# Learning Unsupervised Knowledge-Enhanced Representations to Reduce the Semantic Gap in Information Retrieval

MARISTELLA AGOSTI, STEFANO MARCHESIN, and GIANMARIA SILVELLO,
Department of Information Engineering, University of Padua

The semantic mismatch between query and document terms – i.e., the semantic gap – is a long-standing problem in Information Retrieval (IR). Two main linguistic features related to the semantic gap that can be exploited to improve retrieval are synonymy and polysemy. Recent works integrate knowledge from curated external resources into the learning process of neural language models to reduce the effect of the semantic gap. However, these knowledge-enhanced language models have been used in IR mostly for re-ranking and not directly for document retrieval.

We propose the Semantic-Aware Neural Framework for IR (SAFIR), an unsupervised knowledge-enhanced neural framework explicitly tailored for IR. SAFIR jointly learns word, concept, and document representations from scratch. The learned representations encode both polysemy and synonymy to address the semantic gap. SAFIR can be employed in any domain where external knowledge resources are available. We investigate its application in the medical domain where the semantic gap is prominent and there are many specialized and manually curated knowledge resources. The evaluation on shared test collections for medical literature retrieval shows the effectiveness of SAFIR in terms of retrieving and ranking relevant documents most affected by the semantic gap.

CCS Concepts: • **Information systems → Content analysis and feature selection**; **Retrieval models and ranking**; *Document representation.*

Additional Key Words and Phrases: Knowledge-enhanced retrieval, representation learning, semantic gap, medical literature

**ACM Reference Format:**
Maristella Agosti, Stefano Marchesin, and Gianmaria Silvello. 2020. Learning Unsupervised Knowledge-Enhanced Representations to Reduce the Semantic Gap in Information Retrieval. *ACM Transactions on Information Systems* 1, 1 (July 2020), 48 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

## 1 INTRODUCTION

This paper addresses the semantic gap, a long-standing problem in Information Retrieval (IR). The semantic gap refers to the difference between the machine-level description of document/query contents and the human-level interpretation of their meanings [36, 66]. It can be described also as the mismatch between users' queries and the way retrieval models answer to such queries [83]. We focus on two linguistic features related to the semantic gap: synonymy and polysemy. Synonymy

Authors' address: Maristella Agosti, maristella.agosti@unipd.it; Stefano Marchesin, stefano.marchesin@unipd.it; Gianmaria Silvello, gianmaria.silvello@unipd.it,
Department of Information Engineering, University of Padua, Via Giovanni Gradenigo 6/b, 35131, Padova, Italy.

1046-8188/2020/7-ART $15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

occurs when different words convey the same meaning, whereas polysemy occurs when the same word has different meanings depending on the context.

The semantic gap can affect any search domain, however it is prominent in medical search [18, 35, 36]. Edinger et al. [18] perform a failure analysis on the Text REtrieval Conference (TREC) Medical Records Track [73] to identify what impaired the retrieval systems during the track. One of the outcomes of this failure analysis highlighted that relevant documents were most often infrequently retrieved due to the use of synonyms for topic terms. Koopman and Zuccon [35] investigate if and why assessing relevance of clinical records for a clinical retrieval task is cognitively demanding. The analysis showed, among other things, that the interpretation of a considerable number of queries was subjective and often required careful consideration regarding different possible interpretations. This high degree of subjectivity to interpret queries can increase the mismatch between the machine-level description of document/query contents and their human-level interpretation. Koopman et al. [36] divide the semantic gap into core aspects and analyze them in the medical domain. For each aspect analyzed, the authors provide example queries where the semantic gap is prominent. Similarly, within biomedical literature, the large presence of synonymous and polysemous words – along with the use of acronyms and morphosyntactic variants – poses a critical challenge to retrieval models. For example, an IR model might not effectively answer a query related to the concept of "tumor" if it does not identify the synonymy relationship occurring between "tumor" and, say, "neoplasm". On the other hand, an IR model might retrieve erroneous documents for a query related to the concept of "common cold" if it does not distinguish between the different contextual meanings of the word "cold". In fact, "cold" assumes different meanings depending on the context, including: common cold, cold temperature, and COLD as per Chronic Obstructive Lung Disease. We refer to these queries as "semantically hard" queries.

Two main lines of work have emerged in the past years to bridge the semantic gap between queries and documents: (i) the use of external knowledge resources (e.g., DBpedia[1] or UMLS[2]) to enhance query and document bag-of-words representations [3–5], and (ii) the use of semantic models to perform matching between the latent representations of queries and documents. Semantic models are based on the Distributional Hypothesis [26], which states that words occurring in the same contexts are inclined to convey similar meanings. These models have been revived by the advent of neural language models [6, 39, 45]. Neural language models learn distributed representations of words, also known as word embeddings, based on the context surrounding words. However, their learning process relies exclusively on text corpora and does not consider any external resources, which encode factual knowledge that can help to reduce the semantic gap.

To this end, recent works integrate external knowledge into the learning process of neural language models to reduce the effect of the semantic gap between queries and documents [42, 47, 48]. However, even though knowledge-enhanced language models have been proven effective in many Natural Language Processing (NLP) tasks, their effectiveness is limited in IR [66]. We identify two reasons causing this performance gap. First, knowledge-enhanced language models have been used in IR mostly for re-ranking [42, 48]. In re-ranking, knowledge-enhanced language models are limited to candidate documents retrieved by traditional bag-of-words models, which are not suited to address the semantic gap. Thus, relevant documents most affected by the semantic gap remain undiscovered. Secondly, IR tasks are different from NLP tasks. IR requires to match a given query to a set of relevant documents, whereas NLP mostly deals with the discovery of semantic and linguistic regularities. Therefore, (knowledge-enhanced) neural language models do not encode relevance

[1] https://wiki.dbpedia.org/

[2] https://www.nlm.nih.gov/research/umls/index.html

signals or discriminative aspects between queries and documents – which are fundamental to effectively address IR tasks.

In this work, we consider external knowledge resources as structured and manually curated knowledge bases that store concepts and the relations between them. A concept represents a specific meaning conveyed by a set of related words that are semantic or terminological variants. Depending on the knowledge resource, there can be multiple relations occurring between concepts. Different relations entail different relational constraints. Then, we address the following research questions:

**RQ1** Which feature between synonymy and polysemy can be exploited to reduce the semantic gap and improve retrieval?

**RQ2** How can external knowledge resources help to bridge the semantic gap between queries and documents?

For **RQ1**, we investigate how to leverage synonymy and polysemy in the learning process of neural models. How can we model both features jointly? Which feature is prominent for retrieval effectiveness? To what extent modeling these features is effective for semantically hard queries? For **RQ2**, we explore how integrating external knowledge into neural models impacts retrieval performances. In particular, we seek to understand whether knowledge-enhanced neural models retrieve relevant documents that are most affected by the semantic gap. In other words, to what extent knowledge-enhanced neural models retrieve different relevant documents?

To address our research questions, we propose the Semantic-Aware Neural Framework for IR (SAFIR), an unsupervised knowledge-enhanced neural framework for IR. SAFIR jointly learns word, concept, and document representations from scratch. The learned representations are optimized for IR and encode both polysemy and synonymy to address the semantic gap between queries and documents. SAFIR can be applied to any domain where external knowledge resources are available (e.g., medical, legal, news) and it does not require any labeled data for training – which are scarce and expensive resources.

In this work, we instantiate and evaluate SAFIR for the medical domain, a domain with a high social impact where the semantic gap is prominent and the large presence of specialized knowledge resources, manually curated by professionals, enables us to explore effective ways to integrate external knowledge in IR models to address the semantic gap.

We conduct an experimental evaluation to compare SAFIR with other knowledge-enhanced neural models on a specific task of Clinical Decision Support (CDS): medical literature retrieval. We adopt the Unified Medical Language System (UMLS) Metathesaurus [7] as external knowledge resource and we evaluate models on the TREC CDS collections [53–55] and on the OHSUMED [27] collection.

We consider two retrieval strategies to investigate our research questions: document retrieval and query expansion. Document retrieval gives us the opportunity to investigate the effectiveness of integrating external knowledge into neural models for the typical retrieval scenario, where systems retrieve a set of candidate documents given a query. Query expansion allows us to investigate the effectiveness of knowledge-enhanced neural models – which are specifically designed to address the semantic gap – in retrieving feedback documents for Pseudo Relevance Feedback (PRF) based methods. In other words, we evaluate if knowledge-enhanced neural models provide expansion terms that are more effective at reducing the semantic gap for bag-of-words models.

The main contributions of this work are:

**C1** We introduce SAFIR, an unsupervised knowledge-enhanced neural framework for IR. To the best of our knowledge, SAFIR is the first unsupervised framework that models synonymy and polysemy to jointly learn word, concept, and document representations specifically for

IR. SAFIR does not require any labeled data for training and can be used in domains where explicit relevance labels are scarce and expensive resources.

- **C2** We show how SAFIR integrates synonymy and polysemy for IR tasks. Furthermore, we perform extensive quantitative and qualitative analyses which provide insights into the individual and joint impact of these features in IR. In particular, we investigate the effectiveness of modeling synonymy and polysemy to answer semantically hard queries.
- **C3** We perform quantitative and qualitative analyses that investigate the ability of knowledge-enhanced neural models to retrieve relevant documents affected by the semantic gap. Furthermore, we evaluate the degree of similarity between SAFIR and the considered baselines to understand to what extent they retrieve different relevant documents.
- **C4** We perform in-depth analyses to evaluate the effectiveness of SAFIR compared to other knowledge-enhanced neural approaches and show its robustness for most collections. The analysis for query expansion highlights that knowledge-enhanced neural models grasp different signals than bag-of-words models and retrieve feedback documents that are more effective in providing expansion terms for PRF based methods.

The source code, evaluation results, and statistical analyses developed for this work are publicly available to ease reproducibility.[3]

The rest of the paper is organized as follows: Section 2 reports related work. Section 3 presents the SAFIR framework and the model we developed using it. Section 4 describes the experimental setup. Sections 5 and 6 present the experimental results and provide in-depth quantitative and qualitative analyses for document retrieval and query expansion, respectively. Finally, Section 7 concludes the paper and outlines some possible future work.

## 2 RELATED WORK

Since we focus on neural-based retrieval models addressing the semantic gap without the use of explicit relevance labels, we give an overview of unsupervised neural representation models. We divide these models into two main categories: *corpus-driven*, where the representations are learned relying solely on the corpus, and *knowledge-enhanced*, where the representations are learned relying on the corpus and an external knowledge resource.

### 2.1 Corpus-Driven Representation Models

Since the introduction of probabilistic neural language models [6], building low-dimensional representations of words from large corpora has gained increasing attention in the NLP community. The word2vec models proposed by Mikolov et al. [45] are based on the Distributional Hypothesis [26]. They use the local co-occurrences of words to learn embedded representations of words. In particular, the Continuous Bag-Of-Words (CBOW) architecture predicts a target word by maximizing the log-likelihood of its context words within a fixed-size window; whereas the skip-gram architecture predicts the context words within a fixed-size window given the target word. Conversely, the Global Vector (GloVe) model [50] learns embedded representations of words based on their global co-occurrence. More recently, contextual neural language models have been proposed to overcome the lack of contextualization of traditional word embeddings. Contextual language models generate different word representations for the same word given the context in which the word occurs. Context2vec [44] learns a generic context embedding function using a bidirectional Long Short-Term Memory (LSTM) architecture. ELMo [51] introduces deep contextualized word representations that model both complex characteristics of word use (e.g., syntax and semantics) and how these uses vary across linguistic contexts (i.e., polysemy). The word vectors derive from the internal

[3] source code: https://github.com/stefano-marchesin/SAFIR/ — experimental data: https://doi.org/10.5281/zenodo.3908196

states of a deep bidirectional language model pre-trained on a large corpus. Similarly, BERT [16] models complex characteristics relying on self-attention layers from Transformer networks [71]. Despite being very powerful, the contextual language models have complex architectures and high computational costs. Hence, contextual neural language models have been used in IR only to perform supervised re-ranking [14, 49].

Methods that learn distributed representations of sentences, paragraphs, or documents have also been proposed. Kenter et al. [32] propose the Siamese CBOW model, which takes inspiration from the CBOW model to learn a target sentence from its surrounding (context) sentences. Similarly, the Skip-thought model [34] learns sentence representations by predicting context sentences from the target sentence. As an extension to word2vec, Le and Mikolov [39] propose the doc2vec models. Both doc2vec Distributed Bag-Of-Words (DBOW) and Distributed Memory (DM) architectures jointly learn document and word representations within the same vector space. The DBOW architecture mimics the behavior of word2vec skip-gram architecture, whereas the DM architecture mimics the behavior of word2vec CBOW architecture. Chen [10] presents the Document Vector through Corruption (Doc2VecC), an efficient document representation learning framework. Doc2VecC represents each document as a simple average of word embeddings and it ensures that word representations capture the semantic meanings of the documents during learning. The corruption component introduces a data-dependent regularization that favors informative or rare words and forces the embeddings of common and non-discriminative words to be close to zero.

The advances in representation learning have led the IR community to develop unsupervised retrieval models based on low-dimensional representations of words and documents. We divide unsupervised neural IR models into two groups: (A) methods incorporating features from representation learning models, and (B) methods learning representations of words and documents from scratch.

Within (A), Vulić and Moens [74] propose to compose document representations as the weighted sum of their word embeddings. The method uses the self-information [13] value of each word as its weighting operator. The idea is that corpus-based weights, like Inverse Document Frequency (IDF) or self-information, assign more importance to words bearing more information content during the compositional process. Zuccon et al. [84] combine a traditional retrieval model with a translation model that uses word embeddings to estimate probabilities. Similarly, Ganguly et al. [21] present a generalized language model where the mutual independence between a pair of words no longer holds and word embeddings are used to derive the transformation probabilities between words. Guo et al. [24] introduce the Bag-of-Word-Embeddings (BoWE) model. BoWE represents every document as a matrix of its word embeddings and then models the matching between queries and documents as a non-linear word transportation problem. Ai et al. [2] evaluate the effectiveness of doc2vec DBOW for ad hoc retrieval and, inspired by Levy and Goldberg [40], they perform a deeper analysis later in [1]. Regarding embedding-based query expansion methods, Sordoni et al. [61] propose one of the very first approaches. However, the proposed model learns semantic representations to generate high-quality expansion terms in a supervised way. On the other hand, Zamani and Croft [79, 80] use pre-trained word embeddings to perform query expansion and as an embedding-based relevance model to improve retrieval. Along the same lines, Kuzi et al. [37] present a suite of query expansion methods based on word2vec CBOW embeddings. The expansion terms identified by the embedding-based methods are either used to expand the original query or integrated with PRF based methods. Diaz et al. [17] investigate the effectiveness of local embeddings – learned on topically-constrained corpora – compared to global embeddings – learned on large topically-unconstrained corpora – for query expansion.

Most of the methods in (A) proved to be effective due to their combination with traditional retrieval models. All the methods presented are general and can be applied to any model that provides the required representations (i.e., words, concepts, or documents). Therefore, most of these methods can be applied to SAFIR – in particular embedding-based query expansion methods [37, 79, 80]. However, the focus of this work is the integration of knowledge resources in neural-based retrieval models to address the semantic gap. Thus, we investigate the effectiveness of knowledge-enhanced neural models in terms of the relevant documents they retrieve at early stages of the IR pipeline.

Within (B), Van Gysel et al. [70] introduce an end-to-end representation learning model for expert search that outperforms traditional language models. The model employs only textual evidence to learn word representations – thus avoiding explicit feature engineering – to retrieve experts in online document collections. Van Gysel et al. [68] present the Latent Semantic Entities (LSE) model, a vector space model that jointly learns the representation of words, e-commerce products, and the mapping between them without explicit annotations. LSE directly models the discriminative relation between products and a particular word. Then, Van Gysel et al. [69] present the Neural Vector Space Model (NVSM), which learns word and document representations from scratch without considering any external source of information. NVSM extends the LSE model [68] in three ways: increasing regularization, reducing the internal covariate shift, and incorporating term specificity within word representations. The results showed that NVSM significantly outperforms LSE in news retrieval.

The contribution of this work over (B) lies in the integration of external knowledge resources within neural vector space models to bridge the semantic gap between queries and documents. Compared to NVSM [69], SAFIR jointly learns word, concept, and document representations. The learned representations are optimized for IR and encode both polysemy and synonymy features which are crucial to address the semantic gap between queries and documents. Similarly to NVSM, SAFIR does not require any labeled data for training and can be applied to any domain where external knowledge resources are available. Furthermore, SAFIR is general in the sense that it can enhance any neural model that optimizes representations toward text matching.

## 2.2 Knowledge-Enhanced Representation Models

Distributed representations of words capture the latent relations existing between words by relying only on the corpus as a knowledge resource. In the past few years, several approaches that combine corpus-based information with external knowledge resources to enhance word, sentence, or document representations have emerged. These approaches have been mainly developed to address polysemy and synonymy.

Faruqui et al. [19] propose the retrofitted word2vec (rword2vec). rword2vec retrofits word embeddings using the relational information contained within semantic lexicons. The method forces words connected in the lexicon to have similar representations by minimizing both the distance of each word with its connected words in the lexicon and the distance with its pre-trained representation – namely, the distributed representation obtained with word2vec. Similarly, the counter-fitting method [46] refines distributed word representations relying on both synonymy and antonymy constraints. Johansson and Pina [31] propose a retrofitting approach to address polysemy. First, the approach decomposes the vectors of polysemous words into a convex combination of sense vectors; secondly, it keeps sense vectors similar to those of the neighboring senses in the knowledge resource. Rather than integrating relational constraints directly into the learning objective, Glavas and Vulić [22] transform external lexico-semantic relations into training examples which are used to learn an explicit retrofitting model. The model learns a global specialization function that specializes the vectors of words unobserved during training too.

Yu and Dredze [78] propose a representation model that combines the objective function of neural language models with prior knowledge from external resources to learn improved lexical-semantic word representations. The RC-NET [75] framework exploits both relational and categorical knowledge to produce knowledge-enhanced word representations. In particular, relational and categorical knowledge are encoded through different regularization functions and combined with the original objective of the word2vec skip-gram architecture. Yamada et al. [76] learn separate vector spaces for word and concepts and then aligns them through an anchor-context model which exploits anchors, contained within a knowledge resource, and their context words. The learned word and concept representations are used for Entity Linking (EL). Iacobacci et al. [28] propose an approach to improve semantic similarity that shifts from the word-level to the sense-level by leveraging knowledge from an external resource. Similarly, Mancini et al. [43] propose a model that jointly learns word and sense representations. The model exploits corpus-based information and knowledge from external resources to produce a unified vector space of word and sense embeddings. Conversely, Cheng et al. [11] propose a framework to generate context-aware text representations without diving into the sense space. The proposed framework projects both words and concepts into the same vector space and produces contextual word representations preserving the uniqueness among words while reflecting their context-appropriate meanings. Devine et al. [15] propose to measure semantic similarity between medical concepts using a variation of the neural language models that learn on concepts taken from a knowledge resource and extracted from a corpus. Regarding contextual neural language models, ERNIE models [64, 82] extend BERT by incorporating knowledge resources in the learning process. To the best of our knowledge, ERNIE models have not yet been used in IR.

Sinoara et al. [59] propose an approach that relies on Word Sense Disambiguation (WSD) tools and embedded representations of words and word senses to represent documents. The constructed document representations are then used for text classification. Choi et al. [12] propose a model to learn representations for medical concepts and visits. Given the sequential nature that medical visits possess for each patient, the model treats the document context – i.e., the medical visit – as a temporal feature.

All these models do not target IR and cannot be used straightforwardly to perform retrieval. Moreover, being knowledge-enhanced representation models quite recent, there are only a few methods proposed for IR tasks. Liu et al. [42] exploit word relations from a medical knowledge resource to constrain word representations. The underlying idea is that related words within the knowledge resource should have similar representations. The constrained word representations are then used to perform document re-ranking. The results showed that constrained word representations are more effective than corpus-driven word representations when used together with bag-of-words models for re-ranking. Nguyen et al. [47] present two models: the conceptual doc2vec (cdoc2vec) and the retrofitted doc2vec (rdoc2vec). Similar to the model proposed by Devine et al. [15], cdoc2vec learns document representations built upon concepts that have been previously extracted from text. Then, rdoc2vec retrofits document representations by minimizing the distance between doc2vec and cdoc2vec representations. The learned representations are injected in a text-to-text matching process according to a query expansion technique. Nguyen et al. [48] propose a tri-partite neural language model that leverages explicit knowledge to jointly constrain word, concept, and document representations. The authors employ the model in two retrieval strategies: document re-ranking and query expansion. Tamine et al. [66] extend [47, 48] to investigate the combined use of corpus-based information and external knowledge resources in different NLP and IR tasks. The authors compare the impact of the different learning approaches on the quality of the learned representations. They found that rdoc2vec and tri-partite models show the same level of performance in identifying relevance signals for IR tasks.

SAFIR shows similarities with the works of Liu et al. [42] and Tamine et al. [66]. In particular, SAFIR constrains synonym representations similarly to Liu et al. [42] and learns word, concept, and document representations as in Tamine et al. [66]. Nevertheless, SAFIR models polysemy by combining word and concept representations in the learning process. This creates contextual representations that the model of Liu et al. [42] and those of Tamine et al. [66] do not handle. Furthermore, an important difference between the works of Liu et al. [42], Tamine et al. [66], and this work is that we optimize SAFIR for IR. Conversely, the models proposed by Liu et al. [42] and Tamine et al. [66] are extensions of neural language models – which are optimized for NLP tasks. Therefore, (knowledge-enhanced) neural language models do not encode relevance signals or discriminative aspects between queries and documents – which are fundamental to effectively address IR tasks. This difference reflects on the different loss functions used to train SAFIR and the knowledge-enhanced neural language models. To the best of our knowledge, SAFIR is the first unsupervised knowledge-enhanced framework that learns word, concept, and document representations specifically for IR.

## 3 THE SEMANTIC-AWARE NEURAL FRAMEWORK FOR IR

SAFIR jointly learns word, concept, and document representations from scratch and optimizes them for document retrieval. At the same time, SAFIR addresses the semantic gap by modeling polysemy and synonymy. Regarding polysemy, SAFIR contextualizes word meanings by combining word and concept representations in the learning process. Word and concept representations are optimized to minimize the distance between the so combined word meanings and the documents in the vector space. Thus, word meaning representations are created on-the-fly by combining word and concept representations. This compositional process avoids to create a representation for each word meaning, which is an approach prone to data sparsity [11]. On the other hand, SAFIR models synonymy via multi-task learning. Word representations are shared between two learning tasks that are optimized jointly: text matching and word similarity. For the word similarity task, SAFIR minimizes the distance between word representations for words presenting synonymy relations within an external knowledge resource.

### 3.1 Preliminaries

We call $D$ the set of corpus documents and $V$ the set of unique words in the vocabulary. A document is a sequence of words $d = (w_j)_{j=1}^{m}$, where $w_j$ is the word in the $j^{\text{th}}$ position of $d$ and $m = |d|$ is the document length. Similarly, a query is a sequence of words $q = (w_i)_{i=1}^{n}$, where $w_i$ is the word in the $i^{\text{th}}$ position of $q$ and $n = |q|$ is the query length.

A knowledge resource is a graph $\Omega = (\mathcal{C}, \mathcal{E})$, where $\mathcal{C}$ is the set of nodes (i.e., concepts) and $\mathcal{E}$ is the set of edges (i.e., relations between concepts). Given $\Omega$, we derive the meaning of a word $w$ in $d$ by associating $w$ to a concept $c \in \mathcal{C}$ based on the context of $w$. Therefore, we do not consider phrase-concept associations and we refer to words or terms interchangeably.

We define a knowledge-enhanced document $\phi = (\langle w_j, c_j \rangle)_{j=1}^{m} \in \Phi$ to be an ordered sequence of contextualized word-concept pairs where $w_j \in V$, $c_j \in C$, and $\Phi$ is the set of knowledge-enhanced documents. Symmetrically, a knowledge-enhanced query is defined as $\varphi = (\langle w_i, c_i \rangle)_{i=1}^{n}$.

Given $\Omega = (\mathcal{C}, \mathcal{E})$, we define $C \subseteq \mathcal{C}$ as the set of unique concepts associated with the words contained in the corpus and we consider as synonyms all the semantic and terminological variants that express a concept $c \in C$. This means that also acronyms, graphical variants, and morphosyntactic variants are considered synonyms.

### 3.2 Framework

As shown in Figure 1, SAFIR has three main components: semantic indexing, representation learning, and semantic matching.

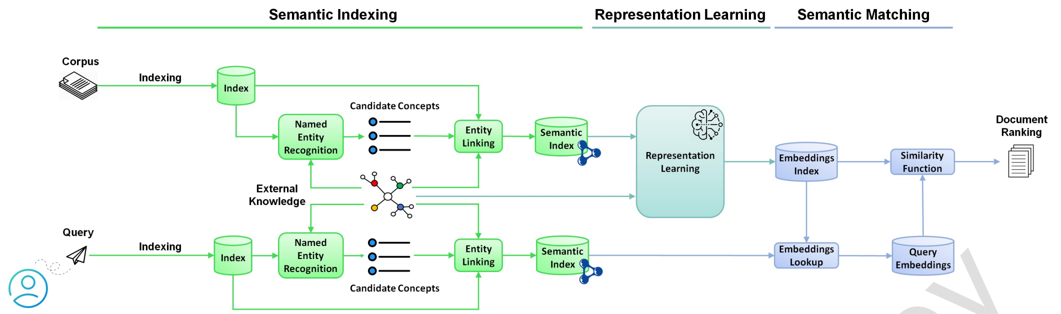


Fig. 1. SAFIR overall architecture. The semantic indexing component produces the knowledge-enhanced documents (and queries) along with the required vocabularies. The representation learning component learns word, concept and document representations. Finally, the semantic matching component computes the similarity score between query and document representations and ranks the documents accordingly.

The **semantic indexing** component takes as input a corpus $D$ and a knowledge resource $\Omega$ and applies Named Entity Recognition (NER) and Entity Linking (EL) techniques to produce the knowledge-enhanced corpus $\Phi$.

The **representation learning** component relies on the output provided by the semantic indexing component to learn word, concept, and document representations. This component models polysemy and synonymy while optimizing representations for document retrieval.

