computed. Consider a bag-of-words representation $\boldsymbol{X}$ of the documents, where X_{ij} corresponds to the number of occurrences of word i in document j. We first re-weight $\boldsymbol{X}$ with respect to inverse document frequency:

$$X'_{ij} = X_{ij} \cdot \text{idf}(j)$$
$$\text{idf}(j) = \log \frac{1+n}{1+\text{df}(D,j)}$$

The document frequency $\text{df}(D, j)$ is the number of documents that contain word j. Then, we again normalize the rows of $\boldsymbol{X}$ to unit L2-norm and compute the centroids: $\boldsymbol{C} = \boldsymbol{X'} \cdot \boldsymbol{W}, \boldsymbol{C} \in \mathbb{R}^{n \times h}$. Finally, we compute the cosine similarity to the query and rank the results in descending order (as in the WCS case).

Re-ranking via Word Mover's distance (IWCS-WMD) The Word Mover's distance [Ku15] (WMD) is a distance metric between two documents. The cumulative cost of moving the words of one document to another document is minimized. The cost function for moving from one word to another is defined as the euclidean distance between the word vectors $c(i, j) = \left\|\boldsymbol{W_i} - \boldsymbol{W_j}\right\|_{\text{L2}}$. For each query, we compute the WMD to all documents and rank the results in ascending order. In addition, we also evaluate a variant which takes the top k documents returned by IWCS and re-ranks them according to Word Mover's distance (IWCS-WMD).

4 Experimental Setup

We desribe the experimental setup and preprocessing, used datasets, and evaluation metrics. The data flow is visualized in Figure 2.

Tasks and Preprocessing Given a collection of documents D, a set of queries Q and relevance scores for each query-document pair $\mathcal{R} : Q \times D \to \mathbb{N}$ (the gold standard), the task is to return a ranked list of k (preferably) relevant documents. We evaluate these results according to $\mathcal{R}$. The values of $\mathcal{R}$ are restricted to binary $\{0, 1\} \subset \mathbb{N}$. Since we are interested in the performance of the retrieval models in a practical setting, we perform a disjunctive

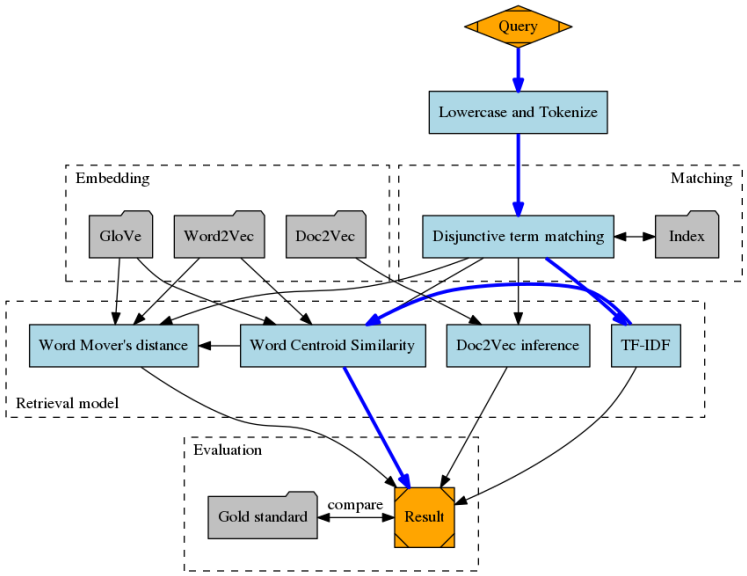

Fig. 2: Data flow graph for a single configuration. Folder shapes indicate persisent data. Rectangular shapes indicate methods and algorithms. Each path from query to result resembles a row in the result tables. As an example, IDF re-weighted Word Centroid Similarity is highlighted in bold (blue edges).

