Complex Systems Informatics and Modeling Quarterly (CSIMQ)
eISSN: 2255-9922
Published online by RTU Press, https://csimq-journals.rtu.lv
Article 227, Issue 41, December 2024/January 2025, Pages 80–111
https://doi.org/10.7250/csimq.2024-41.05

Text Retrieval in Restricted Domains by Pairwise Term Co-occurrence

Eriks Sneiders* and Aron Henriksson

Department of Computer and Systems Sciences, Stockholm University, Postbox 7003, SE-164 07 Kista, Sweden

eriks@dsv.su.se, aronhen@dsv.su.se

Abstract. Text similarity calculation by text embeddings requires fine-tuning of the language model by a large amount of labeled data, which may not be available for small text collections in their specific knowledge domains, in particular, in public organizations. As an alternative to machine learning, this research proposes pairwise term co-occurrence within plain-text matching, i.e., the query and the document share co-occurrences of two terms in a text span. In the entire document, the co-occurrences form the context that affects a term. This is analogous to a contextual word embedding, except our context affects the importance, not the meaning, of the term. Pairwise term co-occurrence has been applied in three text similarity calculation methods: term-pair-based text similarity, BM25 with term weights enhanced by pairwise term co-occurrence, and likewise enhanced cosine similarity. The three methods were evaluated for retrieval of four text types – email messages, web articles, fill-in forms, and brochures from a public organization – by having the first three as queries. Pairwise term co-occurrence performed on par with or better than BERT sentence embeddings without fine-tuning the BERT language model. With some text types, pairwise term co-occurrence outperformed bag-of-words matching by as much as 29.44 (MAP) and 31.71 (P@1) percentage points. Pairwise term co-occurrence can fill a niche by improving text similarity calculation where supervised machine learning is difficult to carry out.

Keywords: Term Co-occurrence, Text Similarity, Text Matching, Term Weights, Document Retrieval, BM25, Embeddings.

1 Introduction

Document retrieval based on bag-of-words matching has dominated Information Retrieval (IR) for decades. A paradigm shift came along with large, public, Internet-based text collections which enabled the introduction of deep learning into Natural Language Processing (NLP) and the

* Corresponding author

Reference: E. Sneiders and A. Henriksson, "Text Retrieval in Restricted Domains by Pairwise Term Co-occurrence," Complex Systems Informatics and Modeling Quarterly, CSIMQ, no. 41, pp. 80–111, 2024. Available: https://doi.org/10.7250/csimq.2024-41.05

Additional information. Author ORCID iD: E. Sneiders – https://orcid.org/0000-0002-2803-5139, and A. Henriksson – https://orcid.org/0000-0001-9731-1048. PII S225599222400227X. Received: 6 December 2024. Accepted: 28 December 2024. Available 31 December 2024.

development of word embeddings [1], [2]. Generative AI [3], large language models (LLMs) [4], and text embeddings are among the latest achievements in NLP. Today, the research in textual IR revolves largely around the combination of Machine Learning (ML) and NLP for solving IR tasks such as question answering [5], [6], text retrieval [7] and categorization [8], and spam filtering [9]. There exist off-the-shelf pre-trained general language models, notably those available on HuggingFace [10], that can be further fine-tuned for a particular text corpus and task. Improving and fine-tuning LLMs is the state of the art in IR research.

Not all IR use cases, for instance, many public organizations are able to or are allowed to follow the latest trends. It is not likely that, in the near future, courts will rely on computer-generated law, although machine learning can help in identifying relevant legal texts [11]. Public organizations communicate to citizens mostly pre-approved texts because a public organization that exercises authority must not make statements that contradict laws and regulations, even by mistake. A court ruling that holds the owner of a chatbot responsible for misinformation [12] puts the correctness of information above information-delivery technology.

An organization may not have enough labeled data for supervised learning in order to fine-tune a pre-trained language model. In our experiment data, the median number of sample queries linked to a document is 3. The problem of insufficient amount of training data has been highlighted by the concept of synthetic data for training language models [13]: computer-generated synthetic data mimics the patterns of real-world data and amplifies the amount of training data. On the other hand, Ling et al. [14] argue that the development of a business-specific language model across organization borders will make LLMs a truly disruptive technology. Alternatively, retrieval-augmented generation [15] can combine a pre-trained LLM with domain-specific knowledge. Nevertheless, organizations may not want to use proprietary LLM-based services, such as ChatGPT, due to privacy reasons or operate a private open-source LLM, such as Llama 3, if the investment is not justified [16]. In the future, inexpensive, privacy-enhanced, and domain-knowledgeable LLMs in the right language could be available in a plug-and-play mode, but that topic lies outside the scope of this article.