The **semantic matching** component uses the learned representations to perform semantic matching between a knowledge-enhanced query $\varphi$ and the documents $\phi$. Documents are ranked in decreasing order of the similarity score computed between query and document representations.

### 3.3 Semantic Indexing

We adopt the UMLS Metathesaurus as the knowledge resource $\Omega$. We rely on QuickUMLS [60], a fast unsupervised concept extractor built on UMLS to perform NER. We use QuickUMLS to map each word within the word vocabulary $V$ with a list of candidate concepts from UMLS. Given a word, QuickUMLS relies on approximate matching to compute the similarity between the word and the concept labels within UMLS. Concept labels are terms used by the knowledge resource to express a concept. Thus, candidate concepts are ranked according to the similarity score between a target word and the concept labels. Finally, candidate concepts with a similarity score below a given threshold are pruned from the resulting ranking list.[4] Then, we perform EL over candidate concepts returned by QuickUMLS using our modified version of the Shallow Word Sense Disambiguation (S-WSD) algorithm proposed by Mancini et al. [43]. The modified S-WSD takes as input a document $d$, the lists of candidate concepts associated with the words in $d$, and $\Omega$, and it outputs the knowledge-enhanced document $\phi$. S-WSD applies to any $\Omega$ and has running time linear in the collection size $|D|$. Below, we report the details of our modified version of the S-WSD algorithm. Algorithm 1 reports the pseudo-code.

First, we create the set $C_d$ with all the candidate concepts extracted by QuickUMLS for each word $w \in d$ (lines 1 to 3). Secondly, for each candidate concept $\hat{c}$ of $w$, we compute the number of

[4] For more details on QuickUMLS, we refer the reader to its reference paper [60].

**Algorithm 1:** Shallow word-sense disambiguation

```
   Input  : document $d$, candidate concepts from $d$, and knowledge base $\Omega(C, \mathcal{E})$
   Output : knowledge-enhanced document $\phi$
1  Set of candidate concepts $C_d \longleftarrow \emptyset$
2  foreach word $w \in d$ do
3      $C_d \longleftarrow C_d \cup C_w$  ($C_w$ : list of candidate concepts associated to $w$ by QuickUMLS)
4  Output list of word-concept pairs $\phi \longleftarrow [\,]$
5  foreach word $w \in d$ do
6      Relative maximum connections $max = 0$
7      List of senses associated with $w$, $S_w \longleftarrow [\,]$
8      foreach candidate concept $\hat{c} \in C_w$ do
9          Number of edges $n = |\hat{c}' \in C_d : (\hat{c}, \hat{c}') \in \mathcal{E} \;\wedge\; \exists\, w' \in d : w' \neq w \;\wedge\; \hat{c}' \in C_{w'}|$
10         if $n \geq max$ then
11             if $n > max$ then
12                 $S_w \longleftarrow [(w, \hat{c})]$
13                 $max \longleftarrow n$
14             else
15                 $S_w \longleftarrow S_w \cup [(w, \hat{c})]$
16     $(w, c^*) \longleftarrow S_w[0]$ ($S_w[0]$ holds the $\hat{c}$ ranked highest by QuickUMLS among candidates left)
17     $\phi \longleftarrow \phi \cup [(w, c^*)]$
18 return knowledge-enhanced document $\phi$
```

concepts which are connected with $\hat{c}$ in the knowledge base $\Omega$ and are included in $C_d$, excluding connections of concepts which only appear as candidates of the same word (lines 5 to 9). Finally, each word $w$ is associated with its top candidate concept $c^*$ according to its number of connections in the document. If there are ties, the concept with the highest rank from QuickUMLS is associated with the word. The set of top candidate concepts that are returned by the algorithm forms the concept vocabulary $C$ (lines 10 to 18).

The approach we propose to obtain $\Phi$ has two main advantages: (i) it does not require an annotated corpus, which is the biggest bottleneck of supervised EL techniques; (ii) it scales linearly with the corpus size when off-the-shelf disambiguation systems do not [58, 60].

### 3.4 Representation Learning

We develop a shallow neural network learning word, concept, and document representations from scratch. Representations are network parameters in the form of matrices $\{\boldsymbol{w}_i\}_{i=1}^{|V|} \in \mathbb{R}^{|V| \times a}$, $\{\boldsymbol{c}_i\}_{i=1}^{|C|} \in \mathbb{R}^{|C| \times a}$, and $\{\boldsymbol{\phi}_i\}_{i=1}^{|\Phi|} \in \mathbb{R}^{|\Phi| \times b}$ for vocabulary words $V$, vocabulary concepts $C$, and knowledge-enhanced documents $\Phi$, respectively, where $a$ denotes the size of word and concept representations and $b$ the size of document representations. The network models polysemy and synonymy while optimizing the representations for retrieval. For polysemy, word and concept representations are composed to generate contextual word meanings at the representation level. Then, the network optimizes sequences of word meanings to be similar to the knowledge-enhanced documents from which they are extracted. This training process approximates query-documents interactions. At the same time, the network constrains the representations of synonyms to be similar to each other. Therefore, we can divide the network into three main parts: **polysemy modeling**,

**retrieval modeling**, and **synonymy modeling**. Figure 2 depicts the general architecture of the representation learning component.

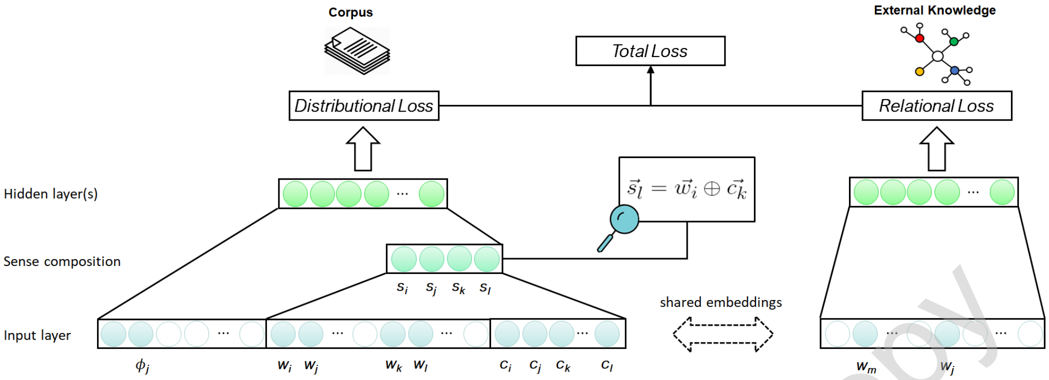


Fig. 2. Neural architecture of the representation learning component. Distributional loss minimizes the distance between document and (contextual) word-concept representations, whereas relational loss minimizes the distance between word representations for words presenting synonymy relations within the knowledge resource.

**Polysemy Modeling**. The network performs a word sense composition process to integrate polysemy. The network models the representation of each word-concept pair $\langle w, c \rangle$ as the element-wise sum of its word and concept representations via a compositional function $f$

$$\boldsymbol{s} = f(\boldsymbol{w}, \boldsymbol{c}) = \boldsymbol{w} \oplus \boldsymbol{c} \tag{1}$$

whose output $\boldsymbol{s}$ is the contextual representation of the word-concept pair $\langle w, c \rangle$. Thus, word meanings are defined in the vector space through a translation process from the word $\boldsymbol{w}$ to its contextual meaning $s$ by the concept $c$, i.e., $\boldsymbol{c}$ acts as a translation vector. In other words, given a word $w$ and, say, two concepts $c_1$ and $c_2$ associated with $w$ in different contexts, the compositional function $f(\cdot,\cdot)$ outputs different representations depending on the concept – and thus the context – considered. Therefore, polysemous words obtain distinct representations according to the context where they appear.

In this way, all the possible combinations of contextual representations are generated on-the-fly offline (training) or online (retrieval). This avoids the need for a word sense vocabulary. Learning representations based on a word sense vocabulary is prone to data sparsity and can lead to underfitting the representations of rare word meanings [11].

Then, the network employs the contextual representations to learn matching relations for retrieval modeling. Matching relations are learned together with synonymy relations via multi-task learning. Word representations are shared between two learning tasks that are optimized jointly: text matching and word similarity.

**Retrieval Modeling**. We adopt neural vector space models [69] for text matching. A neural vector space model takes as input a batch $\mathcal{B}$ of document/sequence pairs and minimizes the distance between their representations. We define a sequence of size $k$ sampled from $\phi$ and starting at position $h$ as $S_h^k(\phi) = (\langle w_j, c_j \rangle)_{j=h}^{h+k-1}$. Then, the representation of the input sequence $S_h^k(\phi)$ is defined as the average of its word-concept pair representations:

$$S^k_h(\phi) = \frac{1}{k}\sum_{i=h}^{h+k-1} f(\boldsymbol{w}_i, \boldsymbol{c}_i) = \frac{1}{k}\sum_{i=h}^{h+k-1} \boldsymbol{s}_i \tag{2}$$

where the word-concept representations $\boldsymbol{s}_i$ are computed as in (1). Then, L2-normalization is applied to the sequence representation $\boldsymbol{S}^k_h(\phi)$, followed by a linear transformation:

$$\boldsymbol{h}^k_h(\phi) = \boldsymbol{W} \cdot \mathrm{norm}(S^k_h(\phi)) \tag{3}$$

where $\boldsymbol{W} \in \mathbb{R}^{b \times a}$ is a projection matrix. The L2 norm$(\cdot)$ function makes the feature values proportionate to each other. Since the objective of the text matching task is to minimize the distance between a document $\phi$ and a sequence $S^k_h(\phi)$ sampled from it, this means that during training the network learns to prioritize some word-concept representations over others when minimizing the distance between a document and a sequence sampled from it. From an IR perspective, the network learns to boost the representation of word-concept pairs that are discriminative for the target document. On the other hand, the linear transformation forces the sequence representation to encode the aspects relevant for text matching into the document space. The network optimizes the projection matrix $\boldsymbol{W}$ to transfer relevant aspects of the sequence representation from the word-concept space $\mathbb{R}^a$ to the document space $\mathbb{R}^b$. Basically, norm$(\cdot)$ boosts the representation of discriminative word-concept pairs and $\boldsymbol{W}$ projects relevant aspects of the sequence representation into the document space.

Before computing the similarity between a sequence $S^k_h(\phi)$ and a document $\phi$, batch normalization [29] is applied to the input sequences, followed by the hard-tanh$(\cdot)$ activation function:

$$\overline{\boldsymbol{h}}^k_h(\phi) = \text{hard-tanh}(\text{batch-norm}(\boldsymbol{h}^k_h(\phi), \mathcal{B})) \tag{4}$$

Batch normalization reduces the internal covariate shift and hard-tanh$(\cdot)$ introduces linear behavior around zero to allow gradients to flow easily when the unit is not saturated, while providing a clear decision in the saturated regime [23].

Thus, the similarity between a document $\phi$ and a sequence $S^k_h(\phi)$ is defined as:

$$P(y|\phi, S^k_h(\phi)) = \sigma(\boldsymbol{\phi} \cdot \overline{\boldsymbol{h}}^k_h(\phi)) \tag{5}$$

where $\overline{\boldsymbol{h}}^k_h(\phi)$ is the standardized representation of the input sequence, $\sigma(\cdot)$ is the sigmoid function, and $y$ is a binary random variable equal to one if $S^k_h(\phi)$ belongs to $\phi$ and zero otherwise.

An adjusted-for-bias variant of the Noise Constrastive Estimation (NCE) loss [25] is used to train the network for the text matching task. NCE maximizes the similarity between the representations of the document $\phi$ and the sequence $S^k_h(\phi)$ sampled from it, while it minimizes the similarity between $S^k_h(\phi)$ and $t$ contrastive documents – i.e., documents not containing the sequence. The re-weighting scheme applied to NCE removes the dependence on the number of contrastive documents $t$, since large values of $t$ bias the network towards contrastive documents. This training procedure mimics query-documents interactions. The log-probability of a document $\phi$ given the sequence $S^k_h(\phi)$ is defined as:

$$\log \overline{P}(\phi|S^k_h(\phi)) = \frac{t+1}{2t}\Big(t \log P(y|\phi, S^k_h(\phi)) + \sum_{\substack{i=1,\\ \phi_i \sim \mathcal{U}(\Phi)}}^{t} \log(1.0 - P(y|\phi_i, S^k_h(\phi)))\Big) \tag{6}$$

where $\mathcal{U}(\Phi)$ represents the uniform distribution over documents $\Phi$ used to obtain the $t$ contrastive examples. Therefore, the loss function used to optimize the network for the text matching task, averaged over the batch size $|\mathcal{B}|$, is:

$$L_{\text{dis}}(\Theta|\mathcal{B}) = -\frac{1}{|\mathcal{B}|}\sum_{i=1}^{|\mathcal{B}|}\log \overline{P}(\phi_i|S_h^k(\phi_i)) \tag{7}$$

where $\Theta$ is the set of parameters $\left\{\{\boldsymbol{w}_i\}_{i=1}^{|V|}, \{\boldsymbol{c}_i\}_{i=1}^{|C|}, \{\boldsymbol{\phi}_i\}_{i=1}^{|\Phi|}, \boldsymbol{W}\right\}$. We refer to this loss as the distributional loss, since it relies on the distributional hypothesis [26].

**Synonymy Modeling**. To integrate synonymy, the network relies on the set of synonym pairs $\mathcal{R} = \{\langle\langle w_i, c_k\rangle, \langle w_j, c_k\rangle\rangle \mid w_i \neq w_j \wedge c_k \in C\}$ of the corpus $\Phi$ and performs word similarity. The objective of the word similarity task is to minimize the distance between two words that are synonyms in $\Omega$. Hence, the network optimizes the representations for words expressing the same concept to be close in the vector space. We define the similarity between two synonyms as:

$$P(y|\langle\langle w_i, c\rangle, \langle w_j, c\rangle\rangle) = \sigma(\boldsymbol{w}_i \cdot \boldsymbol{w}_j) \tag{8}$$

where $y$ is a binary random variable equal to one if both $w_i$ and $w_j$ express $c$ and zero otherwise.

Then, the loss function used to optimize the network for the word similarity task, averaged over the batch size $|\mathcal{B}|$, is:

$$L_{\text{rel}}(\Theta|\mathcal{R}) = -\frac{1}{|\mathcal{B}|}\sum_{\langle\langle w_i, c\rangle, \langle w_j, c\rangle\rangle \in \mathcal{R}} \log P(y|\langle\langle w_i, c\rangle, \langle w_j, c\rangle\rangle) \tag{9}$$

where we recall that $\mathcal{R} = \{\langle\langle w_i, c\rangle, \langle w_j, c\rangle\rangle \mid w_i \neq w_j \wedge c \in C\}$ is the set of synonym pairs of the corpus $\Phi$. We refer to this loss as the relational loss, as it relies on the relational constraints provided by $\Omega$.

The relational loss presents similarities with the constrained embeddings by Liu et al. [42], who propose an online constraining approach that encodes within the objective function of a neural language model the requirement that if a word can be well generated from a given context, its related words should also be well generated from the same context. Compared to that, the relational loss we employ acts as a regularizer which keeps minimizing the distance between words that are synonyms as the training for the text matching task progresses. On the other hand, our approach differs from retrofitting [19] since it is performed during training and not as a second stage of learning. By modeling synonymy as a second-stage regularization, we would end up modifying word representations that have already been optimized towards text matching. In this way, word-concept and document representations could misalign and the network might lose effectiveness on text matching, which is the main task.

Finally, we apply L2 regularization over $\Theta$ parameters:

$$L_{\text{reg}}(\Theta) = \frac{1}{2|\mathcal{B}|}\left(\sum_{i=1}^{|V|}||\boldsymbol{w}_i||_2^2 + \sum_{j=1}^{|C|}||\boldsymbol{c}_j||_2^2 + \sum_{k=1}^{|\Phi|}||\boldsymbol{\phi}_k||_2^2 + ||\boldsymbol{W}||_F^2\right) \tag{10}$$

L2 regularization enforces the network to use all its parameters without depending too heavily on any of them. Therefore, the loss function used to train the entire network is the combination of the L2 regularization and the loss functions for the text matching and word similarity tasks:

$$L(\Theta|\mathcal{B}, \mathcal{R}) = L_{\text{dis}}(\Theta|\mathcal{B}) + \lambda \cdot L_{\text{rel}}(\Theta|\mathcal{R}) + \gamma \cdot L_{\text{reg}}(\Theta) \tag{11}$$

where $\lambda$ controls the extent to which synonym representations are brought close during training and $\gamma$ controls the regularization strength. Parameters $\Theta$ are optimized using Adam [33], an adaptive learning rate optimization function. Adam updates every parameter with every batch. This means that parameters are updated even when they have a zero gradient. Hence, Adam dampens the consequences of applying the hard-tanh activation function, which leads to zero gradients in the saturated regime.

Thus, the network learns contextual representations for word-concept pairs (polysemy) that are close to the corresponding document representations (retrieval) and, at the same time, to synonym representations (synonymy).

### 3.5 Semantic Matching

The learned representations $\{\boldsymbol{w}_i\}_{i=1}^{|V|} \in \mathbb{R}^{|V|\times a}$, $\{\boldsymbol{c}_i\}_{i=1}^{|C|} \in \mathbb{R}^{|C|\times a}$, and $\{\boldsymbol{\phi}_i\}_{i=1}^{|\Phi|} \in \mathbb{R}^{|\Phi|\times b}$ are then used to perform semantic matching between query and document representations. We define the representation of a query $\varphi$ as follows:

$$\boldsymbol{\varphi} = \boldsymbol{W} \cdot \frac{1}{n} \sum_{i=1}^{n} f(\boldsymbol{w}_i, \boldsymbol{c}_i) \tag{12}$$

We treat the query similarly to a sequence by first averaging its word-concept representations and then projecting it into the document space through $\boldsymbol{W}$, which is the projection matrix learned during training. Finally, the matching score between the query $\varphi$ and a document $\phi$ is given by the cosine similarity between their representations $(\boldsymbol{\varphi}, \boldsymbol{\phi})$ in the document space $\mathbb{R}^b$.

## 4 EXPERIMENTAL SETUP

### 4.1 Collections and Knowledge Resource

We consider four standard collections for medical literature retrieval: OHSUMED [27], TREC Clinical Decision Support 2014 [54] (CDS14), 2015 [55] (CDS15), and 2016 [53] (CDS16). Statistics for each collection are reported in Table 1. As knowledge resource, we adopt the 2018AA release of the UMLS Metathesaurus [7].

Table 1. Statistics for the OHSUMED, CDS14, CDS15, and CDS16 collections. Arithmetic mean and standard deviation are reported for document and query lengths.

| | OHSUMED | CDS14 | CDS15 | CDS16 |
|---|---:|---:|---:|---:|
| **Collection** | | | | |
| Document Count | 348,566 | 733,138 | 733,138 | 1,255,259 |
| Vocabulary | 294,520 | 663,528 | 663,528 | 852,739 |
| Document Length | 95.82±62.85 | 117.56±107.16 | 117.56±107.16 | 121.76±142.87 |
| **Queries** | | | | |
| Query Count | 63 | 30 | 30 | 30 |
| Query Length | 7.05±3.00 | 25.63±9.24 | 20.87 ± 6.55 | 32.83 ± 17.29 |

OHSUMED consists of 348,566 references/documents from MEDLINE, the on-line life sciences-biomedicine information database composed of titles and/or abstracts from most of the published medical journals.[5] OHSUMED contains 106 topics, divided into 63 official topics and 43 pre-test topics – that where rejected from official TREC-9 runs for a variety of reasons, but mainly because they had too few relevance judgments. Topics include two fields: *title* (patient description) and *description* (information need). For retrieval, we use the *description* field and we perform experiments on the 63 official topics.

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

CDS14 and CDS15 consist of 733,138 articles from the Open Access Subset of PubMed Central (PMC), an online digital database of freely available full-text biomedical and life sciences literature.[6] CDS14 and CDS15 contain 30 topics, each, representing medical case narratives created by expert topic developers. The case narratives describe information such as a patient's medical history, current symptoms, tests performed by a physician to diagnose the patient's condition, the eventual diagnosis, and any steps taken by a physician to treat the patient. Topics are provided in two variants: a *description*, a complete account of the patients' visits, including details such as their vital statistics, drug dosages, etc.; a *summary*, a simplified version of the narrative that contains less irrelevant information. For retrieval, we use the *summary* variant.

CDS16 consists of 1,255,260 articles from the Open Access Subset of PMC. CDS16 contains 30 topics, representing Electronic Health Records (EHR) admission notes curated by physicians from the MIMIC-III data. Specifically, the notes are extracted from the History of Present Illness (HPI) section of the note. The HPI describes information such as a patient's chief complaint, medical history, tests performed by a physician to diagnose the patient's condition, possibly the current diagnosis, and any steps taken by a physician to treat the patient. Topics are provided in three variants: the EHR admission *note* (only the HPI section); a more layman-friendly *description*, which removes much of the jargon and replaces clinical abbreviations for better readability; a *summary*, a one-or-two sentence summary of the description. For retrieval, we use the *summary* variant.

UMLS Metathesaurus is a large, multi-purpose, and multi-lingual vocabulary database that contains information about biomedical and health related concepts, their name variants, and the relationships among them. The Metathesaurus is built from different thesauri, classifications, code sets, and lists of controlled terms used in patient care, health services, etc. The 2018AA release of the Metathesaurus contains 9,958,614 (English) terms organized by meaning into 3,665,926 concepts and assigned a Concept Unique Identifier (CUI).[7]

## 4.2 Evaluation Measures and Statistical Tests

We use nDCG@1000, nDCG@100, nDCG@10, P@10, and Recall@1000 to evaluate systems. We also consider infNDCG [77] for CDS collections, since it is the reference measure adopted in the TREC CDS tracks. infNDCG cannot be computed for the OHSUMED collection as inferred measures require specific relevance judgments not available for OHSUMED.

We perform the post-hoc Tukey's Honest Significant Differences (HSD) test [67] with one-way ANOVA to test statistical significance. The Tukey's HSD test checks all pairwise differences between runs and, as indicated in [9, 20], it is a viable method for dealing with the multiple comparisons problem [67]. We apply the Tague-Sutcliffe transformation to Tukey's HSD tests [65]. This transformation is applied only if either the Lilliefors or the Jarque-Bera test rejects the normality hypothesis for any experiment.

## 4.3 Retrieval Strategies

We consider two retrieval strategies to investigate our research questions: document retrieval and query expansion.

*Document Retrieval* is the typical retrieval strategy where systems retrieve a set of candidate documents given a query. Documents are ranked according to the similarity score computed between them and the query.

[6] https://www.ncbi.nlm.nih.gov/pmc/

[7] The complete 2018AA statistics can be accessed from: https://www.nlm.nih.gov/research/umls/archive/archive_home.html

*Query Expansion* consists in expanding the original query with additional terms that can help systems to retrieve more relevant documents. Query expansion addresses the semantic gap by using expansion terms to retrieve relevant documents that do not necessarily match the original query. We rely on RM3 [30, 38], an effective PRF based method which typically achieves good retrieval performance at the cost of executing an additional round of retrieval. The set of ranked documents $R_1$ from the first round of retrieval is used to select $m$ expansion terms to augment the query for the second round of retrieval.

### 4.4 Semantic Indexing Setup

We preprocess the document collections using Whoosh,[8] a fast Python search engine library. The preprocessing comprises tokenization and stopwords removal. We rely on the Indri stoplist [63] for stopwords removal. The preprocessed collections are then indexed using Gensim [52]. We index title and abstract fields. This limits noise injection in the training of knowledge-enhanced representation models. Besides, article abstracts from medical literature often present a rich and structured nature that suits to IR tasks and helps us to validate our research questions [8].
For NER and EL, we consider UMLS concepts from the default semantic types provided by QuickUMLS, as they are typically associated with the four aspects of the medical decision criteria: symptoms, diagnostic tests, diagnoses, and treatments. As suggested by Limsopatham et al. [41], these semantic types represent the necessary information health practicioners need to assist their patients. Regarding QuickUMLS, we set the similarity threshold to the default value of 0.7.

Semantic index statistics are presented in Table 2. Table 2(a) shows statistics for the number of candidate concepts per word identified by QuickUMLS (NER), whereas Table 2(b) shows statistics for the number of synonyms per concept after the use of S-WSD (EL). Then, knowledge-enhanced collection statistics are presented in Table 3. Table 3(a-b) show statistics for the number of concepts per document and query, while Table 3(c-d) show statistics for the number of polysemous words per document and query. Finally, statistics for the S-WSD algorithm are reported in Table 4, where statistics are counted over all documents (or queries) and consider the subset of words with at least two candidate concepts associated. In particular, non-disambiguated words refer to those words for which the S-WSD algorithm does not prune the initial list of candidate concepts provided by QuickUMLS.