boolean matching operation and do not make any assumptions about the queries when indexing documents. Considering the analysis procedure of documents and queries, we use the same preprocessing steps for all retrieval models: First, we transform the raw string into lower case. Second, we tokenize the string by splitting it into words of at least two word-characters length, while treating any non-word character as delimiter. Finally, we remove common English stop words. To keep complexity under control, we do not apply stemming and only consider uni-gram models. Furthermore, we do **not** remove queries that contain out-of-vocabulary words. In this setting, we compare the performance of the embedding-based retrieval models with respect to the document fields `title`, `abstract`, and `full-text` using either short or long queries. We evaluate the embedding-based retrieval models WCS, IWCS, WMD, IWCS-WMD as described above. In addition, we evaluate paragraph vector (Doc2Vec) inferencing [LM14].

Datasets The *NTCIR2 dataset* [01] consists of 134,978 documents and 49 topics. The documents are composed of a `title` (e. g., "Development and Evaluation of Parallel Computer System Specific for Monte Carlo Device Simulation") and an `abstract` field. The topics consist of the fields `title`, `description` and `narrative`. From these we use the `title` as *short* query (such as "XML") and the `description` as *long* query (such as "Papers about natural language processing using XML"). Additionally, two sets of relevance scores are

provided that associate topics and documents (boolean). From these we chose the second set of relevance scores `rel2` with on average 43.6 (SD: 48.8) relevant documents per query. The relevance scores of the first set are always included in the second set. This results in a higher diversity for the ranking task. The relevancy judgments are not complete, i. e., there are as usual many query-document pairs for which no judgment is given. We assign these documents a relevancy of zero, when evaluating the models.

The *Economics dataset* consists of 61,792 scientific documents in the field of economics. It has 4,518 topics with an average of 72.98 (SD: 329) relevant documents per query. The documents consist of publications such as "The art of compromise" or "Contagious capitalism", As topics, we use the concepts of a thesaurus in economics, the Standard Thesaurus Wirtschaft[2] (STW) of ZBW. A concept in the STW thesaurus consists of one preferre, label and several alternative label. We employ the preferred labels of the concepts as queries (e. g., "sustainable energy supplies"). Each document of the collection is manually annotated (by domain experts) with a set of concepts. Hence, we consider a document being relevant to a topic, if and only if the document is annotated with the corresponding concept.

The *Reuters dataset* consists of 100,000 documents (random sample from Reuters RCV-1 [Le04]) and 102 topics from the news domain. The headlines of the news articles (e. g., "GERMANY: German institute sees slack consumer demand") are considered titles. The documents were manually annotated with one or more of the topics. On average, there are 3,143 (SD: 6,316) relevant documents per topic. Each document consists of a `title` and a `full-text` field. The descriptor label of a topic consists of two to three words (e. g., "energy markets", "crime, law enforcement"). We employ these descriptor labels as query. The assignment of the label to the document resembles relevancy.

Embedding Models Following the results of Mikolov et al. [Mi13] and Kusner et al. [Ku15], employing a well-trained general purpose embedding model is preferable over a corpus-specific model (caused by the surplus in diversity of contexts for each word during training). For this reason, and for the sake of a consistent comparison over the datasets, we employ pre-trained general-purpose word embeddings. Thus, the evaluation is not sensitive to the dataset and its specific training procedure (hyper-parameters are often sensitive to the training corpus). As representative for Word2Vec, we employ the popular `GoogleNews` model (300 dimensions trained on $100 \cdot 10^9$ tokens, vocabulary size of $3 \cdot 10^6$). For GloVe we employ a similar model (300 dimensions trained on $840 \cdot 10^9$ tokens of Common Crawl[3], vocabulary size of $2.2 \cdot 10^6$). As Doc2Vec model, we consult a model trained on Wikipedia articles with a vocabulary size of $3 \cdot 10^6$.

Please note, from pre-experiments we know that ignoring out-of-vocabulary words results in better overall performance than initializing them with random vectors or up-training the missing words (up to 100 epochs of up-training with unmodified original vectors).