The main *objective* of this work is to develop lightweight text similarity calculation methods for a small text collection (thousands, not millions, of documents) that do not rely on supervised learning or generative LLMs. By "lightweight", we mean the settings similar to those of cosine similarity [17] and BM25 [18]. Text similarity calculation-wise we expect those lightweight methods to compete with off-the-shelf language models not fine-tuned for the particular text collection.

Our text similarity calculation approach assumes that two terms that co-occur in a text span of the query and a text span of the document imply some third meaning on top of the separate meanings of the two terms. We posit that this third meaning, formalized in a text similarity score, generates additional gravitational force between the query and a relevant document. A term pair comprises terms t_1 and t_2, and co-occurrence co_1, which leads to three weights: $w(t_1)$, $w(t_2)$, and $w(co_3)$. The tuple $(w(t_1), w(co_3))$ adjusts the importance of the term by the importance of one co-occurrence. In the entire document, the tuple $(w(t), \{w(co) \mid t \in co\})$ adds even more influence of the t-co-occurrences to the importance of t. This is analogous to a contextual word embedding, with one significant difference: the context of t alters the importance of t, not the meaning of t as in case of word embeddings.

We establish co-occurrence between two terms, not between two concepts made of word embeddings. We experimented with clustering BERT word vectors into latent concepts; unfortunately, the concepts did not work out. The problem and possible solution are discussed along with future research in Section 7.2.

The *novelty* of this work revolves around the development of a method for measuring the strength of pairwise term co-occurrence, adjusting the importance of a term by the importance of its pairwise term co-occurrences, and theoretical motivation of the choices. Adjustment of the importance of a term in the aforementioned manner analogous to contextual word embedding is the centerpiece of this work.

The *contributions* of this work are: (i) three text similarity calculation methods that utilize pairwise term co-occurrence; (ii) applicability criteria for pairwise term co-occurrence in text similarity calculation; (iii) testing *practical usefulness* of pairwise term co-occurrence against sentence/paragraph embeddings created by an off-the-shelf language model, which is one of the baseline methods; (iv) design principles for text similarity calculation by pairwise term co-occurrence; (v) a unique data set, developed largely within the scope of this work, comprising a variety of text types – email messages, web articles, fill-in forms, brochures – in a restricted domain. The variety of the text types has made it possible to establish the applicability criteria referred to in (ii).

Although a text similarity calculation method does not depend on its users, the flagship use case we have in mind is public organizations. There is a large variety of specific knowledge domains across the organizations. For instance, Sweden has 290 municipalities, 21 counties [19], and 367 agencies that report to the central government [20]. There are more than 90 thousand local governments in the United States [21]. Globally, public organizations are significant consumers of IR applications. Nevertheless, the use case of public organizations has not received much attention from the IR research community since the rise of the large web-based and social-media-based text collections. Our experiment data contributes to filling this void: the texts come from the Swedish Social Insurance Agency, and the mixture of the text types is representative of a public organization.

Another use case for pairwise term co-occurrence could be text similarity calculation on the bottom of the technology stack. For instance, corpus-based question answering "relies on efficient passage retrieval to select candidate contexts, where traditional sparse vector space models, such as TF-IDF or BM25, are the de facto method" [22]. Furthermore, languages with scarce NLP resources are generally more dependent on the "bottom of the technology stack" methods than high-resource languages.

The article is organized as follows. The next section summarizes related research. Section 3 designs the three text similarity calculation methods enhanced by pairwise term co-occurrence. Section 4 presents the experimental setup, whereas Section 5 presents and discusses the experiment results. Section 6 summarizes the key takeaways. Section 7 concludes the article.

2 Related Research: Term Dependencies in Text Matching

The academic interest in plain-text term dependencies started with "A theoretical basis for the use of co-occurrence data in Information Retrieval" [23], had its golden age during the first decade of the 21st century, and came to decline along with the rise of word embeddings. Therefore, an overview of the research on term dependencies in plain text is, to a large extent, a history tour. Within the context of this article, two directions regarding term-dependency research in plain text are most relevant. One direction considers term pairs, query terms that co-exist in a span of document text. The other direction calculates the proximity of query terms in the document text. The latest IR research trends assume word embeddings instead of words and make use of detailed dependencies between concepts obtained by machine learning and large amounts of training text.