Table 2. Semantic index statistics for: (a) number of candidate concepts per word, (b) number of synonyms per concept. Statistics are computed for the subset of words/concepts belonging to the term/concept dictionary of SAFIR – therefore they represent a fraction of the collection statistics. (a) considers only words with at least one candidate concept associated (i.e., roughly the 20% of the term dictionary in each collection).

| | (a) concepts/word | | | | (b) synonyms/concept | | | |
|---|---|---|---|---|---|---|---|---|
| | OHSUMED | CDS14 | CDS15 | CDS16 | OHSUMED | CDS14 | CDS15 | CDS16 |
| Max | 67 | 67 | 67 | 67 | 25 | 35 | 35 | 29 |
| Min | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| Median | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| Average | 1.85 | 1.77 | 1.77 | 1.77 | 1.84 | 1.78 | 1.78 | 1.78 |
| Std Dev | 1.96 | 1.94 | 1.94 | 1.87 | 1.63 | 1.64 | 1.64 | 1.62 |

[8] https://whoosh.readthedocs.io/en/latest/intro.html

Table 3. Knowledge-enhanced collection statistics for: (a) number of concepts per document, (b) number of concepts per query, (c) number of polysemous words per document, and (d) number of polysemous words per query.

| | (a) concepts/document | | | | (b) concepts/query | | | |
|---|---|---|---|---|---|---|---|---|
| | OHSUMED | CDS14 | CDS15 | CDS16 | OHSUMED | CDS14 | CDS15 | CDS16 |
| Max | 308 | 14934 | 14934 | 39016 | 9 | 25 | 14 | 38 |
| Min | 0 | 0 | 0 | 0 | 1 | 3 | 3 | 3 |
| Median | 54 | 62 | 62 | 65 | 4 | 10 | 10 | 11 |
| Average | 53.74 | 61.29 | 61.29 | 63.61 | 4.05 | 11.10 | 9.13 | 13.57 |
| Std Dev | 36.01 | 57.74 | 57.74 | 77.01 | 1.63 | 4.81 | 3.19 | 8.31 |
| | (c) polysemy/document | | | | (d) polysemy/query | | | |
| | OHSUMED | CDS14 | CDS15 | CDS16 | OHSUMED | CDS14 | CDS15 | CDS16 |
| Max | 187 | 7164 | 7164 | 18438 | 7 | 16 | 11 | 22 |
| Min | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 2 |
| Median | 29 | 31 | 31 | 32 | 3 | 7 | 6 | 8 |
| Average | 30.98 | 31.75 | 31.75 | 33.13 | 2.78 | 7.20 | 6.17 | 9.07 |
| Std Dev | 21.69 | 30.94 | 30.94 | 40.48 | 1.51 | 3.54 | 2.62 | 5.26 |

Table 4. S-WSD statistics for: % of disambiguated words by S-WSD, % of non-disambiguated words by S-WSD, and execution time of S-WSD. Statistics are counted over all documents/queries and consider the subset of words with at least two candidate concepts associated.

| | % disambiguated words | % non-disambiguated words | Exec. time (sec) |
|---|---|---|---|
| **Collection** | | | |
| OHSUMED | 47.84 | 52.16 | 3,872 |
| CDS14 | 41.49 | 58.51 | 8,360 |
| CDS15 | 41.49 | 58.51 | 8,360 |
| CDS16 | 41.64 | 58.36 | 15,938 |
| **Queries** | | | |
| OHSUMED | 52.00 | 48.00 | – |
| CDS14 | 47.22 | 52.78 | – |
| CDS15 | 52.43 | 47.57 | – |
| CDS16 | 44.49 | 55.51 | – |

### 4.5 Retrieval Models Setup

We consider three categories of retrieval models: bag-of-words models, corpus-driven models, and knowledge-enhanced models. As Bag-of-Words (BoW) models we consider:

(1) BM25 [57] with $k_1 = 1.2$ and $b = 0.75$.
(2) Query Likelihood Model (QLM) [81] with Dirichlet smoothing $\mu = 2000$.

As Corpus-Driven (CD) models we consider only those used by knowledge-enhanced models as part of their learning process, that is:

(3) word2vec [45, 74] with skip-gram architecture, where query and document representations are constructed by summing up the representation of the words contained in them. For document representations, we sum word representations weighted by the term IDF [56] as in [42].

(4) doc2vec [39] with DBOW architecture. The query representation is the sum of its word representations.

(5) Neural Vector Space Model (NVSM) [69]. In NVSM, the query representation is the average of its word representations projected to the document space.

As Knowledge-Enhanced (KE) models we consider:

(6) retrofitted word2vec (rword2vec) [19, 42] with $\alpha_i = 1$ and $\beta_i = \text{degree}(i)^{-1}$, where $i$ is the node the update is applied to. We use rword2vec to retrofit word2vec embeddings.

(7) conceptual doc2vec (cdoc2vec) [47] with DBOW architecture. cdoc2vec is trained over the knowledge-enhanced corpus $\Phi$ relying only on the concept vocabulary $C$. The query representation is the sum of its concept representations.

(8) retrofitted doc2vec (rdoc2vec) [47, 66] retrofits document embeddings from doc2vec and cdoc2vec models. We derive the loss function in [47, 66] to obtain the optimal (closed-form) solution. The weighting factor $\beta$ is optimized in the $(0, 1)$ range with sweep $0.1$. The query representation is the sum of its word representations when $\beta \geq 0.5$ and of its concept representations otherwise.[9]

(9) Semantic-Aware Neural Framework for IR (SAFIR) with word/concept representation size $a = 300$, document representation size $b = 256$, number of contrastive documents $t = 10$, learning rate $\eta = 0.001$, regularization weight $\gamma = 0.001$, synonymy strength $\lambda$ optimized in the $(0, 1]$ range with sweep $0.1$, and batch size $|\mathcal{B}| = 51200$. We consider three variants of SAFIR: SAFIR$_{\text{sp}}$, which integrates both synonymy and polysemy; SAFIR$_{\text{s}}$ which integrates synonymy but not polysemy (i.e., it takes words only as input); and SAFIR$_{\text{p}}$ which integrates polysemy but not synonymy (i.e., it does not consider the word similarity task).

We rely on Elasticsearch to implement BM25 and QLM.[10] For word2vec, doc2vec, and cdoc2vec models we use Gensim, where we disable vocabulary filtering and frequent word sub-sampling to keep the input consistent in all representation models. We set the embedding size to 256, the number of contrastive examples to 10, and the learning rate $\eta = 0.025$ with linear decay $\eta_{\min} = 0.0001$. We set the sequence size of SAFIR and the two-sided window size of neural language models to 16. For NVSM, we disable both the contextual representations (polysemy) and the word similarity task (synonymy) and we set the remaining parameters as for SAFIR. For corpus-driven and knowledge-enhanced models, the word vocabulary size is limited to the the $2^{17}$ most frequent words that have a document frequency greater than 1 and lower than or equal to $\frac{|\Phi|}{2}$. For knowledge-enhanced models, we rely on the 2018AA release of the UMLS Metathesaurus.

SAFIR, NVSM, word2vec, doc2vec, and cdoc2vec are trained for 15 iterations. For each model, we select the iteration that performs best in terms of nDCG@1000, for OHSUMED, and infNDCG for CDS collections. rword2vec and rdoc2vec retrofit optimized word2vec and doc2vec/cdoc2vec, respectively. rword2vec is trained for 10 iterations as the procedure converges to changes lower than $10^{-2}$ in the Euclidean distance [19]. For SAFIR$_{\text{sp}}$ and SAFIR$_{\text{s}}$, we obtain optimal values of the synonymy strength hyperparameter $\lambda$ equal to $0.1$ for both variants in all CDS collections, whereas we obtain values of $\lambda$ equal to $1.0$ and $0.8$ for SAFIR$_{\text{sp}}$ and SAFIR$_{\text{s}}$, respectively, in OHSUMED.

We select the best iteration to evaluate models based on their top performance for the reference measure. On the other hand, the reader can find details on the performances of SAFIR averaged over iterations 10-15 in the electronic appendix, where we compare it with NVSM and BM25/RM3 for document retrieval. We also report the behavior of SAFIR variants in terms of optimization as training progresses. Then, we perform Kendall's $\tau$ correlations between the rankings of the models obtained when we take the best iteration and the average of iterations 10-15. In this way, we can

[9] Information obtained in a personal communication with the authors.

[10] https://www.elastic.co/elasticsearch/

understand to what extent the ranking of the considered models changes when we consider the average of iterations 10-15 instead of the best iteration. The results show that in more than 60% of cases correlation is greater than or equal to 0.80 – which indicates that the differences between rankings do not reflect noticeable changes [72]. The rest of the correlation values divides among 0.60 (13% of cases), 0.40 (17% of cases), and 0.20 (9% of cases). Correlation values of 0.60 occur with two swaps between models in the ranking list, whereas scores of 0.40 and 0.20 with three and four swaps, respectively. Furthermore, low correlations (i.e., 0.40 and 0.20) cluster on Precision-oriented measures, which are highly sensitive to changes across iterations. For these measures, SAFIR$_s$ and SAFIR$_{sp}$ change rank in most collections and BM25/RM3 gains positions. More details can be found in the electronic appendix.

Bag-of-words models are considered to see how they deal with the relevant documents most affected by the semantic gap (**RQ1**). Corpus-driven models are considered as a basis for comparison to evaluate the ability of knowledge-enhanced models to integrate external knowledge in the learning process (**RQ2**). Knowledge-enhanced baselines are compared with SAFIR to investigate both **RQ1** and **RQ2**. Furthermore, the three variants of SAFIR are compared to each other to understand which linguistic feature impacts retrieval the most (**RQ1**) and how knowledge resources are better used to bridge the semantic gap between query and documents (**RQ2**).

### 4.6 Expansion Models Setup

We adopt bag-of-words, corpus-driven, and knowledge-enhanced models to perform the first round of retrieval, whereas we use only bag-of-words models for the second round. We adopt the models optimized for document retrieval and we keep Indri default values for RM3, that is the number of feedback documents $R_1 = 10$, the number of expansion terms $m = 10$, and the interpolation hyperparameter $\alpha = 0.5$.[11]

We consider different categories of retrieval models in the first round of retrieval to evaluate their effectiveness in reducing the semantic gap. Precisely, we investigate whether models that are specifically designed to address the semantic gap retrieve relevant documents that bag-of-words models fail to discover. Our intuition is that semantic models – by retrieving relevant documents different from bag-of-words models – allow RM3 to select expansion terms that are more effective in reducing the semantic gap, thus improving the effectiveness of bag-of-words models in the second round of retrieval. Furthermore, we compare corpus-driven and knowledge-enhanced models to analyze how different linguistic features impact on the choice of expansion terms (**RQ1**) and if knowledge-enhanced models are best suited to this retrieval strategy (**RQ2**).

## 5 DOCUMENT RETRIEVAL: EXPERIMENTAL RESULTS AND DISCUSSION

We present the experimental results for document retrieval and we discuss them based on the research questions. Table 5 shows model performances for document retrieval. In addition to the retrieval models reported above, we also consider BM25/RM3 as a bag-of-words baseline.

### 5.1 The Impact of Polysemy and Synonymy on Document Retrieval

**RQ1** Which feature between synonymy and polysemy can be exploited to reduce the semantic gap and improve retrieval?

We see that all SAFIR variants belong to the top performing group (†) for all measures in all the considered collections. This indicates that SAFIR effectively encodes the text matching signals required to perform retrieval regardless of the linguistic feature(s) modeled. Among the three variants, SAFIR$_p$ provides the best results in CDS collections for most measures. Regarding

[11] https://sourceforge.net/p/lemur/code/HEAD/tree/indri/tags/release-5.16/src/RMExpander.cpp

Table 5. Retrieval performances of considered models. Models are grouped by type: Bag-of-Words (BoW), Corpus-Driven (CD), Knowledge-Enhanced (KE), and SAFIR. In CDS collections, models are optimized by infNDCG, whereas in the OHSUMED collection models are optimized by nDCG@1000. **Bold** values represent the highest scores among the models in each collection. † represents the models belonging to the statistical top group for the given collection with $\alpha \leq 0.05$.

| | | infNDCG | | | | nDCG@1000 | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | CDS14 | CDS15 | CDS16 | OHSUMED | CDS14 | CDS15 | CDS16 | OHSUMED |
| BoW | QLM | 0.1015$^\dagger$ | 0.1277$^\dagger$ | 0.1204$^\dagger$ | – | 0.1750 | 0.1577 | 0.1568$^\dagger$ | 0.5552$^\dagger$ |
| | BM25 | 0.1064$^\dagger$ | 0.1276$^\dagger$ | 0.1399$^\dagger$ | – | 0.1838$^\dagger$ | 0.1579 | 0.1643$^\dagger$ | 0.5875$^\dagger$ |
| | BM25/RM3 | 0.1384$^\dagger$ | **0.1578**$^\dagger$ | **0.1688**$^\dagger$ | – | 0.2316$^\dagger$ | 0.2183$^\dagger$ | **0.2068**$^\dagger$ | **0.6253**$^\dagger$ |
| CD | word2vec | 0.0954$^\dagger$ | 0.1159$^\dagger$ | 0.0928 | – | 0.1548 | 0.1634$^\dagger$ | 0.1054 | 0.5902$^\dagger$ |
| | doc2vec | 0.0242 | 0.0302 | 0.0292 | – | 0.0414 | 0.0453 | 0.0239 | 0.3082 |
| | NVSM | 0.1576$^\dagger$ | 0.1449$^\dagger$ | 0.1475$^\dagger$ | – | 0.2649$^\dagger$ | 0.2213$^\dagger$ | 0.1818$^\dagger$ | 0.5977$^\dagger$ |
| KE | rword2vec | 0.0896$^\dagger$ | 0.1142$^\dagger$ | 0.0790 | – | 0.1501 | 0.1589$^\dagger$ | 0.0980 | 0.5852$^\dagger$ |
| | cdoc2vec | 0.0317 | 0.0517 | 0.0324 | – | 0.0430 | 0.0721 | 0.0335 | 0.2330 |
| | rdoc2vec | 0.0327 | 0.0513 | 0.0292 | – | 0.0429 | 0.0718 | 0.0248 | 0.2067 |
| SAFIR | SAFIR$_s$ | 0.1602$^\dagger$ | 0.1498$^\dagger$ | 0.1546$^\dagger$ | – | 0.2546$^\dagger$ | 0.2240$^\dagger$ | 0.1783$^\dagger$ | 0.6046$^\dagger$ |
| | SAFIR$_p$ | **0.1608**$^\dagger$ | 0.1516$^\dagger$ | 0.1523$^\dagger$ | – | **0.2723**$^\dagger$ | 0.2247$^\dagger$ | 0.1858$^\dagger$ | 0.6106$^\dagger$ |
| | SAFIR$_{sp}$ | 0.1566$^\dagger$ | 0.1515$^\dagger$ | 0.1599$^\dagger$ | – | 0.2651$^\dagger$ | **0.2266**$^\dagger$ | 0.1849$^\dagger$ | 0.6144$^\dagger$ |
| | | nDCG@100 | | | | nDCG@10 | | | |
| | | CDS14 | CDS15 | CDS16 | OHSUMED | CDS14 | CDS15 | CDS16 | OHSUMED |
| BoW | QLM | 0.1035$^\dagger$ | 0.1207$^\dagger$ | 0.1013$^\dagger$ | 0.3974$^\dagger$ | 0.1384$^\dagger$ | 0.2013$^\dagger$ | 0.1150$^\dagger$ | 0.3736$^\dagger$ |
| | BM25 | 0.1098$^\dagger$ | 0.1233$^\dagger$ | 0.1078$^\dagger$ | 0.4392$^\dagger$ | 0.1530$^\dagger$ | **0.2166**$^\dagger$ | **0.1606**$^\dagger$ | 0.4429$^\dagger$ |
| | BM25/RM3 | 0.1338$^\dagger$ | **0.1522**$^\dagger$ | **0.1298**$^\dagger$ | **0.4746**$^\dagger$ | 0.1645$^\dagger$ | 0.1986$^\dagger$ | 0.1518$^\dagger$ | 0.4618$^\dagger$ |
| CD | word2vec | 0.0821 | 0.1064$^\dagger$ | 0.0619 | 0.4461$^\dagger$ | 0.1028 | 0.1435$^\dagger$ | 0.0977$^\dagger$ | **0.4754**$^\dagger$ |
| | doc2vec | 0.0209 | 0.0242 | 0.0196 | 0.1915 | 0.0327 | 0.0211 | 0.0368 | 0.1915 |
| | NVSM | 0.1362$^\dagger$ | 0.1385$^\dagger$ | 0.1077$^\dagger$ | 0.4181$^\dagger$ | 0.1694$^\dagger$ | 0.1664$^\dagger$ | 0.1324$^\dagger$ | 0.3873$^\dagger$ |
| KE | rword2vec | 0.0774 | 0.1032$^\dagger$ | 0.0590 | 0.4421$^\dagger$ | 0.0967$^\dagger$ | 0.1410$^\dagger$ | 0.0930$^\dagger$ | 0.4709$^\dagger$ |
| | cdoc2vec | 0.0215 | 0.0454 | 0.0178 | 0.1355 | 0.0317 | 0.0547 | 0.0225 | 0.1165 |
| | rdoc2vec | 0.0213 | 0.0452 | 0.0202 | 0.1114 | 0.0293 | 0.0588 | 0.0397 | 0.0916 |
| SAFIR | SAFIR$_s$ | 0.1385$^\dagger$ | 0.1411$^\dagger$ | 0.1071$^\dagger$ | 0.4216$^\dagger$ | 0.1729$^\dagger$ | 0.1818$^\dagger$ | 0.1374$^\dagger$ | 0.4121$^\dagger$ |
| | SAFIR$_p$ | **0.1435**$^\dagger$ | 0.1395$^\dagger$ | 0.1113$^\dagger$ | 0.4361$^\dagger$ | **0.1931**$^\dagger$ | 0.2053$^\dagger$ | 0.1519$^\dagger$ | 0.4267$^\dagger$ |
| | SAFIR$_{sp}$ | 0.1401$^\dagger$ | 0.1403$^\dagger$ | 0.1098$^\dagger$ | 0.4397$^\dagger$ | 0.1898$^\dagger$ | 0.1926$^\dagger$ | 0.1475$^\dagger$ | 0.4380$^\dagger$ |
| | | P@10 | | | | Recall@1000 | | | |
| | | CDS14 | CDS15 | CDS16 | OHSUMED | CDS14 | CDS15 | CDS16 | OHSUMED |
| BoW | QLM | 0.1400$^\dagger$ | 0.2233$^\dagger$ | 0.1600$^\dagger$ | 0.4381$^\dagger$ | 0.2375 | 0.1836 | 0.2289$^\dagger$ | 0.7964$^\dagger$ |
| | BM25 | 0.1667$^\dagger$ | 0.2600$^\dagger$ | **0.2167**$^\dagger$ | 0.5016$^\dagger$ | 0.2503$^\dagger$ | 0.1826 | 0.2286$^\dagger$ | 0.7973$^\dagger$ |
| | BM25/RM3 | 0.1833$^\dagger$ | 0.2433$^\dagger$ | 0.2067$^\dagger$ | **0.5413**$^\dagger$ | 0.3151$^\dagger$ | 0.2884$^\dagger$ | **0.3059**$^\dagger$ | 0.8431$^\dagger$ |
| CD | word2vec | 0.1133 | 0.1900$^\dagger$ | 0.1167$^\dagger$ | 0.5048$^\dagger$ | 0.2200 | 0.2194 | 0.1515 | 0.7778 |
| | doc2vec | 0.0367 | 0.0367 | 0.0267 | 0.2190 | 0.0660 | 0.0671 | 0.0305 | 0.4795 |
| | NVSM | 0.2033$^\dagger$ | 0.2333$^\dagger$ | 0.1600$^\dagger$ | 0.4333 | 0.3833$^\dagger$ | 0.3093$^\dagger$ | 0.2617$^\dagger$ | **0.8584**$^\dagger$ |
| KE | rword2vec | 0.1267$^\dagger$ | 0.1967$^\dagger$ | 0.1133 | 0.5048$^\dagger$ | 0.2221 | 0.2151 | 0.1414 | 0.7672 |
| | cdoc2vec | 0.0433 | 0.0933 | 0.0233 | 0.1476 | 0.0658 | 0.1017 | 0.0555 | 0.3889 |
| | rdoc2vec | 0.0367 | 0.1033 | 0.0333 | 0.1175 | 0.0651 | 0.1018 | 0.0313 | 0.3601 |
| SAFIR | SAFIR$_s$ | 0.1967$^\dagger$ | 0.2267$^\dagger$ | 0.1733$^\dagger$ | 0.4619$^\dagger$ | 0.3607$^\dagger$ | **0.3134**$^\dagger$ | 0.2545$^\dagger$ | 0.8582$^\dagger$ |
| | SAFIR$_p$ | **0.2333**$^\dagger$ | **0.2633**$^\dagger$ | 0.1700$^\dagger$ | 0.4762$^\dagger$ | **0.3846**$^\dagger$ | 0.3098$^\dagger$ | 0.2782$^\dagger$ | 0.8548$^\dagger$ |
| | SAFIR$_{sp}$ | 0.2200$^\dagger$ | 0.2467$^\dagger$ | 0.1633$^\dagger$ | 0.4794$^\dagger$ | 0.3733$^\dagger$ | 0.3110$^\dagger$ | 0.2747$^\dagger$ | 0.8520$^\dagger$ |

OHSUMED, SAFIR$_{sp}$ is the top performing variant – closely followed by SAFIR$_{p}$ – for all measures but Recall@1000, where SAFIR$_{s}$ achieves the highest score.

In CDS collections, SAFIR$_{s}$ and SAFIR$_{sp}$ exhibit performances close to or slightly lower than those of NVSM and SAFIR$_{p}$, respectively. We identify two reasons for this. First, NVSM/SAFIR$_{s}$ and SAFIR$_{p}$/SAFIR$_{sp}$ pairs share the same input data, that is words (the former) and word-concept pairs (the latter). Secondly, the optimal values for the hyperparameter $\lambda$ that controls the synonymy strength are equal to 0.1 for both variants in all CDS collections. This suggests that the impact of synonymy in CDS collections might be limited or even detrimental. In particular, we expect that modeling polysemy helps to order relevant documents in top positions of the ranking list, while modeling synonymy helps to retrieve a higher number of relevant documents which contain synonyms of the query terms. While the results confirm this trend for polysemy, they do not for synonymy. In fact, both SAFIR$_{s}$ and SAFIR$_{sp}$ achieve higher results than NVSM and SAFIR$_{p}$, respectively, for Recall@1000 and nDCG@1000 in CDS15 only. The negative results of rword2vec – which models synonymy – compared to those of word2vec further support this intuition. On the other hand, cdoc2vec – which addresses both synonymy and polysemy by learning representations over documents composed only of concepts – achieves better results than doc2vec for most measures. Therefore, the results suggest that polysemy impacts more than synonymy on retrieval performances for CDS collections.

Regarding bag-of-words baselines, all SAFIR variants achieve better performances than QLM and BM25 for most measures in all CDS collections. In particular, for nDCG@1000 and Recall@1000, SAFIR variants statistically outperform QLM in CDS14 and both QLM and BM25 in CDS15.
On the contrary, BM25/RM3, by performing an additional round of retrieval to expand the original query, improves BM25 performances for most measures. Even though the differences between SAFIR variants and BM25/RM3 are not statistically significant, BM25/RM3 achieves performances greater than SAFIR$_{s}$ and SAFIR$_{sp}$ for several cases of the measures considered. Conversely, SAFIR$_{p}$ outperforms BM25/RM3 for the considered measures more than 60% of the time. Interestingly, BM25/RM3 fails to improve BM25 for Precision-oriented measures in CDS15 and CDS16. In both collections, SAFIR$_{p}$ outperforms BM25/RM3 for nDCG@10. This suggests that RM3 might fail to answer semantically hard queries that require to handle polysemy – which reinforces our hypothesis on the impact of polysemy in CDS collections.

For reference, we report the best values obtained during TREC CDS tracks for infNDCG – which is the reference measure adopted in these tracks. In CDS14, the best score for infNDCG is 0.2674 [54]. In CDS15, the best score is 0.2939 [55]. Finally, the best score in CDS16 is 0.2815 [53]. Compared to the results in Table 5, the scores achieved by the best systems submitted to TREC CDS tracks are higher. Note that these systems rely on a variety of different IR components, ranging from pre- and post-retrieval query expansions to re-ranking, and other components, like classifiers. On the other hand, in our work, we exclusively focus on retrieval models and their ability to retrieve relevant documents most affected by the semantic gap. Hence, the integration of SAFIR into multi-stage IR systems and its comparison with state-of-the-art TREC systems is left for future work.

Compared to CDS collections, the results on OHSUMED show a lower gap among the models considered. Bag-of-words models, SAFIR variants, NVSM, and word2vec models behave similarly. The only notable exceptions are for P@10, where NVSM does not belong to the top group (†), and Recall@1000, where word2vec models are statistically outperformed by bag-of-words models, NVSM, and SAFIR variants. Our intuition is that the short, keyword-based nature of OHSUMED queries and the limited corpus vocabulary (see Table 1) impact on the effectiveness of modeling polysemy and synonymy. Besides, short, keyword-based queries favor models relying on corpus-based features. This explains the competitive performances of bag-of-words and word2vec models – which exploit explicit feature engineering by relying on IDF. In particular, word2vec is the top performing system for nDCG@10. Nevertheless, the results of $\text{SAFIR}_{\text{sp}}$ show that polysemy and synonymy can be effectively modeled together. Among the three variants, $\text{SAFIR}_{\text{sp}}$ achieves the best results for nDCG@10, nDCG@100, and nDCG@1000, which indicate its ability to retrieve a higher number of relevant documents (synonymy) and to order them in top positions of the ranking list (polysemy). Furthermore, the differences between NVSM/$\text{SAFIR}_{\text{s}}$ and $\text{SAFIR}_{\text{p}}$/$\text{SAFIR}_{\text{sp}}$ pairs favor SAFIR variants integrating synonymy. Also, the optimal values for the hyperparamter $\lambda$ that controls the synonymy strength are equal to 0.8 and 1.0 for $\text{SAFIR}_{\text{s}}$ and $\text{SAFIR}_{\text{sp}}$, respectively. This indicates the greater impact that modeling synonymy has for OHSUMED rather than CDS.

The performance of BM25/RM3 further confirms the effectiveness of bag-of-words models in OHSUMED. In particular, BM25/RM3 outperforms all the considered models for most measures. The only exceptions are nDCG@10 and Recall@1000, where word2vec and SAFIR/NVSM achieve higher scores, respectively.

To further investigate the impact of polysemy and synonymy in the considered collections, we perform the following quantitative and qualitative analyses.

### 5.1.1 *Polysemy Analysis*

We rely on knowledge-enhanced collection statistics (see Table 3) to identify the degree of polysemy within documents and queries. Then, we perform a qualitative analysis between SAFIR variants to evaluate the impact that the integration of polysemy has in ordering relevant documents in top positions of the ranking list. For each collection, we compute the per-topic differences between SAFIR variants in terms of nDCG@10. We rely on Figures 3–6 to present and discuss the results. The outcomes of the qualitative analysis are used for a second analysis, where we compare SAFIR variants and BM25/RM3 on semantically hard queries. The objective is to understand whether the effectiveness of $\text{SAFIR}_{\text{p}}$ and $\text{SAFIR}_{\text{sp}}$ on highly polysemous queries holds against BM25/RM3. To this end, we compute the per-topic differences between SAFIR variants and BM25/RM3 in terms of nDCG@10. Figures 7–10 complement the analysis.