[2] `zbw.eu/stw`

[3] A dataset of crawled web data from `https://commoncrawl.org/`

We can furthermore report, that the recently proposed All-but-the-Top embedding post-processing [MBV17] did not improve the retrieval results.

Evaluation Metrics We consider three evaluation metrics: mean average precision (MAP), mean reciprocal rank (MRR), and normalized discounted cumulative gain (NDCG). For all metrics, we limit the considered documents to the top $k = 20$ retrieved documents. This reflects the typical user behavior in a practical web search task. Let D be the set of documents, Q the set of queries, and $\mathcal{R} : Q \times D \to N$ the relevance score of a document for a query. Then a retrieval model can be described as $M : Q \to D^k, q \mapsto y$ with $y \in D^k$ being the top-k retrieved documents in rank order. Thus the multi-set of results for queries Q and a retrieval model M can be written as:

$$R_{M,Q} = \left\{ (\mathcal{R}(q,d))_{d \in M(q)} \mid q \in Q \right\}$$

For a proper definition of the metrics, we operate on the relevance scores $r \in R_{M,Q}$. Precision is defined as the fraction of retrieved documents that are relevant and number of retrieved documents: $\text{Precision}(r,k) = \frac{|\{r_i \in r | r_i > 0\}|}{k}$. The average precision (AP) is computed over the precision values, limited to the top $i = 1, \ldots, k$ retrieved documents: $\text{AP}(r,k) = \frac{1}{|r|} \sum_{i=1}^{k} \text{Precision}((r_1, \ldots, r_i), i)$. The *mean average precision (MAP)* is then aggregated over the query set Q. The reciprocal rank of a query's result is the fraction of the index of the first relevant document $\text{RR}(r) = \frac{1}{\min\{i | r_i > 0\}}$. In case none of the retrieved documents is relevant, the reciprocal rank is set to zero. The respective aggregation over the query set is the *mean reciprocal rank (MRR)*. We compute the *normalized discounted cumulative gain (NDCG)* for a single result list as follows: $\text{DCG}(r,k) = r_1 + \sum_{i=2}^{k} \frac{r_i}{\log_2 i}$, $\text{NDCG}_q(r,k) = \frac{\text{DCG}(r,k)}{\text{IDCG}_{q,\mathcal{R},k}}$ where $\text{IDCG}_{\text{q},\mathcal{R},\text{k}}$ is the best possible (ideal) DCG value for the specific query q with respect to the gold standard $\mathcal{R}$. In case there are more relevant documents than k, the IDCG is also computed on the truncated optimal results. We average NDCG over the queries.

5 Results

Considering the results for the *NTCIR2 dataset*, we inspect four configurations of either the `title` or the `abstract` field and either short or long queries (See Table 1). We observe that using the `title` field leads to better results in all metrics and for all techniques (except for Doc2Vec). In case of short queries, both variants of the Word Mover's distance (WMD, IWCS-WMD) perform consistently worse than IWCS as a query-document similarity (compare .41 to .35 and .38 to .30 MAP). In case of long queries and the `title` field, the IWCS-WMD with the GloVe model attained the highest MAP value .42, .02 higher than the one of the baseline and 0.01 higher than IWCS with the Word2Vec model. Still, in case of the `abstract` field, the MAP value of IWCS with the Word2Vec model (.36) is higher than the WMD re-ranked variants (.30 and .35, respectively). The TF-IDF baseline is outperformed by IWCS in terms of MAP in 3 out of 4 configurations. Still, the margin

is rather small (ranging from .01 to .02). In terms of MRR, the baseline could only be outperformed in one configuration by IWCS with a difference of .01. The NDCG values of the baseline are not reached by any embedding-based retrieval model.

Tab. 1: Results for the NTCIR2 dataset using either the title or the abstract field with respect to the evaluation metrics MAP, MRR, and NDCG, limited to $k = 20$ retrieved documents. The embedding-based retrieval models are used with Word2Vec (W2V) and Glove (GLV).