**The Degree of Polysemy**. When we compare the average number of words per document from Table 1 and the average number of concepts per document from Table 3(a) we observe that the average number of concepts is about half the average number of words in all collections. Furthermore, Table 3(c) shows that, on average, more than half of the words presenting concepts are polysemous. Similar observations can also be made for queries, where the average number of concepts per query fluctuates between one third and a half of the average number of words depending on the collection – as indicated in Table 3(b). Besides, Table 3(d) shows that in all collections more than 60% (on average) of the query words presenting concepts are polysemous. These results indicate the large presence of polysemy within the considered collections.

**The Impact of Modeling Polysemy**. Figures 3–6 point out an interesting behavior of the different SAFIR variants on single queries. Each figure shows the per-topic differences between SAFIR variants at nDCG@10 for a given collection. The green (light) stems indicate that the upper-side SAFIR variant achieves a higher nDCG@10 score than the lower-side variant. Vice versa, red (dark) stems indicate that the lower-side SAFIR variant achieves a higher nDCG@10 score than the upper-side variant. Overall, the nDCG@10 results are not consistently in favor of one or the other SAFIR variant. Depending on the query, a particular SAFIR variant outperforms the other and vice versa. Nevertheless, $\text{SAFIR}_{\text{p}}$ and $\text{SAFIR}_{\text{sp}}$ show results closer to each other than to $\text{SAFIR}_{\text{s}}$ since they both model polysemy.

In OHSUMED (Figure 3), $\text{SAFIR}_{\text{p}}$ and $\text{SAFIR}_{\text{s}}$ achieve higher nDCG@10 scores for thirty-one and twenty-nine queries, respectively, whereas on three queries they perform equally. Overall, $\text{SAFIR}_{\text{p}}$

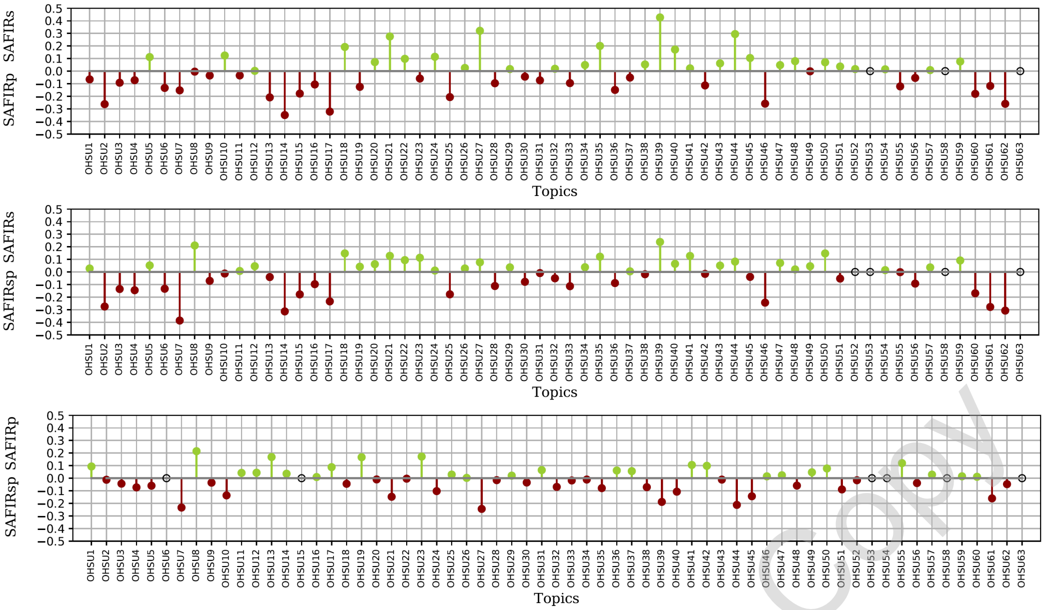


Fig. 3. Per-topic differences between SAFIR variants at nDCG@10 in OHSUMED collection. The green (light) stems indicate that the upper-side SAFIR variant achieves a higher nDCG@10 score than the lower-side variant. Vice versa, red (dark) stems indicate that the lower-side SAFIR variant achieves a higher nDCG@10 score than the upper-side variant.

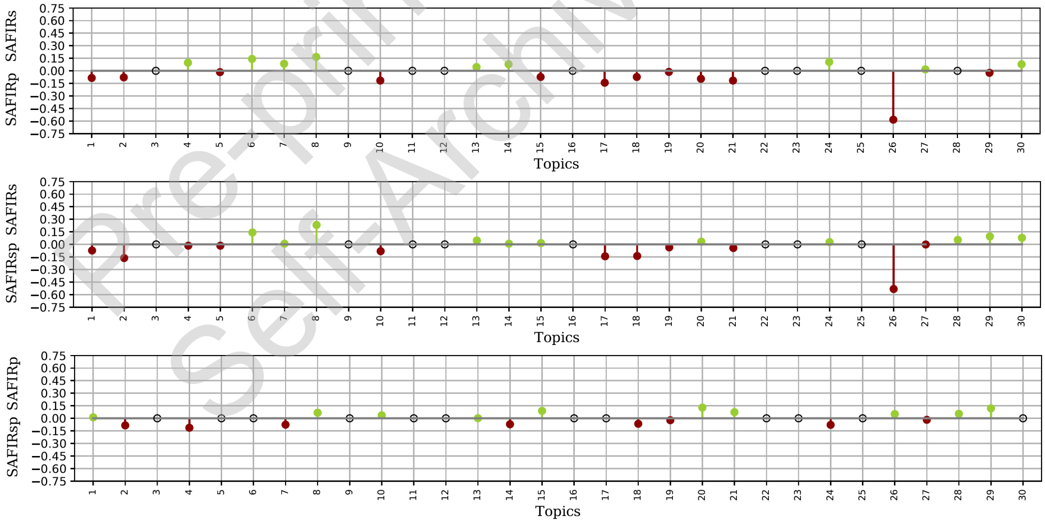


Fig. 4. Per-topic differences between SAFIR variants at nDCG@10 in CDS14 collection. The green (light) stems indicate that the upper-side SAFIR variant achieves a higher nDCG@10 score than the lower-side variant. Vice versa, red (dark) stems indicate that the lower-side SAFIR variant achieves a higher nDCG@10 score than the upper-side variant.

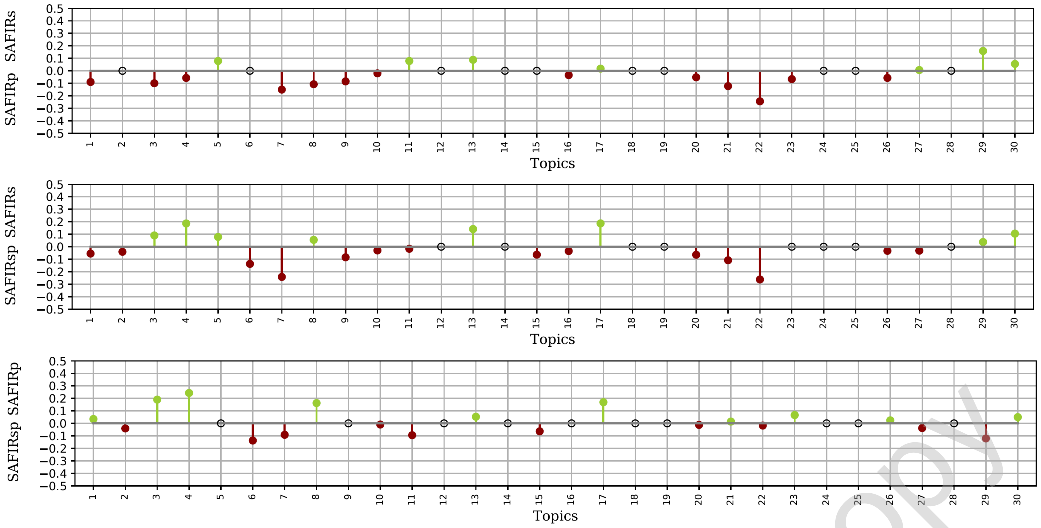


Fig. 5. Per-topic differences between SAFIR variants at nDCG@10 in CDS15 collection. The green (light) stems indicate that the upper-side SAFIR variant achieves a higher nDCG@10 score than the lower-side variant. Vice versa, red (dark) stems indicate that the lower-side SAFIR variant achieves a higher nDCG@10 score than the upper-side variant.

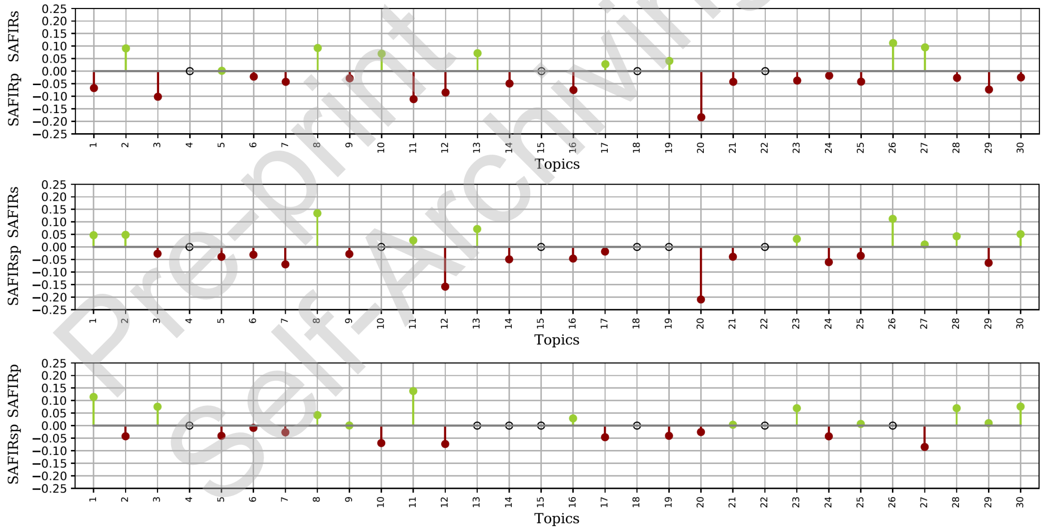


Fig. 6. Per-topic differences between SAFIR variants at nDCG@10 in CDS16 collection. The green (light) stems indicate that the upper-side SAFIR variant achieves a higher nDCG@10 score than the lower-side variant. Vice versa, red (dark) stems indicate that the lower-side SAFIR variant achieves a higher nDCG@10 score than the upper-side variant.

achieves a higher nDCG@10 score than SAFIR$_s$ in more queries (with per-topic difference ≥ 0.10). In particular, SAFIR$_p$ outperforms SAFIR$_s$ by a large margin (≥ 0.30) on topic OHSU14. If we analyze the degree of polysemy of topic OHSU14, we find out that 50% (two out of four) of the query words are polysemous. Thus, polysemy has a strong impact on this query and SAFIR$_p$ (but also SAFIR$_{sp}$) effectively captures it. A similar trend is found for topic OHSU7, where 75% (three out of four) of the words are polysemous. The smaller difference between SAFIR$_s$ and SAFIR$_p$, along with the fact that SAFIR$_{sp}$ achieves the best results among the three variants, suggest that both polysemy and synonymy impact on this query. Conversely, topic OHSU39 presents three polysemous words out of nine (30%). In this case, modeling polysemy impacts less on – or even harms – the query and other factors dominate the performances. We hypothesize that these factors are related to synonymy given the high performance of SAFIR$_s$ and the fact that SAFIR$_{sp}$ outperforms SAFIR$_p$.

In CDS14 (Figure 4), the results show a similar trend to OHSUMED. However, the differences between variants are smaller (with per-topic differences $\leq 0.15$). The only notable exception is topic 26, where both SAFIR$_p$ and SAFIR$_{sp}$ outperform SAFIR$_s$ by a large margin. In this query, 50% of the words are polysemous. Also, the fact that SAFIR$_p$ and SAFIR$_{sp}$ achieve nearly the same nDCG@10 score suggests that polysemy dominates performances on this query. Conversely, in topic 6, where SAFIR$_s$ outperforms both SAFIR$_p$ and SAFIR$_{sp}$, less than 30% of the words are polysemous. For this query, both SAFIR$_p$ and SAFIR$_{sp}$ achieve an nDCG@10 score of zero – which means that polysemy hurts performance even when jointly modeled with synonymy, as in SAFIR$_{sp}$.

The differences between SAFIR variants are smaller in CDS15 (Figure 5), where the largest difference between SAFIR$_s$ and the variants integrating polysemy is found for topic 22 with a value close to 0.30. Also this query presents a large number of polysemous words (50%). Again, SAFIR$_p$ and SAFIR$_{sp}$ achieve nearly the same nDCG@10 score. The fact that the difference is in favor of SAFIR$_{sp}$ indicates that the combination of both synonymy and polysemy is beneficial for this query.

Regarding CDS16 (Figure 6), the results are in line with those from CDS14 and CDS15. The query presenting the largest difference is topic 20, where SAFIR$_p$ and SAFIR$_{sp}$ outperform SAFIR$_s$ by a margin of 0.20. In this case, however, the number of polysemous words is lower than in the previous examples, with a percentage of polysemous words of 40%. As for topic 22 from CDS15, the difference between SAFIR$_p$ and SAFIR$_{sp}$ is in favor of SAFIR$_{sp}$. Finally, we highlight topic 26, where both SAFIR$_p$ and SAFIR$_{sp}$ achieve a score of zero for nDCG@10 – as opposed to SAFIR$_s$. Interestingly, topic 26 is an outlier in terms of query length, with a total of fifty-four words of which twenty-two are polysemous. The results for this query show that integrating synonymy is effective, whereas polysemy harms performances.

Thus, the analysis shows that SAFIR$_p$ and – to a lesser extent – SAFIR$_{sp}$ present a larger number of queries than SAFIR$_s$, where they achieve higher scores for nDCG@10. In particular, when the degree of polysemy within queries is high, SAFIR$_p$ and SAFIR$_{sp}$ effectively capture it and get high results for Precision-oriented measures. On the other hand, SAFIR$_p$ and SAFIR$_{sp}$ are outperformed by SAFIR$_s$ in some queries where the polysemy degree is low. In such cases, modeling synonymy is effective as opposed to polysemy – which leads to detrimental effects on performances.

**The Advantage of Modeling Polysemy**. Figures 7–10 highlight a behavior similar to the one found in the previous analysis. Each figure shows the per-topic differences between SAFIR variants and BM25/RM3 at nDCG@10 for a given collection. The green (light) stems indicate that the SAFIR variant achieves a higher nDCG@10 score than BM25/RM3. Vice versa, red (dark) stems indicate that BM25/RM3 achieves a higher nDCG@10 score than the SAFIR variant. Overall, the nDCG@10 results tend to split between SAFIR variants and BM25/RM3. However, the objective of this analysis is to verify whether the effectiveness of SAFIR$_p$ and SAFIR$_{sp}$ on highly polysemous queries holds against BM25/RM3. Therefore, we compare SAFIR variants with BM25/RM3 on the same highly polysemous queries discussed in the previous analysis.

Regarding OHSUMED (Figure 7), in topic OHSU14 – where 50% of the words are polysemous – neither SAFIR$_p$ nor SAFIR$_{sp}$ outperform BM25/RM3. However, both SAFIR variants present smaller

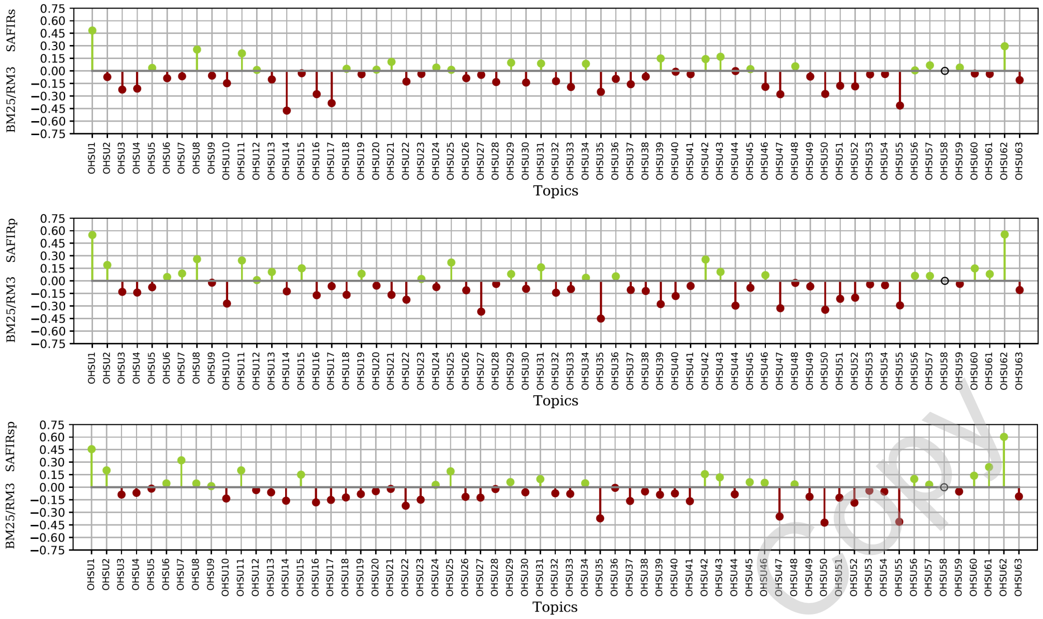


Fig. 7. Per-topic differences between SAFIR variants and BM25/RM3 for nDCG@10 in OHSUMED collection. The green (light) stems indicate that the SAFIR variant achieves a higher nDCG@10 score than BM25/RM3. Vice versa, red (dark) stems indicate that BM25/RM3 achieves a higher nDCG@10 score than the SAFIR variant.

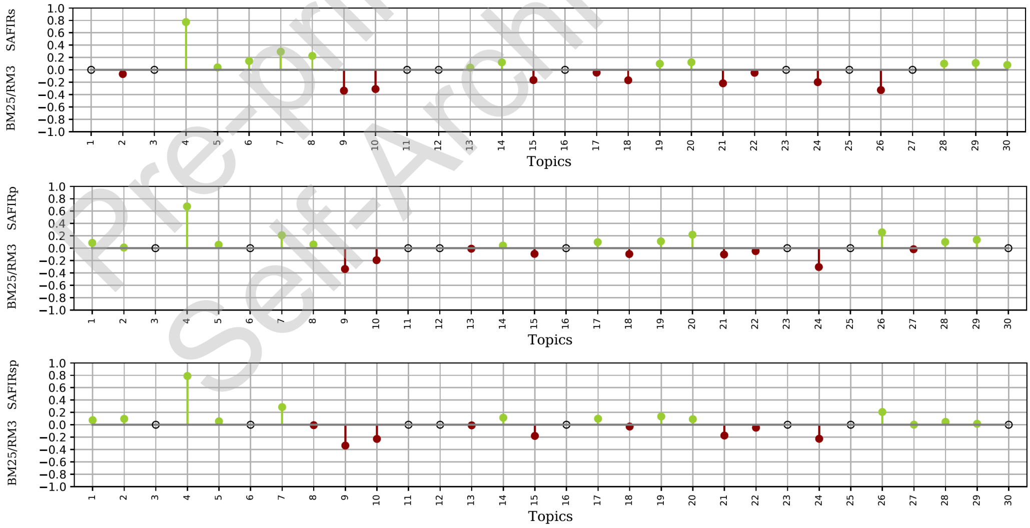


Fig. 8. Per-topic differences between SAFIR variants and BM25/RM3 for nDCG@10 in CDS14 collection. The green (light) stems indicate that the SAFIR variant achieves a higher nDCG@10 score than BM25/RM3. Vice versa, red (dark) stems indicate that BM25/RM3 achieves a higher nDCG@10 score than the SAFIR variant.

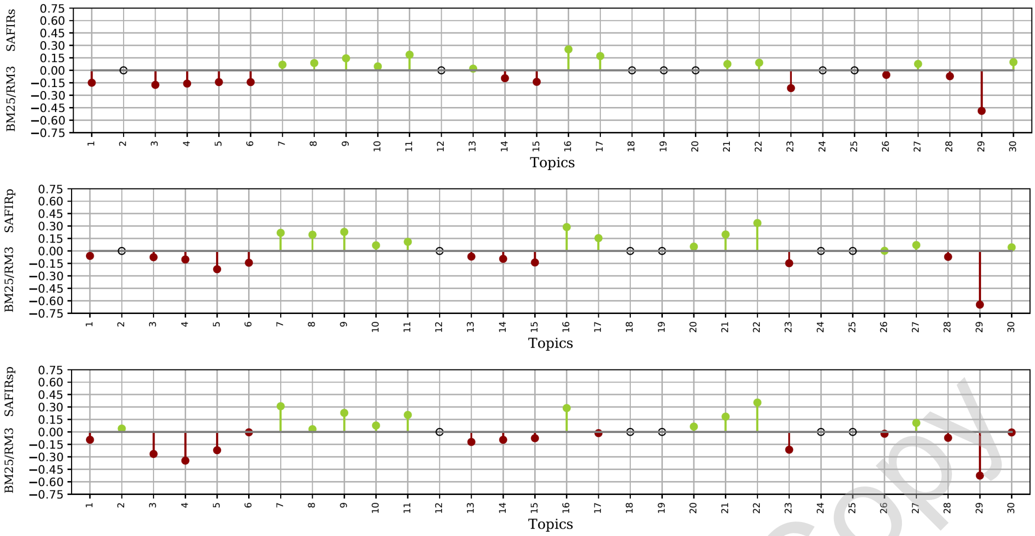


Fig. 9. Per-topic differences between SAFIR variants and BM25/RM3 for nDCG@10 in CDS15 collection. The green (light) stems indicate that the SAFIR variant achieves a higher nDCG@10 score than BM25/RM3. Vice versa, red (dark) stems indicate that BM25/RM3 achieves a higher nDCG@10 score than the SAFIR variant.

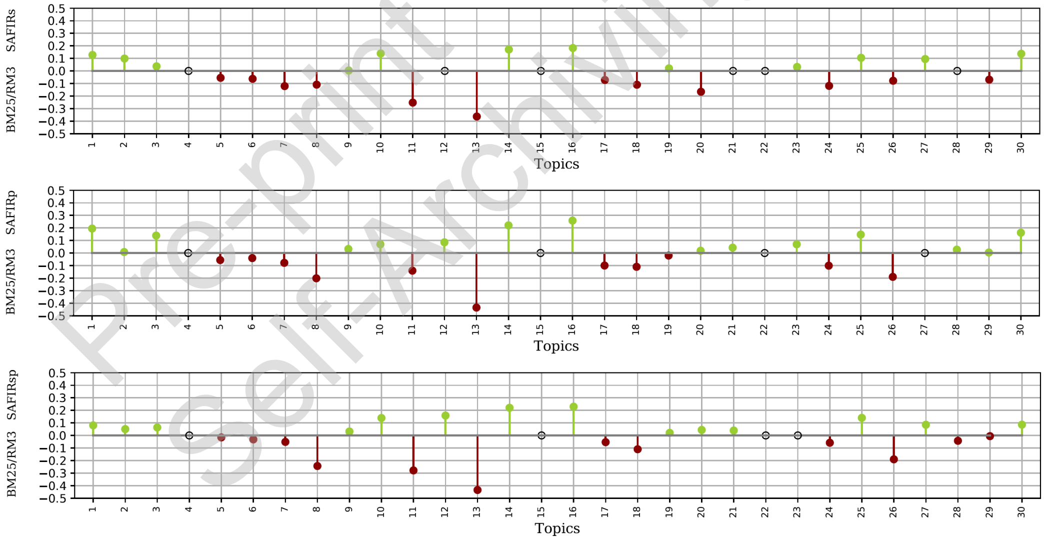


Fig. 10. Per-topic differences between SAFIR variants and BM25/RM3 for nDCG@10 in CDS16 collection. The green (light) stems indicate that the SAFIR variant achieves a higher nDCG@10 score than BM25/RM3. Vice versa, red (dark) stems indicate that BM25/RM3 achieves a higher nDCG@10 score than the SAFIR variant.

differences (about 0.15) with BM25/RM3 if compared to SAFIR$_s$ (> 0.45). Therefore, although not entirely, modeling polysemy helps SAFIR to bridge the performance gap with BM25/RM3. On the other hand, the results for topic OHSU7 – where 75% of the words are polysemous – show that SAFIR$_p$ and SAFIR$_{sp}$ outperform BM25/RM3. Besides, the fact that SAFIR$_s$ performs worse than BM25/RM3 indicates that this topic highly benefits from modeling polysemy.

In CDS14 (Figure 8), topic 26 (50% of polysemous words) shows a similar trend to topic OHSU7. Also in this case, SAFIR$_p$ and SAFIR$_{sp}$ outperform BM25/RM3 – with per-topic difference $\geq 0.20$ – whereas SAFIR$_s$ does not achieve competitive performance – with a gap of almost 0.40 with BM25/RM3. Compared to OHSU7, however, the impact of synonymy on this query is limited.

A different situation occurs for CDS15 (Figure 9), where all SAFIR variants outperform BM25/RM3 in topic 22 (50% of polysemous words). The positive performances of all SAFIR variants – and in particular of SAFIR$_{sp}$ – indicate that modeling both synonymy and polysemy is beneficial for this query.

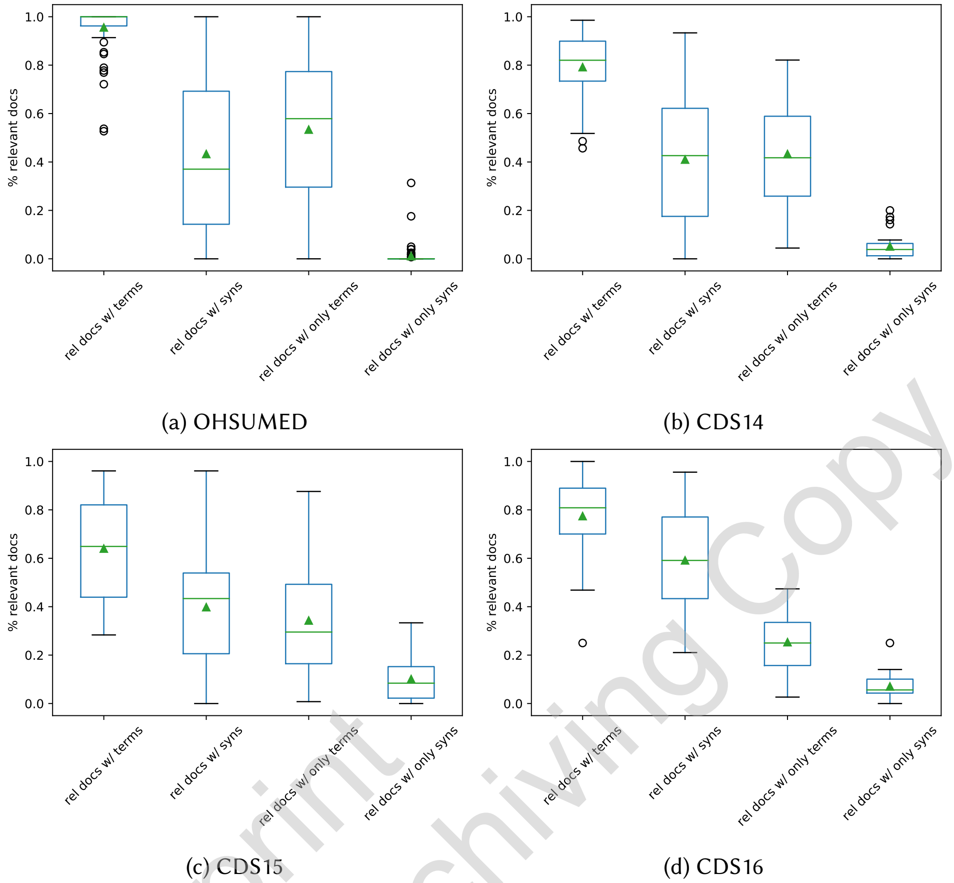
As for CDS16 (Figure 10), topic 20 – where 40% of the words are polysemous – shows similarities with topics OHSU7 (OHSUMED) and 26 (CDS14). Again, SAFIR$_p$ and SAFIR$_{sp}$ outperform BM25/RM3, while SAFIR$_s$ does not. However, the differences between SAFIR$_p$/SAFIR$_{sp}$ and BM25/RM3 are small if compared to the other topics discussed. A possible reason could be the lower number of polysemous words within this query – which leads to a minor impact of polysemy on model performances.

Thus, the analysis confirms the effectiveness of modeling polysemy to answer semantically hard queries with a high degree of polysemy. SAFIR$_p$ and SAFIR$_{sp}$ effectively capture polysemy and, for highly polysemous queries, outperform BM25/RM3.

### *5.1.2 Synonymy Analysis*

To identify the degree of synonymy in the considered collections, we account for (i) the proportion of relevant documents that contain at least one query term; (ii) the proportion of relevant documents that contain at least one synonym related to any query term; (iii) the proportion of relevant documents that contain only query terms; (iv) the proportion of relevant documents that contain only synonyms related to query terms.
Figure 11 shows the distribution of such proportions within each collection. Then, for each collection, we present one query where the integration of synonymy in the learning process produces effective results. Each query is selected to highlight this behavior and has the proportion of relevant documents containing synonyms close to or greater than the third quartile of the distribution generated from (ii). We report the results in Table 6 as a pairwise comparison between NVSM/SAFIR$_s$ and SAFIR$_p$/SAFIR$_{sp}$. In this way, we emphasize the effectiveness of integrating synonymy by comparing pairs of models that rely on the same input data. As done for polysemy, we perform a second analysis where we compare SAFIR variants, NVSM, and BM25/RM3 on semantically hard queries. The objective is to understand whether the effectiveness of SAFIR$_s$ and SAFIR$_{sp}$ on queries with a large proportion of relevant documents containing only query synonyms holds against NVSM and BM25/RM3. For each collection, we consider the five queries that present the largest proportion of relevant documents containing only query synonyms (iv) and we present the results in Table 7.

**The Degree of Synonymy.** The distributions in Figure 11 provide two main insights. First, the proportion of relevant documents that contain only query synonyms is low for all queries in all collections. Therefore, modeling synonymy to retrieve relevant documents has a marginal impact on retrieval performances. Secondly, the proportion of relevant documents that contain at least one query synonym is, on average, lower than the proportion of relevant documents that contain at



(a) OHSUMED

(b) CDS14

(c) CDS15

(d) CDS16

Fig. 11. Distribution of different proportions of relevant documents per topic. For each collection, the box-plots represent, from left to right, the distribution of the proportion of relevant documents per topic containing query terms, per topic containing synonyms related to query terms, per topic containing only query terms, and per topic containing only synonyms related to query terms.

least one query term for all collections. Besides, the proportion of relevant documents that contain only query terms is, on average, close to the proportion of relevant documents that contain at least one query synonym for all collections but CDS16. In practice, this means that the impact of synonymy is also mitigated by the large number of relevant documents that contain (only) query terms. As a side note, the proportion of relevant documents that contain at least one query term has a median value of 1.0 in OHSUMED. This further explains the effectiveness of models relying on corpus-based features – and in particular of bag-of-words models.

Thus, the analysis explains the low – or even detrimental – impact of integrating synonymy and shows why SAFIR$_s$ and SAFIR$_{sp}$ present average performances close to or lower than those of NVSM and SAFIR$_p$, respectively. Nevertheless, we want to understand if modeling synonymy proves effective when the proportion of relevant documents that contain query synonyms is high. Our intuition is that for queries with a large number of relevant documents containing query synonyms, the effectiveness will be higher for SAFIR$_s$ and SAFIR$_{sp}$ than for NVSM and SAFIR$_p$, respectively.

Table 6. Pairwise comparison between SAFIR$_s$/NVSM and SAFIR$_{sp}$/SAFIR$_p$ on specific topics that present a large number of relevant documents containing query synonyms. For each measure, ↑/↓ means that the SAFIR variant integrating synonymy achieves higher/lower scores than its baseline.

OHSUMED - Topic OHSU22

| | infNDCG | nDCG@1000 | nDCG@100 | nDCG@10 | P@10 | Recall@1000 |
|---|---|---|---|---|---|---|
| NVSM | – | 0.4267 | 0.2440 | 0.0526 | 0.1000 | 0.9600 |
| SAFIR$_s$ | – | $0.4795^{\uparrow}$ | $0.2906^{\uparrow}$ | $0.1414^{\uparrow}$ | $0.3000^{\uparrow}$ | $1.0000^{\uparrow}$ |
| SAFIR$_p$ | – | 0.4177 | 0.2491 | 0.0435 | 0.1000 | 0.9600 |
| SAFIR$_{sp}$ | – | $0.4457^{\uparrow}$ | $0.2820^{\uparrow}$ | $0.0473^{\uparrow}$ | 0.1000 | $1.0000^{\uparrow}$ |

CDS14 - Topic 24

| | infNDCG | nDCG@1000 | nDCG@100 | nDCG@10 | P@10 | Recall@1000 |
|---|---|---|---|---|---|---|
| NVSM | 0.3118 | 0.5225 | 0.3300 | 0.2941 | 0.4000 | 0.7222 |
| SAFIR$_s$ | $0.3797^{\uparrow}$ | $0.5877^{\uparrow}$ | $0.4018^{\uparrow}$ | $0.4291^{\uparrow}$ | 0.4000 | $0.7333^{\uparrow}$ |
| SAFIR$_p$ | 0.3281 | 0.5373 | 0.3472 | 0.3236 | 0.4000 | 0.7111 |
| SAFIR$_{sp}$ | $0.3338^{\uparrow}$ | $0.5631^{\uparrow}$ | $0.3532^{\uparrow}$ | $0.4019^{\uparrow}$ | $0.5000^{\uparrow}$ | $0.7222^{\uparrow}$ |

CDS15 - Topic 19

| | infNDCG | nDCG@1000 | nDCG@100 | nDCG@10 | P@10 | Recall@1000 |
|---|---|---|---|---|---|---|
| NVSM | 0.0075 | 0.1222 | 0.0092 | 0.0000 | 0.0000 | 0.2159 |
| SAFIR$_s$ | $0.0171^{\uparrow}$ | $0.1358^{\uparrow}$ | $0.0209^{\uparrow}$ | 0.0000 | 0.0000 | $0.2500^{\uparrow}$ |
| SAFIR$_p$ | 0.0088 | 0.0985 | 0.0108 | 0.0000 | 0.0000 | 0.1705 |
| SAFIR$_{sp}$ | $0.0102^{\uparrow}$ | $0.1243^{\uparrow}$ | $0.0125^{\uparrow}$ | 0.0000 | 0.0000 | $0.2273^{\uparrow}$ |

CDS16 - Topic 26

| | infNDCG | nDCG@1000 | nDCG@100 | nDCG@10 | P@10 | Recall@1000 |
|---|---|---|---|---|---|---|
| NVSM | 0.0899 | 0.2370 | 0.1058 | 0.0347 | 0.1000 | 0.3363 |
| SAFIR$_s$ | $0.1288^{\uparrow}$ | $0.2735^{\uparrow}$ | $0.1359^{\uparrow}$ | $0.1120^{\uparrow}$ | $0.2000^{\uparrow}$ | $0.3717^{\uparrow}$ |
| SAFIR$_p$ | 0.0385 | 0.1928 | 0.0453 | 0.0000 | 0.0000 | 0.3009 |
| SAFIR$_{sp}$ | $0.0486^{\uparrow}$ | $0.2084^{\uparrow}$ | $0.0571^{\uparrow}$ | 0.0000 | 0.0000 | $0.3186^{\uparrow}$ |

**The Impact of Modeling Synonymy**. The results from Table 6 confirm this intuition and show the ability of SAFIR$_s$ and SAFIR$_{sp}$ to retrieve relevant documents that NVSM and SAFIR$_p$ fail to discover. In particular, SAFIR$_s$ and SAFIR$_{sp}$ achieve 100% Recall@1000 for topic OHSU22 (OHSUMED), thus retrieving all the relevant documents that neither NVSM nor SAFIR$_p$ discover. Furthermore, the results for nDCG measures show the ability of SAFIR$_s$ and SAFIR$_{sp}$ to effectively order relevant documents in the ranking list. For instance, SAFIR$_s$ achieves a 0.4291 of nDCG@10 in topic 24 (CDS14), whereas NVSM achieves 0.2941. Similarly, SAFIR$_{sp}$ achieves a 0.4019 of nDCG@10 that outperforms SAFIR$_p$. To a lesser extent, the results for topic 19 (CDS15) follow the same trend found in the topics analyzed for OHSUMED and CDS14. The main difference regards nDCG@10 and P@10 measures, where SAFIR$_s$ and SAFIR$_{sp}$ achieve the same performances of NVSM and SAFIR$_p$. In this case, SAFIR variants and NVSM fail to order relevant documents in the top positions of the ranking list. Finally, the results for topic 26 (CDS16) confirm the findings from the polysemy analysis on this query and highlight the effectiveness of integrating synonymy.

Table 7. Retrieval performances of SAFIR variants, NVSM, and BM25/RM3 on the five topics that present the largest number of relevant documents containing only query synonyms. For each measure, **bold** represents the model with the highest score.

OHSUMED - Topics OHSU63, OHSU31, OHSU54, OHSU22, OHSU32

| | infNDCG | nDCG@1000 | nDCG@100 | nDCG@10 | P@10 | Recall@1000 |
|---|---|---|---|---|---|---|
| BM25/RM3 | – | **0.3926** | **0.2729** | **0.2480** | **0.3600** | 0.6285 |
| NVSM | – | 0.3383 | 0.2248 | 0.1272 | 0.1800 | 0.5875 |
| SAFIR$_s$ | – | 0.3892 | 0.2127 | 0.1860 | 0.2400 | **0.6843** |
| SAFIR$_p$ | – | 0.3725 | 0.2438 | 0.1744 | 0.2200 | 0.6304 |
| SAFIR$_{sp}$ | – | 0.3832 | 0.2457 | 0.1762 | 0.2200 | 0.6637 |

CDS14 - Topics 16, 28, 13, 5, 24

| | infNDCG | nDCG@1000 | nDCG@100 | nDCG@10 | P@10 | Recall@1000 |
|---|---|---|---|---|---|---|
| BM25/RM3 | 0.1241 | 0.2127 | 0.1280 | **0.1482** | **0.2000** | 0.29854 |
| NVSM | 0.1031 | 0.2236 | 0.1137 | 0.1175 | 0.1600 | **0.3395** |
| SAFIR$_s$ | **0.1268** | **0.2388** | **0.1436** | 0.1434 | 0.1400 | 0.3313 |
| SAFIR$_p$ | 0.1091 | 0.2184 | 0.1205 | 0.1165 | 0.1600 | 0.3124 |
| SAFIR$_{sp}$ | 0.1149 | 0.2253 | 0.1323 | 0.1211 | 0.1600 | 0.3179 |

CDS15 - Topics 7, 24, 16, 13, 11

| | infNDCG | nDCG@1000 | nDCG@100 | nDCG@10 | P@10 | Recall@1000 |
|---|---|---|---|---|---|---|
| BM25/RM3 | 0.1080 | 0.1166 | 0.0976 | 0.1515 | 0.2000 | 0.1363 |
| NVSM | 0.1570 | 0.1785 | 0.1489 | 0.2357 | 0.3200 | 0.2154 |
| SAFIR$_s$ | 0.1601 | 0.1822 | 0.1570 | 0.2575 | 0.3400 | 0.2202 |
| SAFIR$_p$ | **0.1794** | 0.1979 | 0.1578 | 0.2612 | 0.3200 | 0.2295 |
| SAFIR$_{sp}$ | 0.1768 | **0.1986** | **0.1633** | **0.2878** | **0.3600** | **0.2348** |

CDS16 - Topics 22, 1, 5, 12, 13

| | infNDCG | nDCG@1000 | nDCG@100 | nDCG@10 | P@10 | Recall@1000 |
|---|---|---|---|---|---|---|
| BM25/RM3 | **0.1192** | **0.1701** | **0.1090** | **0.1443** | **0.2000** | 0.2319 |
| NVSM | 0.0642 | 0.1469 | 0.0626 | 0.0712 | 0.1600 | 0.2492 |
| SAFIR$_s$ | 0.0917 | 0.1503 | 0.0718 | 0.0861 | 0.1800 | 0.2270 |
| SAFIR$_p$ | 0.0839 | 0.1628 | 0.0722 | 0.1021 | 0.1400 | **0.2522** |
| SAFIR$_{sp}$ | 0.0900 | 0.1584 | 0.0683 | 0.1020 | 0.1400 | 0.2438 |

**The Advantage of Modeling Synonymy**. Given the outcomes of the previous analysis, we investigate whether the effectiveness of SAFIR$_s$ and SAFIR$_{sp}$ on semantically hard queries – i.e., queries with a large number of relevant documents containing only query synonyms – holds against NVSM and BM25/RM3. The results from Table 7 mark a clear distinction between OHSUMED, CDS16, and CDS14, CDS15. In OHSUMED and CDS16, BM25/RM3 achieves top performances for most measures. The only notable exception is Recall@1000, where SAFIR$_s$ and SAFIR$_{sp}$ outperform BM25/RM3 by a large margin. On the other hand, the results for CDS14 and CDS15 highlight the effectiveness of SAFIR$_s$ (CDS14) and SAFIR$_{sp}$ (CDS15) to answer semantically hard queries. In particular, SAFIR$_{sp}$ achieves top performances for all measures but infNDCG in CDS15.

If we analyze the proportion of relevant documents containing only query synonyms in the considered queries, we discover that the differences between OHSUMED, CDS16, and CDS14, CDS15 are related to such quantities. In both OHSUMED and CDS16, the number of relevant documents containing only query synonyms is less than 15% of the total number of relevant documents for three out of five queries. Conversely, CDS14 and CDS15 show proportions higher than 15% for most queries. In particular, all the five CDS15 queries present proportions greater than 20% – and close to 30% in three out of five cases.

Therefore, when the proportion of relevant documents containing only query synonyms is considerable, SAFIR$_s$ and SAFIR$_{sp}$ effectively capture synonymy and provide better results to semantically hard queries than NVSM and BM25/RM3 – which do not explicitly model synonymy. However, compared to polysemy, the degree of synonymy is limited. Thus, the impact of modeling synonymy is marginal on average.

**Take-home message**. Modeling polysemy is effective and impacts the most when queries present a high degree of polysemy. On the other hand, the impact of synonymy on average performances is marginal – or even detrimental – due to the limited presence of relevant documents containing (only) query synonyms. Nevertheless, when we look at queries with a large number of relevant documents containing (only) query synonyms, SAFIR$_s$ and SAFIR$_{sp}$ capture synonymy and provide effective results.

### 5.2 The Effectiveness of Knowledge Resources for Document Retrieval

**RQ2** How can external knowledge resources help to bridge the semantic gap between queries and documents?

When we compare knowledge-enhanced models with the corpus-driven baselines used as part of their learning process, we observe different trends. Regarding knowledge-enhanced baselines, we see that retrofitting models fail to enhance the baselines used as part of their learning process. rword2vec performs worse than word2vec for most measures in all collections. Similarly, rdoc2vec fails to improve on both its baselines and performs worse than doc2vec or cdoc2vec for most measures in all collections. The optimization function used by rdoc2vec retrofits document representations but leaves word and concept representations unchanged. Therefore, document and word/concept representations – that were jointly learned by doc2vec/cdoc2vec – misalign. This leads to a mismatch between retrofitted document representations and word/concept representations, which can explain the suboptimal performances achieved by rdoc2vec. On the other hand, the results show that cdoc2vec benefits from learning concepts rather than words for most measures in CDS collections. Conversely, cdoc2vec achieves significantly worse results than doc2vec in OHSUMED. The reason of this significant drop in performances can be attributed to how cdoc2vec builds query representations. In fact, cdoc2vec relies only on the concepts associated to the query terms to build query representations. Therefore, given the short length of OHSUMED queries, this building process leads to noneffective representations.

As for SAFIR, we see that all the variants outperform NVSM for most measures in all collections. Depending on the measure and collection considered, different SAFIR variants achieve the best results. Interestingly, the results for nDCG@10 show that all SAFIR variants order relevant documents in top positions better than NVSM. This highlights the effectiveness, for Precision-oriented measures, of integrating external knowledge while optimizing word, concept, and document representations for retrieval. Of all the SAFIR variants, NVSM gets closer to SAFIR$_s$ performances. In particular, SAFIR$_s$ performs worse than NVSM for nDCG@1000 and Recall@1000 both in CDS14 and CDS16. As seen in the synonymy analysis, SAFIR$_s$ performance is impacted by the limited presence of relevant documents containing (only) query synonyms. However, the fact that SAFIR$_\text{s}$ outperforms NVSM for infNDCG in all CDS collections suggests that, with a larger number of relevance judgments, SAFIR$_\text{s}$ could achieve higher nDCG values than NVSM. Indeed, we recall that infNDCG provides a better estimate of the real value of nDCG in case of incomplete relevance judgments [77].

Compared to the other knowledge-enhanced models, we emphasize the effectiveness of SAFIR for Recall@1000 and nDCG measures. Recall@1000 shows the ability of SAFIR variants to retrieve relevant documents while nDCG measures show how well these documents are ranked at different cutoff levels. Therefore, we perform the following analyses to further evaluate the differences between SAFIR and the considered baselines.

### *5.2.1 Knowledge-Enhanced Relevance Analysis*

We analyze the number of relevant documents retrieved by SAFIR variants and knowledge-enhanced baselines. For each topic of CDS collections, we compare the number of exclusive relevant documents that only SAFIR retrieves with respect to the union of the relevant documents retrieved by all the knowledge-enhanced baselines. This means that we compare SAFIR against a "fictitious and boosted" model that considers all the relevant documents retrieved by the knowledge-enhanced baselines. We adopt this solution instead of comparing SAFIR individually with each knowledge-enhanced baseline to save space and compare SAFIR with a highly challenging baseline. Figure 12 reports the per-topic results of this analysis.

The analysis shows how exclusive SAFIR is in retrieving relevant documents that none of the knowledge-enhanced baselines retrieve. SAFIR retrieves more exclusive relevant documents than all the other knowledge-enhanced models together. In particular, SAFIR variants retrieve more exclusive relevant documents for almost all the topics of CDS14 and CDS16 collections. The exclusiveness of SAFIR is less evident in CDS15, as the impact of rword2vec in the "fictitious" model reduces the gap with SAFIR. If we analyze the per-topic behavior of each SAFIR variant, we observe that, with the due differences, all variants have a similar trend in terms of exclusive relevant documents retrieved. This suggests that SAFIR prioritizes text matching when learning representations and relies on polysemy and synonymy to refine such representations towards one, or both, features.

We present three examples – one for each SAFIR variant – where we qualitatively analyze the impact of knowledge resources in modeling synonymy, polysemy, or both. Each example represents a query where a particular SAFIR variant retrieves the highest number of exclusive relevant documents compared to the other variants and the fictitious model. Then, we present a fourth example where SAFIR is outperformed by the fictitious model, which retrieves more exclusive relevant documents.

**Example 5.1. CDS14 Topic 4**: "*2-year-old boy with fever and irritability for 5 days. Physical exam findings include conjunctivitis, strawberry tongue, and desquamation of the fingers and toes. Lab results include low albumin, elevated white blood cell count and C-reactive protein, and urine leukocytes. Echo shows moderate dilation of the coronary arteries.*"

- {SAFIR$_\text{s}$} \ {knowledge-enhanced baselines (union)}: 25
- {knowledge-enhanced baselines (union)} \ {SAFIR$_\text{s}$}: 0

For topic 4 of CDS14, SAFIR$_\text{s}$ retrieves twenty-five documents that the fictitious model does not retrieve. Conversely, the fictitious model does not retrieve any relevant documents that SAFIR$_\text{s}$ does not retrieve. For this query, an interesting example is provided by document `3152734`. Document `3152734` describes common associated symptoms (e.g., strawberry tongue) and their clinical

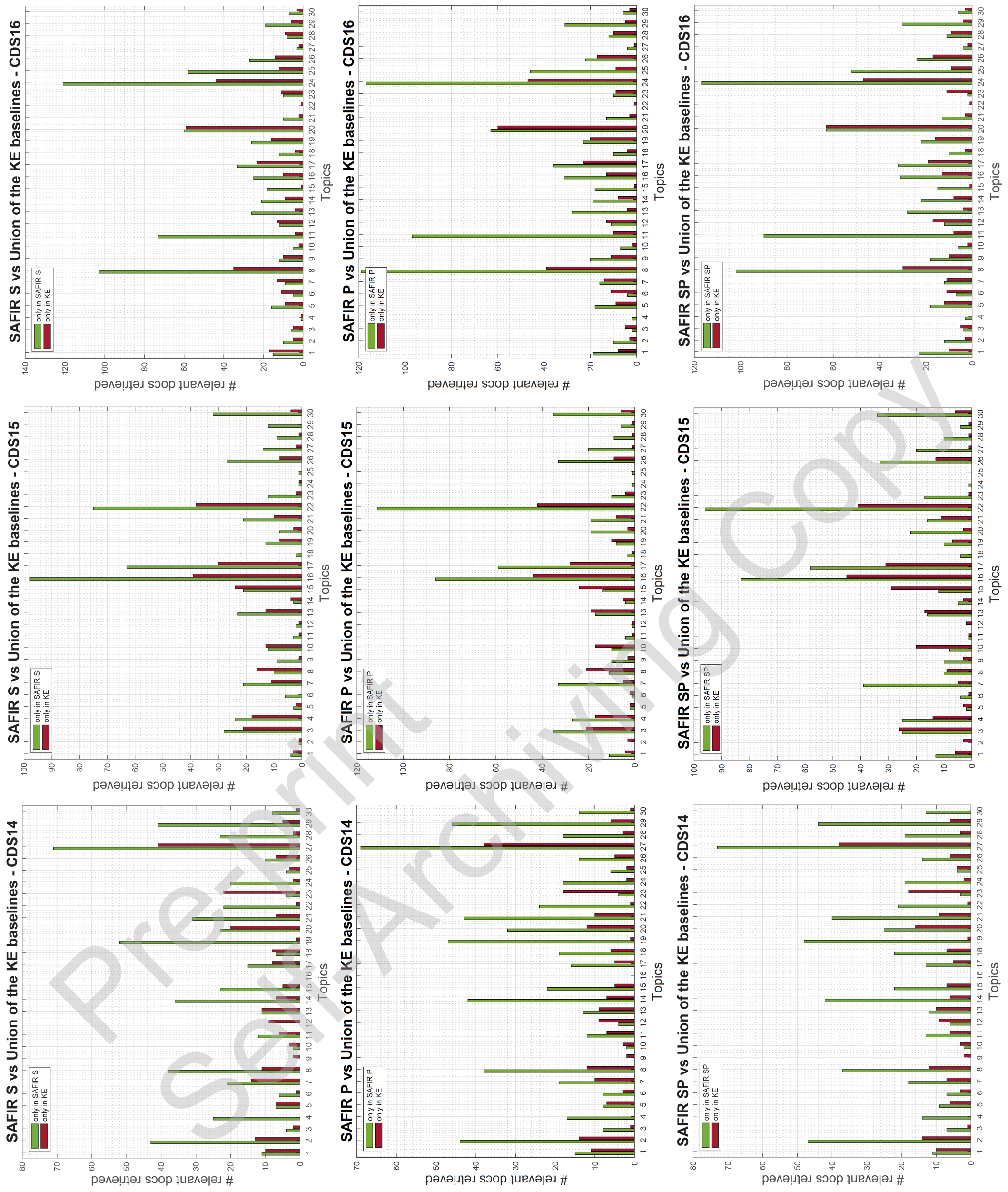


Fig. 12. Per-topic analysis of the number of relevant documents retrieved by SAFIR variants and by the union of the knowledge-enhanced baselines. For each topic, the green (light) bar represents the number of relevant documents that only SAFIR retrieves and the red (dark) bar represents the union of the number of relevant documents that the knowledge-enhanced baselines retrieve.

significance in children affected with the Kawaski disease. The document contains words like "children" and "febrile", which convey the same meaning of query words "boy" and "fever". Therefore, by modeling synonymy, SAFIR$_s$ reduces the semantic gap between the query and this (relevant) document and improves retrieval. This document is not retrieved by SAFIR$_p$, which does not model synonymy, and neither by bag-of-words models, since query words are not contained within it.