Field	`title`			`abstract`		
Metric	MAP	MRR	NDCG	MAP	MRR	NDCG
short queries						
TF-IDF	**.46** (.38)	**.55** (.45)	**.19** (.18)	.35 (.37)	.41 (.43)	**.18** (.20)
WCS_{GLV}	.37 (.36)	.42 (.42)	.16 (.18)	.29 (.31)	.40 (.43)	.15 (.17)
WCS_{W2V}	.33 (.34)	.35 (.38)	.14 (.16)	.33 (.35)	.39 (.43)	.13 (.15)
$IWCS_{GLV}$	.41 (.36)	.49 (.44)	.18 (.18)	.32 (.32)	.39 (.41)	.17 (.18)
$IWCS_{W2V}$	.38 (.35)	.45 (.43)	.17 (.18)	**.36** (.34)	**.42** (.41)	.17 (.18)
$IWCS\text{-}WMD_{GLV}$	.35 (.32)	.40 (.38)	.17 (.17)	.35 (.36)	.41 (.42)	.17 (.18)
$IWCS\text{-}WMD_{W2V}$	.30 (.31)	.34 (.37)	.15 (.17)	.29 (.32)	.33 (.39)	.15 (.17)
WMD_{GLV}	.25 (.33)	.27 (.37)	.11 (.17)	.18 (.27)	.21 (.33)	.08 (.14)
WMD_{W2V}	.27 (.35)	.29 (.40)	.11 (.16)	.22 (.29)	.24 (.34)	.10 (.14)
D2V	.27 (.32)	.33 (.39)	.13 (.16)	.29 (.34)	.35 (.42)	.13 (.16)
long queries						
TF-IDF	.40 (.29)	**.51** (.39)	**.20** (.15)	.35 (.32)	**.47** (.43)	**.20** (.21)
WCS_{GLV}	.29 (.29)	.38 (.41)	.15 (.16)	.27 (.26)	.35 (.37)	.14 (.14)
WCS_{W2V}	.30 (.26)	.38 (.38)	.15 (.15)	.30 (.32)	.37 (.41)	.13 (.14)
$IWCS_{GLV}$	.37 (.34)	.45 (.43)	.17 (.16)	.33 (.30)	.44 (.41)	.16 (.16)
$IWCS_{W2V}$	.41 (.35)	.50 (.41)	.19 (.15)	**.36** (.33)	**.47** (.43)	.17 (.16)
$IWCS\text{-}WMD_{GLV}$	**.42** (.36)	.50 (.44)	.17 (.14)	.30 (.30)	.37 (.38)	.17 (.18)
$IWCS\text{-}WMD_{W2V}$	.40 (.31)	**.51** (.41)	.18 (.14)	.35 (.34)	.40 (.41)	.16 (.16)
WMD_{GLV}	.10 (.22)	.12 (.26)	.04 (.08)	.12 (.21)	.14 (.25)	.06 (.10)
WMD_{W2V}	.22 (.33)	.25 (.39)	.08 (.11)	.30 (.32)	.37 (.41)	.13 (.14)
D2V	.24 (.31)	.27 (.37)	.11 (.16)	.16 (.25)	.19 (.31)	.08 (.11)

For the *Economics dataset* (see Table 2), we observe that once again the retrieval over titles yields consistently higher metric values in terms of MAP, MRR, and NDCG. Considering the `title` field, the IWCS is similar to the baseline in terms of MAP (.37). The MRR and NDCG values attained by IWCS are slightly higher than the ones of WCS (.01). In case of `full-text`, no embedding-based technique could outperform the baseline. Doc2Vec inference is the closest competitor with .28 compared to .34 MAP of the baseline. We canceled the experiments with Word Mover's distance related techniques on the full-text after 200 hours. The computational effort disqualifies them for being suitable for full-text retrieval in practice.