**Example 5.2. CDS15 Topic 22**: "*A 65-year-old male complains of productive cough with tinges of blood. Chest X-ray reveals a round opaque mass within a cavity in his lung. Culture of the sputum revealed fungal elements.*"

- {SAFIR$_p$} \ {knowledge-enhanced baselines (union)}: 111
- {knowledge-enhanced baselines (union)} \ {SAFIR$_p$}: 42

For topic 22 of CDS15, SAFIR$_p$ retrieves 111 documents that the fictitious model does not retrieve. On the other hand, the fictitious model retrieves forty-two relevant documents that SAFIR$_p$ does not. Among the unique relevant documents retrieved by SAFIR$_p$, document `3014676` presents interesting aspects. Document `3014676` describes treatments for the allergic bronchopulmonary aspergillosis. The disease derives from the Aspergillus, a soil-dwelling fungus known to cause significant pulmonary infection in immunocompromised patients. The document presents various acronyms and morphosyntactic variants. In particular, the acronym "ABPA" – which stands for "Allergic Bronchopulmonary Aspergillosis" – can be especially ambiguous for an automatic system. In fact, within UMLS the acronym "ABPA" can be associated to five different meanings (CUIs) like: "Aspergillosis, Allergic Bronchopulmonary" (`C0004031`), "FLNC gene" (`C1414637`), and "AbpA protein, Streptococcus gordonii" (`C1308582`). Therefore, to relate such word to discriminative words within the query (e.g., the query words "lung" and "fungal") it is important to disambiguate its meaning. By modeling polysemy, SAFIR$_p$ removes this ambiguity in document and query words and improves retrieval. It is worth mentioning that this document is not retrieved by SAFIR$_s$, which does not model polysemy, and neither by bag-of-words models.

**Example 5.3. CDS16 Topic 14**: "*A 52 year-old woman with history of COPD and breast cancer who presents with SOB, hypoxia, cough, fevers and sore throat for several weeks.*"

- {SAFIR$_{sp}$} \ {knowledge-enhanced baselines (union)}: 22
- {knowledge-enhanced baselines (union)} \ {SAFIR$_{sp}$}: 8

For topic 14 of CDS16, SAFIR$_{sp}$ retrieves twenty-two documents that the fictitious model does not retrieve. Instead, the fictitious model retrieves eight relevant documents that SAFIR$_{sp}$ does not find. The query presents two interesting acronyms: COPD and SOB. The former stands for chronic obstructive pulmonary disease, whereas the latter for shortness of breath. COPD is a type of obstructive lung disease characterized by long-term breathing problems and poor airflow. In COPD, shortness of breath is a common respiratory symptom. Therefore, both acronyms need to be correctly disambiguated to retrieve relevant documents associated with them. We focus on document `3266210`, which describes a clinical trial for the treatment of COPD. In particular, document `3266210` contains the word "dyspnoea" – which is a synonym of SOB. Thus, by modeling both synonymy and polysemy together, SAFIR$_{sp}$ encodes semantic features required to effectively retrieve this document. Interestingly, SAFIR$_{sp}$ is the only variant retrieving this document.

**Example 5.4. CDS14 Topic 23**: "*63-year-old heavy smoker with productive cough, shortness of breath, tachypnea, and oxygen requirement. Chest x-ray shows hyperinflation with no consolidation.*" For topic 23 of CDS14, none of the SAFIR variants retrieve more than four documents that the fictitious model does not retrieve. In particular, SAFIR$_s$ and SAFIR$_p$ retrieve four documents that the fictitious model does not find, whereas SAFIR$_{sp}$ only three. On the other hand, the fictitious model retrieves twenty-two documents that SAFIR$_s$ does not retrieve and eighteen documents that neither

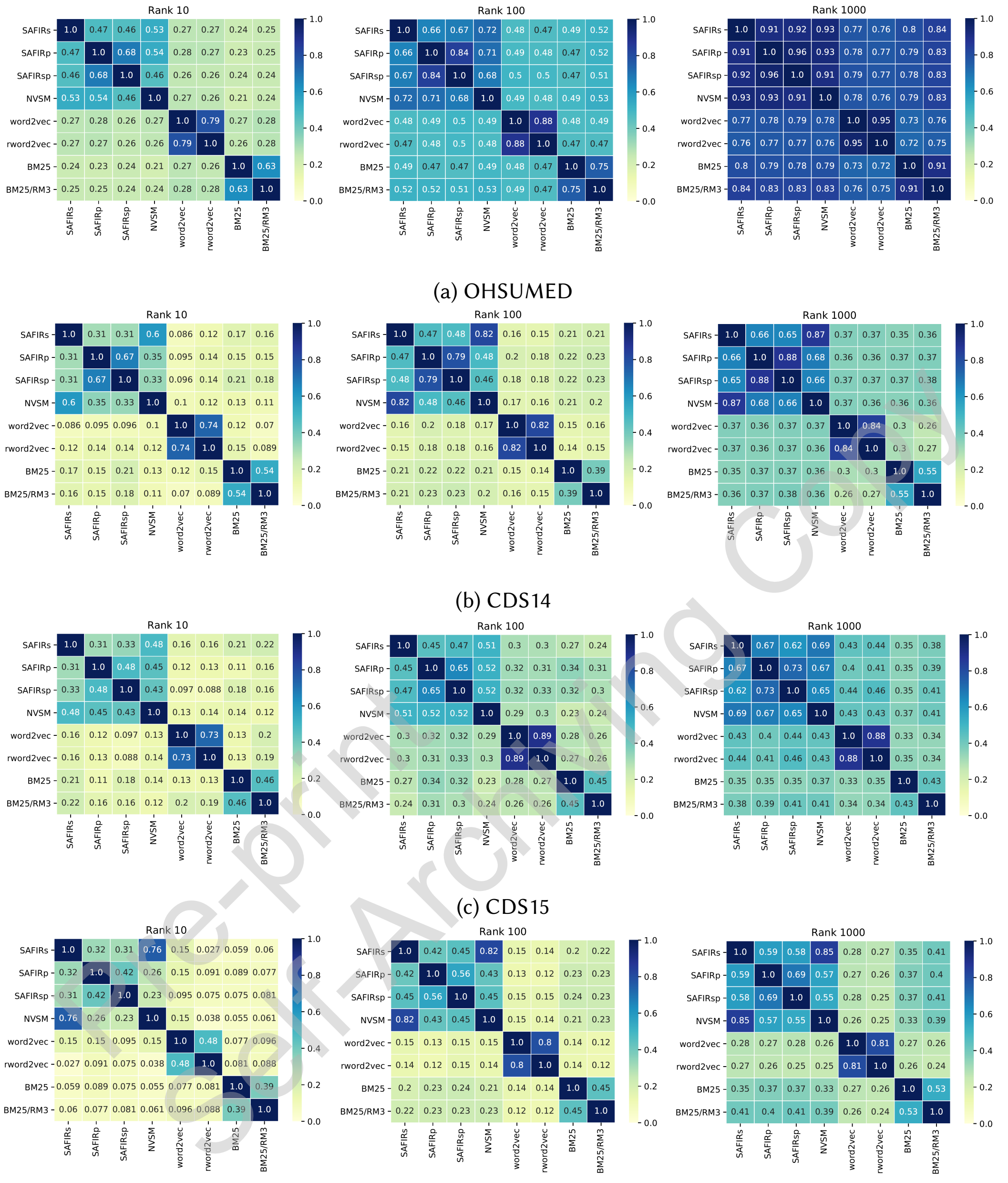
SAFIR$_{\mathrm{p}}$ nor SAFIR$_{\mathrm{sp}}$ retrieve. It is worth mentioning that the knowledge-enhanced baseline that impacts the most within the fictitious model is rword2vec. In fact, rword2vec retrieves nineteen of the twenty-two documents that SAFIR$_{\mathrm{s}}$ does not find and sixteen of the eighteen documents that neither SAFIR$_{\mathrm{p}}$ nor SAFIR$_{\mathrm{sp}}$ discover.

### *5.2.2 Relevance Similarity Analysis*

We evaluate to what extent SAFIR variants and the considered baselines retrieve different relevant documents. For each collection and pair of models, we compute the mean Jaccard index between the sets of relevant documents retrieved at different cutoffs. Given a pair of models, we compute the per-topic Jaccard index as the cardinality of the intersection divided by the cardinality of the union of the sets of relevant documents retrieved by the two considered models at a given cutoff. Then, the mean Jaccard index takes the average of the per-topic indices computed at the corresponding cutoff. When computing the mean Jaccard index, we do not count topics where none of the two considered models retrieve relevant documents (i.e., missing values). We use the same cutoff values used for nDCG measures, that is 10, 100, and 1000. In particular, we evaluate the degree of similarity between models that exhibit similar average performances. We want to understand to what extent these models retrieve different relevant documents. We do not report Jaccard index values for QLM and doc2vec models to save space and ease visualization. The performances of QLM are always comparable or lower than those of BM25, whereas doc2vec models never belong to the top statistical group (†).

Figure 13 shows the heatmaps of the mean Jaccard indices between the sets of relevant documents retrieved by each pair of models across topics at cutoffs 10, 100, and 1000, respectively, for each collection. The heatmaps highlight three clusters of models with higher similarity scores. The first cluster is composed of SAFIR variants and NVSM, the second of word2vec and rword2vec, whereas the third one comprises BM25 and BM25/RM3. Within the first cluster, the NVSM/SAFIR$_{\mathrm{s}}$ and SAFIR$_{\mathrm{p}}$/SAFIR$_{\mathrm{sp}}$ pairs show higher scores due to the inherent similarity between the models. Nevertheless, we observe that NVSM/SAFIR$_{\mathrm{s}}$ and SAFIR$_{\mathrm{p}}$/SAFIR$_{\mathrm{sp}}$ pairs never exceed a similarity score of 0.70 at cutoff 10. The only exception is in CDS16, where the NVSM/SAFIR$_{\mathrm{s}}$ pair shows a similarity score of 0.76. Therefore, all these models retrieve a significant number of different relevant documents in top positions of the ranking list. This behavior is even more pronounced when we consider the similarity scores between NVSM and either SAFIR$_{\mathrm{p}}$ or SAFIR$_{\mathrm{sp}}$. In fact, the scores for the NVSM/SAFIR$_{\mathrm{p}}$ and NVSM/SAFIR$_{\mathrm{sp}}$ pairs keep low for all cutoffs in CDS collections – never exceeding values of 0.50, 0.55, and 0.70 at cutoffs 10, 100, and 1000, respectively. Within the second cluster, word2vec and rword2vec consistently present the same level of similarity for all cutoffs in all collections. The only exception is in CDS16, where they show a similarity score of 0.48 at rank 10. Within the third cluster, BM25 and BM25/RM3 show similarity scores lower than 0.60 in all CDS collections – regardless of the cutoff. This reflects the impact of expanding the original query with RM3, which enables BM25 to discover more relevant documents compared to the first round of retrieval.

Outside the clusters, the low similarity scores in CDS collections indicate that all the models retrieve different relevant documents regardless of the cutoff. Conversely, all the considered pairs present high similarity scores at cutoffs 100 and 1000 in OHSUMED. We attribute this behavior to two main reasons: (i) the high proportion of relevant documents that contain at least one query term in OHSUMED (see Figure 11), which favors models relying on corpus-based features; (ii) the small size of corpus and vocabulary in OHSUMED (see Table 1), which reduces the amount of polysemous and synonymous words within the collection.



Rank 10

| | SAFIRs | SAFIRp | SAFIRsp | NVSM | word2vec | rword2vec | BM25 | BM25/RM3 |
|---|---|---|---|---|---|---|---|---|
| SAFIRs | 1.0 | 0.47 | 0.46 | 0.53 | 0.27 | 0.27 | 0.24 | 0.25 |
| SAFIRp | 0.47 | 1.0 | 0.68 | 0.54 | 0.28 | 0.27 | 0.23 | 0.25 |
| SAFIRsp | 0.46 | 0.68 | 1.0 | 0.46 | 0.26 | 0.26 | 0.24 | 0.24 |
| NVSM | 0.53 | 0.54 | 0.46 | 1.0 | 0.27 | 0.26 | 0.21 | 0.24 |
| word2vec | 0.27 | 0.28 | 0.26 | 0.27 | 1.0 | 0.79 | 0.27 | 0.28 |
| rword2vec | 0.27 | 0.27 | 0.26 | 0.26 | 0.79 | 1.0 | 0.26 | 0.28 |
| BM25 | 0.24 | 0.23 | 0.24 | 0.21 | 0.27 | 0.26 | 1.0 | 0.63 |
| BM25/RM3 | 0.25 | 0.25 | 0.24 | 0.24 | 0.28 | 0.28 | 0.63 | 1.0 |

Rank 100

| | SAFIRs | SAFIRp | SAFIRsp | NVSM | word2vec | rword2vec | BM25 | BM25/RM3 |
|---|---|---|---|---|---|---|---|---|
| SAFIRs | 1.0 | 0.66 | 0.67 | 0.72 | 0.48 | 0.47 | 0.49 | 0.52 |
| SAFIRp | 0.66 | 1.0 | 0.84 | 0.71 | 0.49 | 0.48 | 0.47 | 0.52 |
| SAFIRsp | 0.67 | 0.84 | 1.0 | 0.68 | 0.5 | 0.5 | 0.47 | 0.51 |
| NVSM | 0.72 | 0.71 | 0.68 | 1.0 | 0.49 | 0.48 | 0.49 | 0.53 |
| word2vec | 0.48 | 0.49 | 0.5 | 0.49 | 1.0 | 0.88 | 0.48 | 0.49 |
| rword2vec | 0.47 | 0.48 | 0.5 | 0.48 | 0.88 | 1.0 | 0.47 | 0.47 |
| BM25 | 0.49 | 0.47 | 0.47 | 0.49 | 0.48 | 0.47 | 1.0 | 0.75 |
| BM25/RM3 | 0.52 | 0.52 | 0.51 | 0.53 | 0.49 | 0.47 | 0.75 | 1.0 |

Rank 1000

| | SAFIRs | SAFIRp | SAFIRsp | NVSM | word2vec | rword2vec | BM25 | BM25/RM3 |
|---|---|---|---|---|---|---|---|---|
| SAFIRs | 1.0 | 0.91 | 0.92 | 0.93 | 0.77 | 0.76 | 0.8 | 0.84 |
| SAFIRp | 0.91 | 1.0 | 0.96 | 0.93 | 0.78 | 0.77 | 0.79 | 0.83 |
| SAFIRsp | 0.92 | 0.96 | 1.0 | 0.91 | 0.79 | 0.77 | 0.78 | 0.83 |
| NVSM | 0.93 | 0.93 | 0.91 | 1.0 | 0.78 | 0.76 | 0.79 | 0.83 |
| word2vec | 0.77 | 0.78 | 0.79 | 0.78 | 1.0 | 0.95 | 0.73 | 0.76 |
| rword2vec | 0.76 | 0.77 | 0.77 | 0.76 | 0.95 | 1.0 | 0.72 | 0.75 |
| BM25 | 0.8 | 0.79 | 0.78 | 0.79 | 0.73 | 0.72 | 1.0 | 0.91 |
| BM25/RM3 | 0.84 | 0.83 | 0.83 | 0.83 | 0.76 | 0.75 | 0.91 | 1.0 |

(a) OHSUMED

Rank 10

| | SAFIRs | SAFIRp | SAFIRsp | NVSM | word2vec | rword2vec | BM25 | BM25/RM3 |
|---|---|---|---|---|---|---|---|---|
| SAFIRs | 1.0 | 0.31 | 0.31 | 0.6 | 0.086 | 0.12 | 0.17 | 0.16 |
| SAFIRp | 0.31 | 1.0 | 0.67 | 0.35 | 0.095 | 0.14 | 0.15 | 0.15 |
| SAFIRsp | 0.31 | 0.67 | 1.0 | 0.33 | 0.096 | 0.14 | 0.21 | 0.18 |
| NVSM | 0.6 | 0.35 | 0.33 | 1.0 | 0.1 | 0.12 | 0.13 | 0.11 |
| word2vec | 0.086 | 0.095 | 0.096 | 0.1 | 1.0 | 0.74 | 0.12 | 0.07 |
| rword2vec | 0.12 | 0.14 | 0.14 | 0.12 | 0.74 | 1.0 | 0.15 | 0.089 |
| BM25 | 0.17 | 0.15 | 0.21 | 0.13 | 0.12 | 0.15 | 1.0 | 0.54 |
| BM25/RM3 | 0.16 | 0.15 | 0.18 | 0.11 | 0.07 | 0.089 | 0.54 | 1.0 |

Rank 100

| | SAFIRs | SAFIRp | SAFIRsp | NVSM | word2vec | rword2vec | BM25 | BM25/RM3 |
|---|---|---|---|---|---|---|---|---|
| SAFIRs | 1.0 | 0.47 | 0.48 | 0.82 | 0.16 | 0.15 | 0.21 | 0.21 |
| SAFIRp | 0.47 | 1.0 | 0.79 | 0.48 | 0.2 | 0.18 | 0.22 | 0.23 |
| SAFIRsp | 0.48 | 0.79 | 1.0 | 0.46 | 0.18 | 0.18 | 0.22 | 0.23 |
| NVSM | 0.82 | 0.48 | 0.46 | 1.0 | 0.17 | 0.16 | 0.21 | 0.2 |
| word2vec | 0.16 | 0.2 | 0.18 | 0.17 | 1.0 | 0.82 | 0.15 | 0.16 |
| rword2vec | 0.15 | 0.18 | 0.18 | 0.16 | 0.82 | 1.0 | 0.14 | 0.15 |
| BM25 | 0.21 | 0.22 | 0.22 | 0.21 | 0.15 | 0.14 | 1.0 | 0.39 |
| BM25/RM3 | 0.21 | 0.23 | 0.23 | 0.2 | 0.16 | 0.15 | 0.39 | 1.0 |

Rank 1000

| | SAFIRs | SAFIRp | SAFIRsp | NVSM | word2vec | rword2vec | BM25 | BM25/RM3 |
|---|---|---|---|---|---|---|---|---|
| SAFIRs | 1.0 | 0.66 | 0.65 | 0.87 | 0.37 | 0.37 | 0.35 | 0.36 |
| SAFIRp | 0.66 | 1.0 | 0.88 | 0.68 | 0.36 | 0.36 | 0.37 | 0.37 |
| SAFIRsp | 0.65 | 0.88 | 1.0 | 0.66 | 0.37 | 0.37 | 0.37 | 0.38 |
| NVSM | 0.87 | 0.68 | 0.66 | 1.0 | 0.37 | 0.36 | 0.36 | 0.36 |
| word2vec | 0.37 | 0.36 | 0.37 | 0.37 | 1.0 | 0.84 | 0.3 | 0.26 |
| rword2vec | 0.37 | 0.36 | 0.37 | 0.36 | 0.84 | 1.0 | 0.3 | 0.27 |
| BM25 | 0.35 | 0.37 | 0.37 | 0.36 | 0.3 | 0.3 | 1.0 | 0.55 |
| BM25/RM3 | 0.36 | 0.37 | 0.38 | 0.36 | 0.26 | 0.27 | 0.55 | 1.0 |

(b) CDS14

Rank 10

| | SAFIRs | SAFIRp | SAFIRsp | NVSM | word2vec | rword2vec | BM25 | BM25/RM3 |
|---|---|---|---|---|---|---|---|---|
| SAFIRs | 1.0 | 0.31 | 0.33 | 0.48 | 0.16 | 0.16 | 0.21 | 0.22 |
| SAFIRp | 0.31 | 1.0 | 0.48 | 0.45 | 0.12 | 0.13 | 0.11 | 0.16 |
| SAFIRsp | 0.33 | 0.48 | 1.0 | 0.43 | 0.097 | 0.088 | 0.18 | 0.16 |
| NVSM | 0.48 | 0.45 | 0.43 | 1.0 | 0.13 | 0.14 | 0.14 | 0.12 |
| word2vec | 0.16 | 0.12 | 0.097 | 0.13 | 1.0 | 0.73 | 0.13 | 0.2 |
| rword2vec | 0.16 | 0.13 | 0.088 | 0.14 | 0.73 | 1.0 | 0.13 | 0.19 |
| BM25 | 0.21 | 0.11 | 0.18 | 0.14 | 0.13 | 0.13 | 1.0 | 0.46 |
| BM25/RM3 | 0.22 | 0.16 | 0.16 | 0.12 | 0.2 | 0.19 | 0.46 | 1.0 |

Rank 100

| | SAFIRs | SAFIRp | SAFIRsp | NVSM | word2vec | rword2vec | BM25 | BM25/RM3 |
|---|---|---|---|---|---|---|---|---|
| SAFIRs | 1.0 | 0.45 | 0.47 | 0.51 | 0.3 | 0.3 | 0.27 | 0.24 |
| SAFIRp | 0.45 | 1.0 | 0.65 | 0.52 | 0.32 | 0.31 | 0.34 | 0.31 |
| SAFIRsp | 0.47 | 0.65 | 1.0 | 0.52 | 0.32 | 0.33 | 0.32 | 0.3 |
| NVSM | 0.51 | 0.52 | 0.52 | 1.0 | 0.29 | 0.3 | 0.23 | 0.24 |
| word2vec | 0.3 | 0.32 | 0.32 | 0.29 | 1.0 | 0.89 | 0.28 | 0.26 |
| rword2vec | 0.3 | 0.31 | 0.33 | 0.3 | 0.89 | 1.0 | 0.27 | 0.26 |
| BM25 | 0.27 | 0.34 | 0.32 | 0.23 | 0.28 | 0.27 | 1.0 | 0.45 |
| BM25/RM3 | 0.24 | 0.31 | 0.3 | 0.24 | 0.26 | 0.26 | 0.45 | 1.0 |

Rank 1000

| | SAFIRs | SAFIRp | SAFIRsp | NVSM | word2vec | rword2vec | BM25 | BM25/RM3 |
|---|---|---|---|---|---|---|---|---|
| SAFIRs | 1.0 | 0.67 | 0.62 | 0.69 | 0.43 | 0.44 | 0.35 | 0.38 |
| SAFIRp | 0.67 | 1.0 | 0.73 | 0.67 | 0.4 | 0.41 | 0.35 | 0.39 |
| SAFIRsp | 0.62 | 0.73 | 1.0 | 0.65 | 0.44 | 0.46 | 0.35 | 0.41 |
| NVSM | 0.69 | 0.67 | 0.65 | 1.0 | 0.43 | 0.43 | 0.37 | 0.41 |
| word2vec | 0.43 | 0.4 | 0.44 | 0.43 | 1.0 | 0.88 | 0.33 | 0.34 |
| rword2vec | 0.44 | 0.41 | 0.46 | 0.43 | 0.88 | 1.0 | 0.35 | 0.34 |
| BM25 | 0.35 | 0.35 | 0.35 | 0.37 | 0.33 | 0.35 | 1.0 | 0.43 |
| BM25/RM3 | 0.38 | 0.39 | 0.41 | 0.41 | 0.34 | 0.34 | 0.43 | 1.0 |

(c) CDS15

Rank 10

| | SAFIRs | SAFIRp | SAFIRsp | NVSM | word2vec | rword2vec | BM25 | BM25/RM3 |
|---|---|---|---|---|---|---|---|---|
| SAFIRs | 1.0 | 0.32 | 0.31 | 0.76 | 0.15 | 0.027 | 0.059 | 0.06 |
| SAFIRp | 0.32 | 1.0 | 0.42 | 0.26 | 0.15 | 0.091 | 0.089 | 0.077 |
| SAFIRsp | 0.31 | 0.42 | 1.0 | 0.23 | 0.095 | 0.075 | 0.075 | 0.081 |
| NVSM | 0.76 | 0.26 | 0.23 | 1.0 | 0.15 | 0.038 | 0.055 | 0.061 |
| word2vec | 0.15 | 0.15 | 0.095 | 0.15 | 1.0 | 0.48 | 0.077 | 0.096 |
| rword2vec | 0.027 | 0.091 | 0.075 | 0.038 | 0.48 | 1.0 | 0.081 | 0.088 |
| BM25 | 0.059 | 0.089 | 0.075 | 0.055 | 0.077 | 0.081 | 1.0 | 0.39 |
| BM25/RM3 | 0.06 | 0.077 | 0.081 | 0.061 | 0.096 | 0.088 | 0.39 | 1.0 |

Rank 100

| | SAFIRs | SAFIRp | SAFIRsp | NVSM | word2vec | rword2vec | BM25 | BM25/RM3 |
|---|---|---|---|---|---|---|---|---|
| SAFIRs | 1.0 | 0.42 | 0.45 | 0.82 | 0.15 | 0.14 | 0.2 | 0.22 |
| SAFIRp | 0.42 | 1.0 | 0.56 | 0.43 | 0.13 | 0.12 | 0.23 | 0.23 |
| SAFIRsp | 0.45 | 0.56 | 1.0 | 0.45 | 0.15 | 0.15 | 0.24 | 0.23 |
| NVSM | 0.82 | 0.43 | 0.45 | 1.0 | 0.15 | 0.14 | 0.21 | 0.23 |
| word2vec | 0.15 | 0.13 | 0.15 | 0.15 | 1.0 | 0.8 | 0.14 | 0.12 |
| rword2vec | 0.14 | 0.12 | 0.15 | 0.14 | 0.8 | 1.0 | 0.14 | 0.12 |
| BM25 | 0.2 | 0.23 | 0.24 | 0.21 | 0.14 | 0.14 | 1.0 | 0.45 |
| BM25/RM3 | 0.22 | 0.23 | 0.23 | 0.23 | 0.12 | 0.12 | 0.45 | 1.0 |

Rank 1000

| | SAFIRs | SAFIRp | SAFIRsp | NVSM | word2vec | rword2vec | BM25 | BM25/RM3 |
|---|---|---|---|---|---|---|---|---|
| SAFIRs | 1.0 | 0.59 | 0.58 | 0.85 | 0.28 | 0.27 | 0.35 | 0.41 |
| SAFIRp | 0.59 | 1.0 | 0.69 | 0.57 | 0.27 | 0.26 | 0.37 | 0.4 |
| SAFIRsp | 0.58 | 0.69 | 1.0 | 0.55 | 0.28 | 0.25 | 0.37 | 0.41 |
| NVSM | 0.85 | 0.57 | 0.55 | 1.0 | 0.26 | 0.25 | 0.33 | 0.39 |
| word2vec | 0.28 | 0.27 | 0.28 | 0.26 | 1.0 | 0.81 | 0.27 | 0.26 |
| rword2vec | 0.27 | 0.26 | 0.25 | 0.25 | 0.81 | 1.0 | 0.26 | 0.24 |
| BM25 | 0.35 | 0.37 | 0.37 | 0.33 | 0.27 | 0.26 | 1.0 | 0.53 |
| BM25/RM3 | 0.41 | 0.4 | 0.41 | 0.39 | 0.26 | 0.24 | 0.53 | 1.0 |

(d) CDS16

Fig. 13. Heatmaps of the mean Jaccard indices between the sets of relevant documents retrieved by each pair of models across topics at cutoffs 10, 100, and 1000, respectively, for each collection.

Thus, the results show how different models are in terms of relevant documents retrieved. In particular, SAFIR variants and NVSM significantly differ in the relevant documents retrieved at cutoff 10 in CDS collections. SAFIR$_p$ and SAFIR$_{sp}$ keep this behavior also at cutoffs 100 and 1000, whereas SAFIR$_s$ becomes similar to NVSM. This means that, even though SAFIR$_p$, SAFIR$_{sp}$, and NVSM present similar average performances at cutoffs 100 and 1000 (see Table 5), they achieve such performances by retrieving different relevant documents. On the other hand, the low similarity between SAFIR variants and BM25/RM3 – in terms of relevant documents retrieved – highlights the difference between semantic models and PRF based methods in addressing the semantic gap. This suggests that SAFIR and RM3 can be used as complementary approaches to address the semantic gap.

**Take-home message**. The integration of knowledge resources into the learning process of neural IR models is effective and helps to bridge the semantic gap between queries and documents. The learned representations encode text matching signals, necessary for IR tasks, and linguistic features to retrieve relevant documents that are most affected by the semantic gap. In particular, integrating external knowledge helps to boost the results at the top positions of the ranking list.

## 6 QUERY EXPANSION: EXPERIMENTAL RESULTS AND DISCUSSION

We present the experimental results for query expansion and we discuss them based on the research questions. Table 8 shows the performances for query expansion. We do not report the results of RM3 with QLM for the sake of simplicity. Indeed, the performances with QLM were always comparable or lower than those obtained using BM25. Also, we do not consider doc2vec-based models because of their poor performances.

### 6.1 The Impact of Polysemy and Synonymy on Query Expansion

**RQ1** Which feature between synonymy and polysemy can be exploited to reduce the semantic gap and improve retrieval?

We see that for most measures there is no statistical difference among all RM3-enhanced models. In particular, RM3-enhanced models do not present statistical differences for P@10, nDCG@10, and infNDCG in all collections. Besides, in cases where there is statistical significance, the only RM3-enhanced models that do not belong to the top group are those using word2vec and rword2vec for the first round of retrieval. Nevertheless, RM3-enhanced models based on SAFIR variants achieve the best results for most measures in CDS collections. The only exceptions are in CDS16, where the RM3-enhanced model using BM25 for both rounds of retrieval achieve better performances in Recall@1000, nDCG@100, and P@10.

Among SAFIR variants, SAFIR$_{sp}$ provides expansion terms that allow BM25 to achieve the best scores for most measures in CDS collections. This is an interesting result as it shows that modeling both synonymy and polysemy is effective to retrieve feedback documents from which expansion terms are extracted. In other words, SAFIR$_{sp}$ helps BM25 to bridge the semantic gap more effectively than the other models for CDS collections. Even when different RM3-enhanced models achieve better performances, like SAFIR$_s$/RM3 and SAFIR$_p$/RM3 in CDS14 or BM25/RM3 in CDS16, the improvements over SAFIR$_{sp}$/RM3 are small in most cases.

Given that SAFIR$_p$ outperforms SAFIR$_{sp}$ for nDCG@10 in all CDS collections (see Table 5), we provide the following explanation to motivate the higher effectiveness of SAFIR$_{sp}$/RM3 compared to SAFIR$_p$/RM3. First of all, the differences between SAFIR$_{sp}$ and SAFIR$_p$ for nDCG@10 are small. This means that SAFIR$_{sp}$ and SAFIR$_p$ have similar effectiveness in retrieving and ordering relevant documents in top positions of the ranking list. On the other hand, SAFIR$_{sp}$ and SAFIR$_p$ significantly

Table 8. RM3-enhanced models performances. RM3-enhanced models are grouped by the type of the model used in the first round of retrieval: Bag-of-Words (BoW), Corpus-Driven (CD), Knowledge-Enhanced (KE), and SAFIR. BM25 is always used for the second round of retrieval. The scores in parentheses represent the scores achieved by the model used in the first round of retrieval. **Bold** values represent the highest scores among RM3-enhanced models in each collection. † represents the models belonging to the statistical top group for the given collection with $\alpha \leq 0.05$.

| | | infNDCG | | | | nDCG@1000 | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | CDS14 | CDS15 | CDS16 | OHSUMED | CDS14 | CDS15 | CDS16 | OHSUMED |
| BoW | BM25/RM3 | 0.1384<br>(0.1064) | 0.1578<br>(0.1276) | 0.1688<br>(0.1399) | –<br>(–) | 0.2316†<br>(0.1838) | 0.2183<br>(0.1579) | 0.2068†<br>(0.1643) | 0.6253<br>(0.5875) |
| CD | word2vec/RM3 | 0.1157<br>(0.0954) | 0.1403<br>(0.1159) | 0.1292<br>(0.0928) | –<br>(–) | 0.1895<br>(0.1548) | 0.2061<br>(0.1634) | 0.1492<br>(0.1054) | 0.6507<br>(0.5902) |
| | NVSM/RM3 | 0.1673<br>(0.1576) | 0.1453<br>(0.1449) | 0.1425<br>(0.1475) | –<br>(–) | 0.2724†<br>(0.2649) | 0.2081<br>(0.2213) | 0.1941†<br>(0.1818) | 0.6511<br>(0.5977) |
| KE | rword2vec/RM3 | 0.1100<br>(0.0896) | 0.1383<br>(0.1142) | 0.1314<br>(0.0790) | –<br>(–) | 0.1836<br>(0.1501) | 0.2063<br>(0.1589) | 0.1531<br>(0.0980) | **0.6539**<br>(0.5852) |
| SAFIR | SAFIR$_{s}$/RM3 | 0.1680<br>(0.1602) | 0.1582<br>(0.1498) | 0.1490<br>(0.1546) | –<br>(–) | 0.2774†<br>(0.2546) | 0.2236<br>(0.2240) | 0.1942†<br>(0.1783) | 0.6509<br>(0.6046) |
| | SAFIR$_{p}$/RM3 | **0.1899**<br>(0.1608) | 0.1660<br>(0.1516) | 0.1463<br>(0.1523) | –<br>(–) | **0.2979**†<br>(0.2723) | 0.2276<br>(0.2247) | 0.1975†<br>(0.1858) | 0.6477<br>(0.6106) |
| | SAFIR$_{sp}$/RM3 | 0.1898<br>(0.1566) | **0.1756**<br>(0.1515) | **0.1726**<br>(0.1599) | –<br>(–) | 0.2948†<br>(0.2651) | **0.2410**<br>(0.2266) | **0.2092**†<br>(0.1849) | 0.6470<br>(0.6144) |

| | | nDCG@100 | | | | nDCG@10 | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | CDS14 | CDS15 | CDS16 | OHSUMED | CDS14 | CDS15 | CDS16 | OHSUMED |
| BoW | BM25/RM3 | 0.1338<br>(0.1098) | 0.1522<br>(0.1233) | **0.1298**†<br>(0.1078) | 0.4746<br>(0.4392) | 0.1645<br>(0.1530) | 0.1986<br>(0.2166) | 0.1518<br>(0.1606) | 0.4618<br>(0.4429) |
| CD | word2vec/RM3 | 0.1025<br>(0.0821) | 0.1302<br>(0.1064) | 0.0893<br>(0.0619) | 0.5010<br>(0.4461) | 0.1346<br>(0.1028) | 0.1705<br>(0.1435) | 0.1121<br>(0.0977) | 0.4985<br>(0.4754) |
| | NVSM/RM3 | 0.1548<br>(0.1362) | 0.1374<br>(0.1385) | 0.1076†<br>(0.1077) | 0.4956<br>(0.4181) | 0.2060<br>(0.1694) | 0.1702<br>(0.1664) | 0.1296<br>(0.1324) | **0.4989**<br>(0.3873) |
| KE | rword2vec/RM3 | 0.0970<br>(0.0774) | 0.1308<br>(0.1032) | 0.0922<br>(0.0590) | **0.5069**<br>(0.4421) | 0.1180<br>(0.0967) | 0.1742<br>(0.1410) | 0.1175<br>(0.0930) | 0.4977<br>(0.4709) |
| SAFIR | SAFIR$_{s}$/RM3 | 0.1585<br>(0.1385) | 0.1521<br>(0.1411) | 0.1106†<br>(0.1071) | 0.4962<br>(0.4216) | 0.2229<br>(0.1729) | 0.2033<br>(0.1818) | 0.1249<br>(0.1374) | 0.4902<br>(0.4121) |
| | SAFIR$_{p}$/RM3 | 0.1724<br>(0.1435) | 0.1594<br>(0.1395) | 0.1102†<br>(0.1113) | 0.4941<br>(0.4361) | 0.2185<br>(0.1931) | 0.2046<br>(0.2053) | 0.1225<br>(0.1519) | 0.4911<br>(0.4267) |
| | SAFIR$_{sp}$/RM3 | **0.1762**<br>(0.1401) | **0.1696**<br>(0.1403) | 0.1219†<br>(0.1098) | 0.4948<br>(0.4397) | **0.2253**<br>(0.1898) | **0.2407**<br>(0.1926) | **0.1572**<br>(0.1475) | 0.4856<br>(0.4380) |

| | | P@10 | | | | Recall@1000 | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | CDS14 | CDS15 | CDS16 | OHSUMED | CDS14 | CDS15 | CDS16 | OHSUMED |
| BoW | BM25/RM3 | 0.1833<br>(0.1667) | 0.2433<br>(0.2600) | **0.2067**<br>(0.2167) | 0.5413<br>(0.5016) | 0.3151†<br>(0.2503) | 0.2884<br>(0.1826) | **0.3059**†<br>(0.2286) | 0.8431<br>(0.7973) |
| CD | word2vec/RM3 | 0.1667<br>(0.1133) | 0.2233<br>(0.1900) | 0.1300<br>(0.1167) | 0.5651<br>(0.5048) | 0.2770<br>(0.2200) | 0.2795<br>(0.2194) | 0.2185<br>(0.1515) | 0.8644<br>(0.7778) |
| | NVSM/RM3 | 0.2400<br>(0.2033) | 0.2333<br>(0.2333) | 0.1767<br>(0.1600) | 0.5603<br>(0.4333) | 0.3760†<br>(0.3833) | 0.2882<br>(0.3093) | 0.2822†<br>(0.2617) | 0.8669<br>(0.8584) |
| KE | rword2vec/RM3 | 0.1500<br>(0.1267) | 0.2400<br>(0.1967) | 0.1433<br>(0.1133) | **0.5714**<br>(0.5048) | 0.2712<br>(0.2221) | 0.2816<br>(0.2151) | 0.2239<br>(0.1414) | 0.8680<br>(0.7672) |
| SAFIR | SAFIR$_{s}$/RM3 | **0.2600**<br>(0.1967) | 0.2667<br>(0.2267) | 0.1667<br>(0.1733) | 0.5587<br>(0.4619) | 0.3853†<br>(0.3607) | 0.2970<br>(0.3134) | 0.2814†<br>(0.2545) | **0.8755**<br>(0.8582) |
| | SAFIR$_{p}$/RM3 | 0.2433<br>(0.2333) | 0.2600<br>(0.2633) | 0.1600<br>(0.1700) | 0.5698<br>(0.4762) | **0.4100**†<br>(0.3846) | 0.3006<br>(0.3098) | 0.2866†<br>(0.2782) | 0.8687<br>(0.8548) |
| | SAFIR$_{sp}$/RM3 | 0.2433<br>(0.2200) | **0.3067**<br>(0.2467) | 0.1933<br>(0.1633) | 0.5571<br>(0.4794) | 0.3991†<br>(0.3733) | **0.3134**<br>(0.3110) | 0.3047†<br>(0.2747) | 0.8708<br>(0.8520) |

differ in the relevant documents retrieved at cutoff 10 (see Figure 13). In particular, the similarity score between the two models is lower than 0.50 in both CDS15 and CDS16. Thus, our intuition is that SAFIR$_{sp}$ – by modeling both polysemy and synonymy – retrieves feedback documents that provide better expansion terms than those retrieved by SAFIR$_{p}$. To support this intuition, let us consider topic 25 from CDS16. For this query, the difference between SAFIR$_{sp}$ and SAFIR$_{p}$ in terms of nDCG@10 is close to zero (see Figure 6), whereas SAFIR$_{sp}$/RM3 outperforms SAFIR$_{p}$/RM3 by a large margin ($> 0.60$) for infNDCG, nDCG@10, and P@10.

**CDS16 Topic 25:** "*An elderly female with history of atrial fibrillation, Chronic Obstructive Pulmonary Disease, hypertension, hyperlipidemia and previous repair of atrial septum defect, presenting with shortness of breath and atrial fibrillation resistant to medication.*"

The query describes a woman presenting shortness of breath and atrial fibrillation, with history of arrhythmia and other correlated diseases. In this case, SAFIR$_{sp}$ and SAFIR$_{p}$ provide six exclusive expansion terms:

SAFIR$_{sp}$: amiodarone, cardiac, dronedarone, metaprolol, procedure, rhythm
SAFIR$_{p}$: atrium, cha, chads, flutter, hypertensive, permanent

The expansion terms from SAFIR$_{sp}$ show a higher diversity than those from SAFIR$_{p}$. Three of these terms, namely "amiodarone", "dronedarone", and "metaprolol", refer to drugs or medications used to treat (and prevent) a number of types of arrhythmia – including atrial fibrillation. The other terms are highly correlated ("rhythm") and help to contextualize the anatomical region of interest ("cardiac"). On the other hand, three of the six exclusive terms provided by SAFIR$_{p}$ are terminological variants of the query terms – that is, "atrium", "flutter", and "hypertensive". As for the other terms, both "cha" and "chads" refer to clinical prediction rules used to estimate the risk of stroke in patients with atrial fibrillation. Thus, although both SAFIR variants provide highly related expansion terms, those obtained using SAFIR$_{sp}$ have more variety and help BM25 to bridge the semantic gap more effectively.

A different situation occurs in OHSUMED, where all the RM3-enhanced models show similar results. Depending on the measure, different RM3-enhanced models achieve the best results. In particular, rword2vec/RM3 provides the highest scores for nDCG@1000, nDCG@100, and P@10. In this case, SAFIR$_{s}$ and SAFIR$_{p}$ provide better expansion terms than SAFIR$_{sp}$ as both SAFIR$_{s}$/RM3 and SAFIR$_{p}$/RM3 achieve higher scores than SAFIR$_{sp}$/RM3 for most measures. Nevertheless, the only RM3-enhanced model relying on SAFIR that achieves top performances is SAFIR$_{s}$/RM3 for Recall@1000. Thus, the results strengthen our hypothesis that OHSUMED favors models relying on corpus-based features, such as rword2vec.

**Take-home message**. The effectiveness of SAFIR$_{sp}$ to provide expansion terms that help BM25 to fill the semantic gap in CDS collections shows the importance of modeling both synonymy and polysemy.

### 6.2 The Effectiveness of Knowledge Resources for Query Expansion

**RQ2** How can external knowledge resources help to bridge the semantic gap between queries and documents?

We see that SAFIR provides better expansion terms than BM25. In fact, RM3-enhanced models relying on SAFIR variants achieve higher results than BM25/RM3 for most measures in all collections. Furthermore, all the SAFIR-based RM3-enhanced models outperform NVSM/RM3 for most measures in CDS collections. Thus, the effectiveness of SAFIR variants for nDCG@10 (see Table 5), along with their exclusiveness in terms of relevant documents retrieved (see Figure 13), make them more suitable than BM25 or NVSM to perform the first round of retrieval in RM3-enhanced models.

Regarding rword2vec, we see that rword2vec/RM3 achieves performances higher than word2vec/-RM3 for many measures in all collections. However, the differences between rword2vec/RM3 and word2vec/RM3 are less prominent than those between SAFIR-based RM3-enhanced models and NVSM/RM3. Nevertheless, we advocate that knowledge-enhanced models provide better expansion terms than corpus-driven ones.

**Take-home message**. Knowledge-enhanced models grasp different signals than bag-of-words or corpus-driven models and retrieve documents in top positions that are effective in providing expansion terms for PRF models.

## 7 CONCLUSIONS

We have introduced the Semantic-Aware Neural Framework for IR (SAFIR), an unsupervised knowledge-enhanced neural framework for IR. SAFIR jointly learns word, concept, and document representations from scratch. The learned representations are optimized for IR and encode polysemy and/or synonymy with the aim of addressing the semantic gap between queries and documents. Regarding polysemy, SAFIR contextualizes word meanings by combining word and concept representations in the learning process. Thus, word meaning representations are created on-the-fly by combining word and concept representations. This compositional process avoids the creation of a representation for each word meaning. Then, word and concept representations are optimized to minimize the distance between the word meanings and the documents in the vector space. At the same time, SAFIR models synonymy via multi-task learning. Word representations are shared between text matching and word similarity tasks. For the word similarity task, SAFIR minimizes the distance between word representations for words presenting synonymy relations within an external knowledge resource.

For evaluation, we considered three variants of SAFIR: SAFIR$_{sp}$, which integrates both synonymy and polysemy; SAFIR$_{s}$ which integrates synonymy but not polysemy; and SAFIR$_{p}$ which integrates polysemy but not synonymy. We compared SAFIR variants with other knowledge-enhanced neural models on medical literature retrieval considering two strategies: document retrieval and query expansion. The experimental results we obtained led to the following conclusions:

**RQ1** Which feature between synonymy and polysemy can be exploited to reduce the semantic gap and improve retrieval?

**Document Retrieval:** modeling polysemy is effective and impacts the most when queries present a high degree of polysemy. On the other hand, the impact of synonymy on average performances is marginal – or even detrimental – due to the limited presence of relevant documents containing (only) query synonyms. Nevertheless, when we look at queries with a large number of relevant documents containing (only) query synonyms, SAFIR$_{s}$ and SAFIR$_{sp}$ capture synonymy and provide effective results.

**Query Expansion:** the effectiveness of SAFIR$_{sp}$ to provide expansion terms that help BM25 to fill the semantic gap in CDS collections shows the importance of modeling both synonymy and polysemy.

**RQ2** How can external knowledge resources help to bridge the semantic gap between queries and documents?

**Document Retrieval:** the integration of knowledge resources into the learning process of neural IR models is effective and helps to bridge the semantic gap between queries and documents. The learned representations encode text matching signals, necessary for IR tasks, and linguistic features to retrieve relevant documents that are most affected by the semantic gap. In particular, integrating external knowledge helps to boost the results at the top positions of the ranking list.

**Query Expansion:** knowledge-enhanced models grasp different signals than bag-of-words or corpus-driven models and retrieve documents in top positions that are effective in providing expansion terms for Pseudo Relevance Feedback (PRF) models.

The evaluation showed that SAFIR retrieves more exclusive relevant documents than knowledge-enhanced language models for most queries in all collections. Furthermore, the effectiveness of SAFIR for Precision-oriented measures, along with its exclusiveness in terms of relevant documents retrieved, makes it suitable for PRF models. Therefore, our evaluation suggests that unsupervised knowledge-enhanced models should be used at the early stages of the IR pipeline rather than in re-ranking scenarios – where interaction-based re-ranking models can easily outperform them. In this way, the different signals that knowledge-enhanced models provide can be used by multi-stage IR systems to obtain a richer pool of relevant documents, thus leading to better answers for semantically hard queries.

As future work, we plan to integrate deeper neural architectures into SAFIR representation learning component to better model linguistic features and their interactions with IR-oriented objective functions. Specifically, we want to investigate how attention layers from Transformer networks [71] can be employed to model the *term specificity* [62] of word meanings. Two other interesting directions we plan to investigate are the extension of SAFIR to phrase-concept associations and the analysis of the sensitivity of the learned representations to the errors introduced by Named Entity Recognition (NER) and Entity Linking (EL) components.

## ACKNOWLEDGMENTS

The work was partially supported by the ExaMode project, as part of the European Union H2020 program under Grant Agreement no. 825292. We also sincerely thank the associate editor and the anonymous reviewers whose suggestions helped in improving this paper.

## REFERENCES

[1] Q. Ai, L. Yang, J. Guo, and W. B. Croft. 2016. Analysis of the Paragraph Vector Model for Information Retrieval. In *Proc. of the 2016 ACM International Conference on the Theory of Information Retrieval, ICTIR 2016.* ACM, 133–142.

[2] Q. Ai, L. Yang, J. Guo, and W. B. Croft. 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, 869–872.

[3] S. Balaneshinkordan and A. Kotov. 2016. Optimization Method for Weighting Explicit and Latent Concepts in Clinical Decision Support Queries. In *Proc. of the 2016 ACM on International Conference on the Theory of Information Retrieval, ICTIR 2016.* ACM.

[4] S. Balaneshinkordan and A. Kotov. 2019. Bayesian approach to incorporating different types of biomedical knowledge bases into information retrieval systems for clinical decision support in precision medicine. *J. Biomed. Informatics* (2019).

[5] S. Balaneshinkordan, A. Kotov, and R. Xisto. 2015. WSU-IR at TREC 2015 Clinical Decision Support Track: Joint Weighting of Explicit and Latent Medical Query Concepts from Diverse Sources. In *Proc. of The Twenty-Fourth Text REtrieval Conference, TREC 2015.* NIST.

[6] Y. Bengio, R. Ducharme, P. Vincent, and C. Jauvin. 2003. A Neural Probabilistic Language Model. *J. of Mach. Learn. Res.* 3 (2003), 1137–1155.

[7] O. Bodenreider. 2004. The Unified Medical Language System (UMLS): integrating biomedical terminology. *Nucleic acids research* 32, suppl_1 (2004), D267–D270.

[8] A. Booth and A. O'Rourke. 1997. The value of structured abstracts in information retrieval from MEDLINE. *Health Libraries Review* 14, 3 (1997), 157–166.

[9] B. A. Carterette. 2012. Multiple Testing in Statistical Analysis of Systems-Based Information Retrieval Experiments. *ACM Trans. Inf. Syst.* 30, 1 (2012), 4:1–4:34.

[10] M. Chen. 2017. Efficient Vector Representation for Documents through Corruption. In *Proc. of the 5th International Conference on Learning Representations, ICLR 2017.* OpenReview.net.

[11] J. Cheng, Z. Wang, J. R. Wen, J. Yan, and Z. Chen. 2015. Contextual Text Understanding in Distributional Semantic Space. In *Proc. of the 24th ACM International Conference on Information and Knowledge Management, CIKM 2015.* ACM, 133–142.

[12] E. Choi, M. T. Bahadori, E. Searles, C. Coffey, M. Thompson, J. Bost, J. Tejedor-Sojo, and J. Sun. 2016. Multi-layer Representation Learning for Medical Concepts. In *Proc. of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.* ACM, 1495–1504.

[13] T. M. Cover and J. A. Thomas. 2012. *Elements of Information Theory.* John Wiley & Sons.

[14] N. Craswell, B. Mitra, E. Yilmaz, D. Campos, and E. M. Voorhees. [n.d.]. Overview of the TREC 2019 deep learning track. *CoRR* abs/2003.07820.

[15] L. De Vine, G. Zuccon, B. Koopman, L. Sitbon, and P. Bruza. 2014. Medical Semantic Similarity with a Neural Language Model. In *Proc. of the 23rd ACM International Conference on Information and Knowledge Management, CIKM 2014.* ACM, 1819–1822.

[16] J. Devlin, M. W. Chang, K. Lee, and K. Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proc. of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019.* ACL, 4171–4186.

[17] F. Diaz, B. Mitra, and N. Craswell. 2016. Query Expansion with Locally-Trained Word Embeddings. In *Proc. of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016.* ACL.

[18] T. Edinger, A. M. Cohen, S. Bedrick, K. H. Ambert, and W. R. Hersh. 2012. Barriers to Retrieving Patient Information from Electronic Health Record Data: Failure Analysis from the TREC Medical Records Track. In *AMIA 2012, American Medical Informatics Association Annual Symposium.* AMIA.

[19] M. Faruqui, J. Dodge, S. K. Jauhar, C. Dyer, E. Hovy, and N. A. Smith. 2015. Retrofitting Word Vectors to Semantic Lexicons. In *Proc. of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies.* ACL, 1606–1615.

[20] N. Fuhr. 2017. Some Common Mistakes In IR Evaluation, And How They Can Be Avoided. *SIGIR Forum* 51, 3, 32–41.

[21] D. Ganguly, D. Roy, M. Mitra, and G. J. F. Jones. 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.* ACM, 795–798.

[22] G. Glavaš and I. Vulić. 2018. Explicit Retrofitting of Distributional Word Vectors. In *Proc. of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018.* ACL, 34–45.

[23] Ç. Gülçehre, M. Moczulski, M. Denil, and Y. Bengio. 2016. Noisy Activation Functions. In *Proc. of the 33nd International Conference on Machine Learning, ICML 2016.* JMLR.org, 3059–3068.

[24] J. Guo, Y. Fan, Q. Ai, and W. B. Croft. 2016. Semantic Matching by Non-Linear Word Transportation for Information Retrieval. In *Proc. of the 25th ACM International Conference on Information and Knowledge Management, CIKM 2016.* ACM, 701–710.

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

[26] Z. S. Harris. 1954. Distributional structure. *Word* 10, 2-3 (1954), 146–162.

[27] W. R. Hersh, C. Buckley, T. J. Leone, and D. Hickam. 1994. OHSUMED: An Interactive Retrieval Evaluation and New Large Test Collection for Research. In *Proc. of the 17th Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval.* ACM, 192–201.