Considering the results for the *Reuters dataset* (see Table 3), we observe that IWCS outperforms the baseline in case of the `title` as well as the `full-text` field. The IWCS attains a MAP of .60 compared to .52 of TF-IDF ($\approx$ 15% relative improvement). The results

Tab. 2: Results for the Economics dataset using either the title or the full-text field with respect to the evaluation metrics MAP, MRR, and NDCG, limited to $k = 20$ retrieved documents. Again Word2Vec (W2V) and Glove (GLV) are used as embedding models for the embedding-based similarty metrics.

Field	`title`			`full-text`		
Metric	MAP	MRR	NDCG	MAP	MRR	NDCG
TF-IDF	**.37** (.38)	.42 (.44)	.26 (.30)	**.34** (.35)	**.40** (.43)	**.26** (.30)
WCS_{GLV}	.36 (.37)	.42 (.44)	.25 (.29)	.21 (.29)	.25 (.36)	.13 (.19)
WCS_{W2V}	.36 (.37)	.41 (.43)	.25 (.29)	.26 (.31)	.32 (.40)	.19 (.24)
$IWCS_{GLV}$	**.37** (.37)	**.43** (.43)	.26 (.29)	.23 (.30)	.28 (.37)	.16 (.22)
$IWCS_{W2V}$	**.37** (.37)	**.43** (.43)	**.27** (.30)	.26 (.31)	.32 (.40)	.19 (.24)
$IWCS\text{-}WMD_{GLV}$	.33 (.35)	.38 (.41)	.25 (.28)		did not finish	
$IWCS\text{-}WMD_{W2V}$	.32 (.34)	.36 (.41)	.25 (.28)		did not finish	
WMD_{GLV}	.28 (.34)	.32 (.41)	.19 (.27)		did not finish	
WMD_{W2V}	.27 (.34)	.31 (.41)	.19 (.27)		did not finish	
D2V	.30 (.36)	.35 (.42)	.21 (.28)	.28 (.31)	.33 (.39)	.22 (.26)

for the two embeddings Word2Vec and GloVe are more or less tied in all cases. In case of full-text with the Word2Vec model, re-weighting the top k documents with WMD could slightly improve the MAP (.56 compared to .55), while the NDCG is equal to one of IWCS and the MRR is slightly lower (.58 of TF-IDF compared to .60).

Tab. 3: Results for the Reuters dataset using either the title or the full-text field with respect to the evaluation metrics MAP, MRR, and NDCG, limited to $k = 20$ retrieved documents and use of the two embedding models Word2Vec (W2V) and Glove (GLV).

Field	`title`			`full-text`		
Metric	MAP	MRR	NDCG	MAP	MRR	NDCG
TF-IDF	.52 (.35)	.61 (.43)	.41 (.32)	.51 (.37)	.58 (.43)	.44 (.36)
WCS_{GLV}	.55 (.31)	.63 (.40)	.42 (.29)	.51 (.33)	.60 (.41)	.44 (.33)
WCS_{W2V}	.54 (.33)	.63 (.41)	.43 (.31)	.52 (.35)	.57 (.41)	.46 (.35)
$IWCS_{GLV}$	.58 (.31)	**.69** (.39)	.45 (.29)	.54 (.34)	**.63** (.41)	.47 (.33)
$IWCS_{W2V}$	**.60** (.33)	**.69** (.40)	**.47** (.32)	.55 (.35)	.60 (.41)	**.49** (.36)
$IWCS\text{-}WMD_{GLV}$	.54 (.30)	.62 (.39)	.43 (.49)	.55 (.34)	.61 (.41)	.46 (.33)
$IWCS\text{-}WMD_{W2V}$	.54 (.33)	.58 (.40)	.44 (.32)	**.56** (.37)	.58 (.42)	**.49** (.37)
WMD_{GLV}	.49 (.32)	.54 (.39)	.38 (.29)	.43 (.32)	.50 (.41)	.37 (.31)
WMD_{W2V}	.48 (.34)	.53 (.41)	.39 (.31)	.41 (.34)	.45 (.41)	.33 (.32)
D2V	.48 (.32)	.55 (.41)	.36 (.30)	.43 (.33)	.52 (.43)	.36 (.32)

6 Discussion

Aggregating the MAP values over all datasets, configurations, and embedding models, WCS attained a mean score of .36 (SD: .10), whereas our novel IWCS attains a value of .40 (.11) ($\approx$ 11% relative improvement). The TF-IDF baseline attains a value of .41 (.07). Word

Mover's distance attained a value of .29 (.12), whereas the IWCS-WMD hybrid approach attains a value of .40 (.10). Doc2Vec attains an aggregated MAP score of .31 (.10). We conclude that WCS is the best-performing embedding-based retrieval model and can be extended by a IDF-reweighting (IWCS) to be competitive to the TF-IDF baseline.

For detailed inspection of the difference between the used embedding model, we aggregate the values of IWCS using Word2Vec and IWCS using GloVe. IWCS using Word2Vec attains an aggregated MAP score of .41 (.10), while IWCS using GloVe attains an aggregated MAP score of .39 (.11). Thus, the Word2Vec model is preferable for the investigated datasets. A theoretical benefit of the skip-gram negative sampling algorithm is that it can be used to incrementally learn vectors for out of vocabulary words. Considering the comparison of the document fields `title`, `abstract`, and `full-text`, we also aggregate the respective MAP values. The TF-IDF baseline on the `title` field attains a score of .44 (.06), whereas WCS and IWCS attain aggregated scores of .39 (.09) and .44 (.09), respectively. On the `abstract` and `full-text` fields, the aggregated MAP values are .39 (.07) for TF-IDF, .34 (.11) for WCS and .37 (.11) for IWCS. Thus, the IDF re-weighted aggregation of word vectors can be considered competitive to TF-IDF. The results indicate that embedding-based models especially seem to be advantageous on short texts, such as the `title` field of the documents.

7 Conclusion

We confirm that word embeddings can be successfully employed in a practical information retrieval setting. The proposed cosine similarity of IDF re-weighted, aggregated word vectors is competitive to the TF-IDF baseline. Over all datasets, IWCS improves the performance of WCS by 11%. In case of the news domain, IWCS outperforms the TF-IDF baseline with a with a relative percentage of 15%.

Reproducibility The code for reproducing the experiments is available at `github.com/lgalke/vec4ir`.

Acknowledgment This work was supported by the EU's Horizon 2020 programme under grant agreement H2020-693092 MOVING.

References

[01] Proceedings of the Second Workshop Meeting on Evaluation of Chinese & Japanese Text Retrieval and Text Summarization, NTCIR-2, Tokyo, Japan, March 7-9, 2001, National Institute of Informatics (NII), 2001.

[AR02] Amati, G.; van Rijsbergen, C. J.: Probabilistic models of information retrieval based on measuring the divergence from randomness. ACM Trans. Inf. Syst. 20/4, pp. 357–389, 2002.

[BA16] Balikas, G.; Amini, M.: An empirical study on large scale text classification with skip-gram embeddings. CoRR abs/1606.06623/, 2016.

[BBL99] Beeferman, D.; Berger, A. L.; Lafferty, J. D.: Statistical Models for Text Segmentation. Machine Learning 34/1-3, pp. 177–210, 1999.

[BCV13] Bengio, Y.; Courville, A. C.; Vincent, P.: Representation Learning: A Review and New Perspectives. IEEE Trans. Pattern Anal. Mach. Intell. 35/8, pp. 1798–1828, 2013.

[Be03] Bengio, Y.; Ducharme, R.; Vincent, P.; Janvin, C.: A Neural Probabilistic Language Model. Journal of Machine Learning Research 3/, pp. 1137–1155, 2003.

[BNJ03] Blei, D. M.; Ng, A. Y.; Jordan, M. I.: Latent Dirichlet Allocation. Journal of Machine Learning Research 3/, pp. 993–1022, 2003.

[CP13] Clinchant, S.; Perronnin, F.: Textual Similarity with a Bag-of-Embedded-Words Model. In: ICTIR. ACM, p. 25, 2013.