[28] I. Iacobacci, M. T. Pilehvar, and R. Navigli. 2015. SensEmbed: Learning Sense Embeddings for Word and Relational Similarity. In *Proc. of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, ACL 2015.* ACL, 95–105.

[29] S. Ioffe and C. Szegedy. 2015. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. In *Proc. of the 32nd International Conference on Machine Learning, ICML 2015.* JMLR.org, 448–456.

[30] N. A. Jaleel, J. Allan, W. B. Croft, F. Diaz, L. S. Larkey, X. Li, M. D. Smucker, and C. Wade. 2004. UMass at TREC 2004: Novelty and HARD. In *Proc. of the Thirteenth Text REtrieval Conference, TREC 2004.* NIST.

[31] R. Johansson and L. N. Piña. 2015. Embedding a Semantic Network in a Word Space. In *Proc. of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies.* ACL, 1428–1433.

[32] T. Kenter, A. Borisov, and M. de Rijke. 2016. Siamese CBOW: Optimizing Word Embeddings for Sentence Representations. In *Proc. of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016.* ACL, 941–951.

[33] D. P. Kingma and J. Ba. 2015. Adam: A Method for Stochastic Optimization. In *Proc. of the 3rd International Conference on Learning Representations, ICLR 2015*.

[34] R. Kiros, Y. Zhu, R. R. Salakhutdinov, R. Zemel, R. Urtasun, A. Torralba, and S. Fidler. 2015. Skip-Thought Vectors. In *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015*. 3294–3302.

[35] B. Koopman and G. Zuccon. 2014. Why Assessing Relevance in Medical IR is Demanding. In *Proc. of the Medical Information Retrieval Workshop at SIGIR co-located with the 37th annual international ACM SIGIR conference (ACM SIGIR 2014) (CEUR Workshop Proceedings)*, Vol. 1276. CEUR-WS.org, 16–19.

[36] B. Koopman, G. Zuccon, P. Bruza, L. Sitbon, and M. Lawley. 2016. Information retrieval as semantic inference: a Graph Inference model applied to medical search. *Inf. Retr. Journal* 19, 1-2 (2016), 6–37.

[37] S. Kuzi, A. Shtok, and O. Kurland. 2016. Query Expansion Using Word Embeddings. In *Proc. of the 25th ACM International Conference on Information and Knowledge Management, CIKM 2016*. ACM, 1929–1932.

[38] V. Lavrenko and W. B. Croft. 2001. Relevance-Based Language Models. In *Proc. of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2001*. ACM, 120–127.

[39] Q. Le and T. Mikolov. 2014. Distributed Representations of Sentences and Documents. In *Proc. of the 31th International Conference on Machine Learning, ICML 2014*. JMLR.org, 1188–1196.

[40] O. Levy and Y. Goldberg. 2014. Neural Word Embedding as Implicit Matrix Factorization. In *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014*. 2177–2185.

[41] N. Limsopatham, C. Macdonald, and I. Ounis. 2013. Inferring Conceptual Relationships to Improve Medical Records Search. In *Open research Areas in Information Retrieval, OAIR '13*. ACM, 1–8.

[42] X. Liu, J. Y. Nie, and A. Sordoni. 2016. Constraining Word Embeddings by Prior Knowledge - Application to Medical Information Retrieval. In *Proc. of the 12th Asia Information Retrieval Societies Conference, AIRS 2016*. Springer, 155–167.

[43] M. Mancini, J. Camacho-Collados, I. Iacobacci, and R. Navigli. 2017. Embedding Words and Senses Together via Joint Knowledge-Enhanced Training. In *Proc. of the 21st Conference on Computational Natural Language Learning (CoNLL 2017)*. ACL, 100–111.

[44] O. Melamud, J. Goldberger, and I. Dagan. 2016. context2vec: Learning Generic Context Embedding with Bidirectional LSTM. In *Proc. of the 20th Conference on Computational Natural Language Learning, CoNLL 2016*. ACL, 51–61.

[45] T. Mikolov, K. Chen, G. Corrado, and J. Dean. 2013. Efficient estimation of word representations in vector space. *CoRR* abs/1301.3781 (2013). arXiv:1301.3781

[46] N. Mrkšic, D. OSéaghdha, B. Thomson, M. Gašic, L. Rojas-Barahona, P. H. Su, D. Vandyke, T. H. Wen, and S. Young. 2016. Counter-fitting Word Vectors to Linguistic Constraints. In *Proc. of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. ACL, 142–148.

[47] G. H. Nguyen, L. Tamine, L. Soulier, and N. Souf. 2017. Learning Concept-Driven Document Embeddings for Medical Information Search. In *Proc. of the 16th Conference on Artificial Intelligence in Medicine, AIME 2017*. Springer, 160–170.

[48] G. H. Nguyen, L. Tamine, L. Soulier, and N. Souf. 2018. A Tri-Partite Neural Document Language Model for Semantic Information Retrieval. In *Proc. of the 15th European Semantic Web Conference, ESWC 2018*. Springer, 445–461.

[49] R. Nogueira and K. Cho. 2019. Passage Re-ranking with BERT. *CoRR* abs/1901.04085 (2019). arXiv:1901.04085

[50] J. Pennington, R. Socher, and C. Manning. 2014. Glove: Global Vectors for Word Representation. In *Proc. of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014*. ACL, 1532–1543.

[51] M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer. 2018. Deep Contextualized Word Representations. In *Proc. of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2018*. ACL, 2227–2237.

[52] R. Řehůřek and P. Sojka. 2010. Software Framework for Topic Modelling with Large Corpora. In *Proc. of the LREC 2010 Workshop on New Challenges for NLP Frameworks*. ELRA, 45–50.

[53] K. Roberts, D. Demner-Fushman, E. M. Voorhees, and W. R. Hersh. 2016. Overview of the TREC 2016 Clinical Decision Support Track. In *Proc. of The Twenty-Fifth Text REtrieval Conference, TREC 2016*. NIST.

[54] K. Roberts, M. Simpson, D. Demner-Fushman, E. Voorhees, and W. Hersh. 2016. State-of-the-art in biomedical literature retrieval for clinical cases: a survey of the TREC 2014 CDS track. *Inf. Retr. Journal* 19, 1-2 (2016), 113–148.

[55] K. Roberts, M. S. Simpson, E. M. Voorhees, and W. R. Hersh. 2015. Overview of the TREC 2015 Clinical Decision Support Track. In *Proc. of The Twenty-Fourth Text REtrieval Conference, TREC 2015*. NIST.

[56] S. E. Robertson. 2004. Understanding inverse document frequency: on theoretical arguments for IDF. *J. of Documentation* 60, 5 (2004), 503–520.

[57] S. E. Robertson and H. Zaragoza. 2009. The Probabilistic Relevance Framework: BM25 and Beyond. *Found. Trends Inf. Retr.* 3, 4 (2009), 333–389.

[58] N. H. Shah, N. Bhatia, C. Jonquet, D. L. Rubin, A. P. Chiang, and M. A. Musen. 2009. Comparison of concept recognizers for building the Open Biomedical Annotator. *BMC Bioinformatics* 10, S-9 (2009), 14.

[59] R. A. Sinoara, J. Camacho-Collados, R. G. Rossi, R. Navigli, and S. O. Rezende. 2019. Knowledge-Enhanced Document Embeddings for Text Classification. *Knowl.-Based Syst.* 163 (2019), 955–971.

[60] L. Soldaini and N. Goharian. 2016. Quickumls: a fast, unsupervised approach for medical concept extraction. In *MedIR Workshop, SIGIR 2016*.

[61] A. Sordoni, Y. Bengio, and J. Y. Nie. 2014. Learning Concept Embeddings for Query Expansion by Quantum Entropy Minimization. In *Proc. of the Twenty-Eighth AAAI Conference on Artificial Intelligence*. AAAI Press, 1586–1592.

[62] K. Spärck Jones. 1972. A statistical interpretation of term specificity and its application in retrieval. *J. of Documentation* 28, 1 (1972), 11–21.

[63] T. Strohman, D. Metzler, H. Turtle, and W. B. Croft. 2005. Indri: A language model-based search engine for complex queries. In *Proc. of the International Conference on Intelligent Analysis*. Citeseer, 2–6.

[64] Y. Sun, S. Wang, Y. Li, S. Feng, X. Chen, H. Zhang, X. Tian, D. Zhu, H. Tian, and H. Wu. 2019. ERNIE: Enhanced Representation through Knowledge Integration. *CoRR* abs/1904.09223 (2019). arXiv:1904.09223

[65] J. Tague-Sutcliffe. 1992. The Pragmatics of Information Retrieval Experimentation Revisited. *Inf. Proc. Manage.* 28, 4 (1992), 467–490.

[66] L. Tamine, L. Soulier, G. H. Nguyen, and N. Souf. 2019. Offline Versus Online Representation Learning of Documents Using External Knowledge. *ACM Trans. Inf. Syst.* 37, 4 (2019), 42:1–42:34.

[67] B. Thompson. 2006. *Foundations of Behavioral Statistics: An Insight-Based Approach*. Guilford Press.

[68] C. Van Gysel, M. de Rijke, and E. Kanoulas. 2016. Learning Latent Vector Spaces for Product Search. In *Proc. of the 25th ACM International Conference on Information and Knowledge Management, CIKM 2016*. ACM, 165–174.

[69] C. Van Gysel, M. de Rijke, and E. Kanoulas. 2018. Neural Vector Spaces for Unsupervised Information Retrieval. *ACM Trans. Inf. Syst.* 36, 4 (2018), 38:1–38:25.

[70] C. Van Gysel, M. de Rijke, and M. Worring. 2016. Unsupervised, Efficient and Semantic Expertise Retrieval. In *Proc. of the 25th International Conference on World Wide Web, WWW 2016*. ACM, 1069–1079.

[71] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. 2017. Attention is All you Need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017*. 5998–6008.

[72] E. M. Voorhees. 2001. Evaluation by Highly Relevant Documents. In *Proc. of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, 74–82.

[73] E. M. Voorhees and R. M. Tong. 2011. Overview of the TREC 2011 Medical Records Track. In *Proc. of The Twentieth Text REtrieval Conference, TREC 2011*. NIST.

[74] I. Vulić and M. F. Moens. 2015. Monolingual and Cross-Lingual Information Retrieval Models Based on (Bilingual) Word Embeddings. In *Proc. of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, 363–372.

[75] C. Xu, Y. Bai, J. Bian, B. Gao, G. Wang, X. Liu, and T. Y. Liu. 2014. RC-NET: A General Framework for Incorporating Knowledge into Word Representations. In *Proc. of the 23rd ACM International Conference on Information and Knowledge Management, CIKM 2014*. ACM, 1219–1228.

[76] I. Yamada, H. Shindo, H. Takeda, and Y. Takefuji. 2016. Joint Learning of the Embedding of Words and Entities for Named Entity Disambiguation. In *Proc. of the 20th Conference on Computational Natural Language Learning, CoNLL 2016*. ACL, 250–259.

[77] E. Yilmaz, E. Kanoulas, and J. A. Aslam. 2008. A simple and efficient sampling method for estimating AP and NDCG. In *Proc. of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2008*. ACM, 603–610.

[78] M. Yu and M. Dredze. 2014. Improving Lexical Embeddings with Semantic Knowledge. In *Proc. of the 52nd Annual Meeting of the Association for Computational Linguistics, ACL 2014*. ACL, 545–550.

[79] H. Zamani and W. B. Croft. 2016. Embedding-based Query Language Models. In *Proc. of the 2016 ACM International Conference on the Theory of Information Retrieval, ICTIR 2016*. ACM, 147–156.

[80] H. Zamani and W. B. Croft. 2016. Estimating Embedding Vectors for Queries. In *Proc. of the 2016 ACM International Conference on the Theory of Information Retrieval, ICTIR 2016*. ACM, 123–132.

[81] C. Zhai. 2008. Statistical Language Models for Information Retrieval: A Critical Review. *Found. Trends Inf. Retr.* 2, 3 (2008), 137–213.

[82] Z. Zhang, X. Han, Z. Liu, X. Jiang, M. Sun, and Q. Liu. 2019. ERNIE: Enhanced Language Representation with Informative Entities. In *Proc. of the 57th Conference of the Association for Computational Linguistics, ACL 2019*. ACL, 1441–1451.

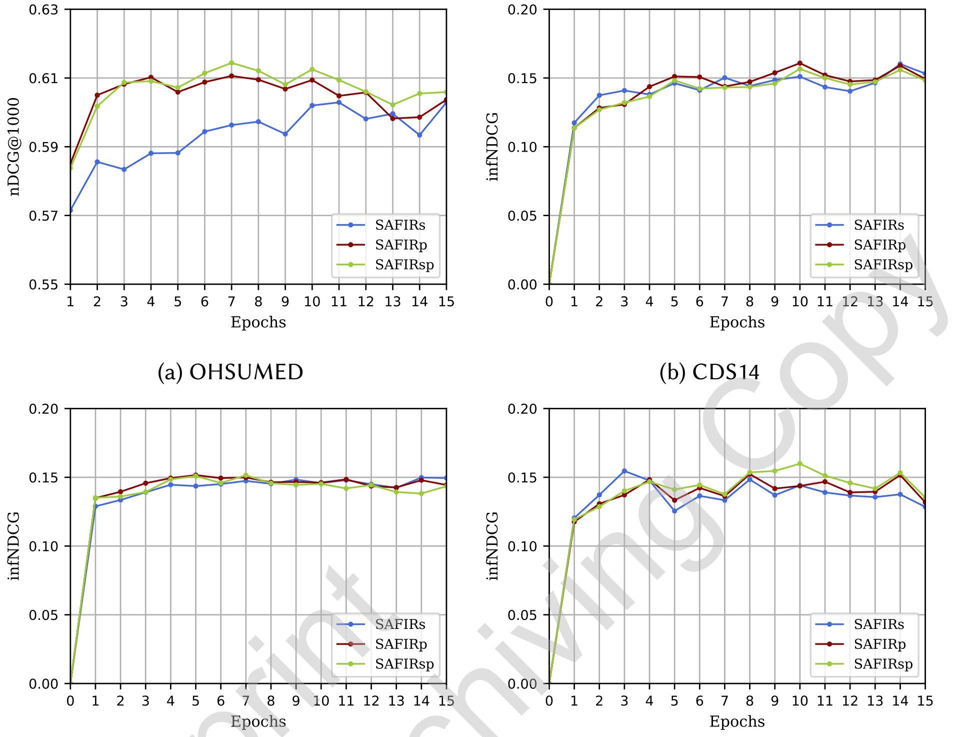
[83] R. Zhao and W. I. Grosky. 2002. Narrowing the semantic gap - improved text-based web document retrieval using visual features. *IEEE Trans. Multimedia* 4, 2 (2002), 189–200.

[84] G. Zuccon, B. Koopman, P. Bruza, and L. Azzopardi. 2015. Integrating and Evaluating Neural Word Embeddings in Information Retrieval. In *Proc. of the 20th Australasian Document Computing Symposium, ADCS 2015*. ACM, 12:1–12:8.

# ELECTRONIC APPENDIX

## A RETRIEVAL PERFORMANCES AVERAGED OVER ITERATIONS

The behavior of SAFIR variants in terms of optimization as training progresses is shown in Figure 14.



(a) OHSUMED

(b) CDS14

(c) CDS15

(d) CDS16

Fig. 14. nDCG@1000/infNDCG scores as training of SAFIR variants progresses on each collection. For OHSUMED, we present the optimization results from epoch 1 to ease visualization.

The curves show that SAFIR variants improve up to a certain value and then start to oscillate across iterations – although these oscillations tend to be small in most cases. Therefore, we investigate how the performances change when we consider the average over iterations 10-15 instead of the best iteration. To this end, we compare SAFIR variants with NVSM – averaged over iterations 10-15 – and BM25/RM3. The results are reported in Table 9

The results from Table 9 show that the performances obtained by SAFIR variants averaged over iterations 10-15 are similar – although often lower – to those obtained with the best iterations (cf. Table 5). The most notable exceptions are nDCG@10 and P@10, where the differences between averaged and best performances are larger. However, Precision-oriented measures are highly sensitive to performance variations. Therefore, the different representations used at each iteration to perform retrieval can have a large impact on the results for these measures – especially when the considered representations are far from being optimal.

Overall, the models that perform best are consistent between averaged and best iterations. In particular, the average of SAFIR over iterations 10-15 achieves top performances in most of the measures in which also the best iteration of SAFIR achieves them. The only exceptions are

Table 9. Retrieval performances of considered models averaged over epochs 10-15. Models are grouped by type: Bag-of-Words (BoW), Corpus-Driven (CD), and SAFIR. The values in parentheses represent the standard deviation values. **Bold** values represent the highest scores among the models in each collection.

| | | infNDCG | | | | nDCG@1000 | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | CDS14 | CDS15 | CDS16 | OHSUMED | CDS14 | CDS15 | CDS16 | OHSUMED |
| BoW | BM25/RM3 | 0.1384<br>(0.0000) | **0.1578**<br>(0.0000) | **0.1688**<br>(0.0000) | –<br>(–) | 0.2316<br>(0.0000) | 0.2183<br>(0.0000) | **0.2068**<br>(0.0000) | **0.6253**<br>(0.0000) |
| CD | NVSM | 0.1507<br>(0.0061) | 0.1436<br>(0.0012) | 0.1367<br>(0.0045) | –<br>(–) | 0.2658<br>(0.0030) | 0.2168<br>(0.0061) | 0.1839<br>(0.0037) | 0.5947<br>(0.0021) |
| SAFIR | SAFIRs | 0.1491<br>(0.0066) | 0.1467<br>(0.0026) | 0.1369<br>(0.0046) | –<br>(–) | 0.2559<br>(0.0031) | 0.2222<br>(0.0021) | 0.1778<br>(0.0039) | 0.6002<br>(0.0040) |
| | SAFIRp | **0.1529**<br>(0.0052) | 0.1455<br>(0.0022) | 0.1421<br>(0.0063) | –<br>(–) | **0.2696**<br>(0.0032) | **0.2242**<br>(0.0020) | 0.1882<br>(0.0025) | 0.6034<br>(0.0040) |
| | SAFIRsp | 0.1506<br>(0.0043) | 0.1421<br>(0.0026) | 0.1479<br>(0.0080) | –<br>(–) | 0.2652<br>(0.0032) | 0.2195<br>(0.0019) | 0.1832<br>(0.0025) | 0.6069<br>(0.0033) |

| | | nDCG@100 | | | | nDCG@10 | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | CDS14 | CDS15 | CDS16 | OHSUMED | CDS14 | CDS15 | CDS16 | OHSUMED |
| BoW | BM25/RM3 | 0.1338<br>(0.0000) | **0.1522**<br>(0.0000) | **0.1298**<br>(0.0000) | **0.4746**<br>(0.0000) | 0.1645<br>(0.0000) | **0.1986**<br>(0.0000) | **0.1518**<br>(0.0000) | **0.4618**<br>(0.0000) |
| CD | NVSM | 0.1347<br>(0.0042) | 0.1348<br>(0.0024) | 0.1033<br>(0.0040) | 0.4140<br>(0.0028) | 0.1611<br>(0.0147) | 0.1636<br>(0.0030) | 0.1236<br>(0.0094) | 0.3793<br>(0.0062) |
| SAFIR | SAFIRs | 0.1340<br>(0.0040) | 0.1386<br>(0.0026) | 0.1029<br>(0.0028) | 0.4196<br>(0.0055) | 0.1639<br>(0.0108) | 0.1726<br>(0.0098) | 0.1218<br>(0.0112) | 0.4040<br>(0.0068) |
| | SAFIRp | **0.1420**<br>(0.0040) | 0.1378<br>(0.0017) | 0.1077<br>(0.0042) | 0.4254<br>(0.0049) | 0.1769<br>(0.0121) | 0.1821<br>(0.0024) | 0.1450<br>(0.0146) | 0.4038<br>(0.0042) |
| | SAFIRsp | 0.1412<br>(0.0031) | 0.1365<br>(0.0020) | 0.1083<br>(0.0034) | 0.4308<br>(0.0042) | **0.1869**<br>(0.0109) | 0.1694<br>(0.0046) | 0.1493<br>(0.0121) | 0.4183<br>(0.0088) |

| | | P@10 | | | | Recall@1000 | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | CDS14 | CDS15 | CDS16 | OHSUMED | CDS14 | CDS15 | CDS16 | OHSUMED |
| BoW | BM25/RM3 | 0.1833<br>(0.0000) | **0.2433**<br>(0.0000) | **0.2067**<br>(0.0000) | **0.5413**<br>(0.0000) | 0.3151<br>(0.0000) | 0.2884<br>(0.0000) | **0.3059**<br>(0.0000) | 0.8431<br>(0.0000) |
| CD | NVSM | 0.1933<br>(0.0176) | 0.2274<br>(0.0063) | 0.1439<br>(0.0068) | 0.4376<br>(0.0079) | **0.3903**<br>(0.0036) | 0.3073<br>(0.0120) | 0.2765<br>(0.0032) | **0.8582**<br>(0.0016) |
| SAFIR | SAFIRs | 0.1922<br>(0.0101) | 0.2272<br>(0.0073) | 0.1328<br>(0.0139) | 0.4598<br>(0.0066) | 0.3703<br>(0.0052) | **0.3123**<br>(0.0024) | 0.2661<br>(0.0060) | 0.8564<br>(0.0018) |
| | SAFIRp | 0.2161<br>(0.0124) | 0.2350<br>(0.0069) | 0.1644<br>(0.0133) | 0.4659<br>(0.0067) | 0.3839<br>(0.0032) | 0.3107<br>(0.0047) | 0.2794<br>(0.0051) | 0.8564<br>(0.0030) |
| | SAFIRsp | **0.2195**<br>(0.0097) | 0.2211<br>(0.0097) | 0.1656<br>(0.0170) | 0.4733<br>(0.0120) | 0.3723<br>(0.0030) | 0.3053<br>(0.0031) | 0.2697<br>(0.0026) | 0.8504<br>(0.0019) |

P@10 in CDS15 and Recall@1000 in CDS14, where BM25/RM3 and NVSM achieve the best results, respectively. However, the SAFIR variants that achieve top results are different for some measures in CDS collections. SAFIR$_{sp}$ achieves top performances instead of SAFIR$_{p}$ for nDCG@10 and P@10 in CDS14, whereas SAFIR$_{p}$ replaces SAFIR$_{sp}$ for nDCG@1000 in CDS15. On the other hand, the ranking of the models for OHSUMED does not change regardless of the approach selected – be it the average of the iterations 10-15 or the best iteration. The only exception is for nDCG@10, where SAFIR$_{s}$ outperforms SAFIR$_{p}$.

To understand to what extent the rankings of considered models change when we take the average of iterations 10-15 instead of the best iteration, we perform Kendall's $\tau$ correlations between model rankings obtained in one way or the other. Table 10 reports correlation values.

Table 10. Kendall $\tau$ correlations computed between the rankings of the considered models from Table 9 (average of iterations 10-15) and Table 5 (best iteration) for each measure in each collection.

| | infNDCG | nDCG@1000 | nDCG@100 | nDCG@10 | P@10 | Recall@1000 |
|---|---|---|---|---|---|---|
| OHSUMED | – | 1.00 | 1.00 | 0.80 | 1.00 | 1.00 |
| CDS14 | 0.60 | 0.80 | 0.80 | 0.40 | 0.80 | 0.80 |
| CDS15 | 0.40 | 0.40 | 0.80 | 0.60 | 0.20 | 0.60 |
| CDS16 | 0.80 | 0.80 | 0.80 | 0.40 | 0.20 | 0.80 |

Table 10 shows that in more than 60% of cases correlation values are greater than or equal to 0.80 – which indicates that the differences between rankings do not reflect noticeable changes [72]. The rest of the correlation values divides among 0.60 (13% of cases), 0.40 (17% of cases), and 0.20 (9% of cases). Given the short length of the considered ranking lists (only five elements), a correlation value of 0.80 means that the two rankings differ by a single swap between positions. As a result, correlation values of 0.60 occur with two swaps between positions of the ranking list, whereas scores of 0.40 and 0.20 with three and four swaps, respectively. Below, we focus on low correlation values – i.e., 0.40 and 0.20 – and we detail the differences between the ranking lists obtained taking the average of iterations 10-15 and the best iteration.

As expected, low correlations cluster on Precision-oriented measures. nDCG@10 presents correlation values of 0.40 in CDS14 and CDS16. In CDS14, SAFIR$_{sp}$ outperforms SAFIR$_{p}$ and becomes the top performing model, whereas BM25/RM3 moves from last to the third position. In CDS16, BM25/RM3 and SAFIR$_{sp}$ outperform SAFIR$_{p}$ achieving the first and second positions, respectively. On the other hand, SAFIR$_{s}$ moves from the fourth position to the last. As for P@10, CDS15 and CDS16 show correlation values of 0.20. In CDS15, BM25/RM3 moves from the third to the top position. Also, SAFIR$_{s}$ outperforms SAFIR$_{sp}$ – which becomes the worst performing model. In CDS16, SAFIR$_{sp}$ outperforms both SAFIR$_{s}$ and SAFIR$_{p}$ achieving the second position, whereas SAFIR$_{s}$ moves to the last position.

Other than nDCG@10 and P@10, CDS15 exhibits low correlations (0.40) also for infNDCG and nDCG@1000. For infNDCG, SAFIR$_{s}$ gains two positions (from fourth to second) and SAFIR$_{sp}$ becomes the worst performing model. For nDCG@1000, SAFIR$_{p}$ and SAFIR$_{s}$ outperform SAFIR$_{sp}$ achieving the first and second positions, respectively. Moreover, BM25/RM3 outperforms NVSM and moves from the last to the fourth position.

Thus, the results of this analysis show that the performances obtained by SAFIR variants averaged over iterations 10-15 are similar – although often lower – to those obtained with the best iterations. Furthermore, the average of SAFIR over iterations 10-15 achieves top performances in most of the measures in which also the best iteration of SAFIR achieves them. However, the SAFIR variants that achieve top results are different for some measures. Finally, the rankings obtained when we take the average of iterations 10-15 present high correlation values with the rankings obtained considering the best iteration in most cases. As expected, most of the low correlations occur for Precision-oriented measures – which are highly sensitive to variations in models performance.