[CW08] Collobert, R.; Weston, J.: A unified architecture for natural language processing: deep neural networks with multitask learning. In: ICML. Vol. 307. ACM International Conference Proceeding Series, ACM, pp. 160–167, 2008.

[Go16] Goth, G.: Deep or shallow, NLP is breaking out. Commun. ACM 59/3, pp. 13–16, 2016.

[Hi98] Hiemstra, D.: A Linguistically Motivated Probabilistic Model of Information Retrieval. In: ECDL. Vol. 1513. Lecture Notes in Computer Science, Springer, pp. 569–584, 1998.

[Ho99] Hofmann, T.: Probabilistic Latent Semantic Indexing. In: SIGIR. ACM, pp. 50–57, 1999.

[Ku15] Kusner, M. J.; Sun, Y.; Kolkin, N. I.; Weinberger, K. Q.: From Word Embeddings To Document Distances. In: ICML. Vol. 37. JMLR Workshop and Conference Proceedings, JMLR.org, pp. 957–966, 2015.

[Le04] Lewis, D. D.; Yang, Y.; Rose, T. G.; Li, F.: RCV1: A New Benchmark Collection for Text Categorization Research. Journal of Machine Learning Research 5/, pp. 361–397, 2004.

[LM14] Le, Q. V.; Mikolov, T.: Distributed Representations of Sentences and Documents. In: ICML. Vol. 32. JMLR Workshop and Conference Proceedings, JMLR.org, pp. 1188–1196, 2014.

[MBV17] Mu, J.; Bhat, S.; Viswanath, P.: All-but-the-Top: Simple and Effective Postprocessing for Word Representations. CoRR abs/1702.01417/, 2017.

[MH08] Mnih, A.; Hinton, G. E.: A Scalable Hierarchical Distributed Language Model. In: NIPS. Curran Associates, Inc., pp. 1081–1088, 2008.

[Mi13] Mikolov, T.; Sutskever, I.; Chen, K.; Corrado, G. S.; Dean, J.: Distributed Representations of Words and Phrases and their Compositionality. In: NIPS. Pp. 3111–3119, 2013.

[MLS99] Miller, D. R. H.; Leek, T.; Schwartz, R. M.: A Hidden Markov Model Information Retrieval System. In: SIGIR. ACM, pp. 214–221, 1999.

[MRS08] Manning, C. D.; Raghavan, P.; Schütze, H.: Introduction to information retrieval. Cambridge University Press, 2008.

[PC98] Ponte, J. M.; Croft, W. B.: A Language Modeling Approach to Information Retrieval. In: SIGIR. ACM, pp. 275–281, 1998.

[PSM14] Pennington, J.; Socher, R.; Manning, C. D.: Glove: Global Vectors for Word Representation. In: EMNLP. ACL, pp. 1532–1543, 2014.

[Ro92] Robertson, S. E.; Walker, S.; Hancock-Beaulieu, M.; Gull, A.; Lau, M.: Okapi at TREC. Special Publication 500-207/, pp. 21–30, 1992.

[SB88] Salton, G.; Buckley, C.: Term-Weighting Approaches in Automatic Text Retrieval. Inf. Process. Manage. 24/5, pp. 513–523, 1988.

[SFW83] Salton, G.; Fox, E. A.; Wu, H.: Extended Boolean Information Retrieval. Commun. ACM 26/11, pp. 1022–1036, 1983.

[TRB10] Turian, J. P.; Ratinov, L.; Bengio, Y.: Word Representations: A Simple and General Method for Semi-Supervised Learning. In: ACL. The Association for Computer Linguistics, pp. 384–394, 2010.

[ZC15] Zheng, G.; Callan, J.: Learning to Reweight Terms with Distributed Representations. In: SIGIR. ACM, pp. 575–584, 2015.

[ZC16] Zamani, H.; Croft, W. B.: Embedding-based Query Language Models. In: ICTIR. ACM, pp. 147–156, 2016.