2.1 Term-Pair Enhanced Text Similarity Score

Some researchers have used term pairs as text categorization features by, for instance, Naïve Bayes Classifier. The sources of the term pairs differ. Terms from the title of the document may be paired with terms from the abstract of the document [24], or link anchor texts from Wikipedia articles can be the sources of potentially relevant pairwise term co-occurrences, as long as an anchor text contains a "master unigram" from the document [25], [26] considered pairwise term co-occurrence in a document without any restriction regarding the order and distance between the terms. The relevance of co-occurrence was calculated as the number of documents that contained the term pair in the targeted text category divided by the number of equivalent documents in the other text categories.

More often, however, term pairs have been applied in text matching for ad-hoc retrieval. Yang et al. [27] calculated distance between tweets, average 8.56 words in a tweet. A set of term pairs was built from terms that co-existed in individual short texts; stop words and rare term pairs were removed. Thus, each short text became a subset of the term pairs. Let N(w) be a set of all terms that co-occur with term w, and the distance measure between two terms w and v be Jacc(w,v) = 1-|N(w)∩N(v)|/|N(w)∪N(v)|. Then the distance between two term pairs is d({w_1,w_2},{v_1,v_2}) = Jacc(w_1,v_1)·Jacc(w_1,v_2)·Jacc(w_2,v_1)·Jacc(w_2,v_2)/4. The distance between two short texts is the sum of the distances between each term pair in one text with each term pair in the other text.

A more straightforward way to establish a term co-occurrence weight is to calculate normalized pointwise mutual information between two terms [28] and add pre-calculated pairwise term co-occurrence weights to unigram weights to build the text similarity score. In a similar spirit, Kim et al. [29] established unidirectional associations between terms in a document and calculated confidence of an association as c(t_1,t_2) = n_{12} / n_1, where n_{12} is the number of documents that contain t_1 and t_2, whereas n_1 is the number of documents that contain t_1. The weight of t was recalculated following the format λ·w(t)+(1-λ)·Σw(t_i)·c(t_i,t)/dl, where dl is document length, and used in vector-space and language modeling framework for query-document similarity calculation.

Within the language modeling framework for text similarity calculation, Shi and Nie [30] calculated the strength of pairwise term co-occurrence as probability P({t_1,t_2}|Q)·log(P({t_1,t_2}|D_w)), where Q is the query and D_w is a context window in document text. The final query-document similarity score was a polynomial of three equivalent scores calculated for term co-occurrence, bigrams, and unigrams. Gao et al. [31] used statistical dependency parsing in a query to create an acyclic, planar graph that linked query terms. Such a graph is, in essence, a set of term pairs. The query-document similarity score was made from probabilities that individual terms and linkages (i.e., terms pairs) appeared in the document.

2.2 Term-Proximity Enhanced Text Similarity Score

Term proximity has been considered for text retrieval with short queries because "phrases and proximity terms have a strong positive impact on web retrieval effectiveness for extremely short queries (2 or 3 terms), while they have less, or even negative, effect on longer queries" [32]. Our literature study has identified two main paths towards utilizing term proximity in plain-text matching. One path calculates query-document similarity as a polynomial of unigram weights and the weights of term proximity features. The other path places the weights of term proximity features into BM25 or the language modeling framework as modified term frequency.

2.2.1 Polynomial of Unigram Weights and Term-Proximity Weights

Query-document similarity score may take the format λ·Σw(t)+(1-λ)·Σw(c), where Σw(t) is a score by BM25 or the language modeling framework, w(c) is a weight of term proximity features, and λ is a tuning parameter. The rest of this subsection describes how these w(c) or Σw(c) have been calculated.

Rasolofo and Savoy [33] paired every query term with each other. In the document, a term-pair instance weight was calculated as 1/d(t_1,t_2)^2, where d(t_1,t_2) is the number of spaces between terms t_1 and t_2. The final term-pair weight was calculated by the BM25 term frequency formula where the sum of term-pair instance weights was used instead of the number of term occurrences in a document. Some modifications of this approach have been tested [34]–[36].

Following the same logic, Lu et al. [37] built a collection of subqueries from the original query. For each subquery, the document was split into non-overlapping intervals where one interval contained all the subquery terms. Interval score(t_{left},t_{right}) was calculated as IDF(t_{left})·IDF(t_{right}) / (pos(t_{right})-pos(t_{left})+1)^2, where IDF is inverse document frequency. The score of the subquery was calculated from the interval scores by the BM25 term frequency formula the same way as Rasolofo and Savoy did [33].

Tao and Zhai [38] explored span-based (the largest or smallest span that covers all the query terms in a document) and pairwise (the largest, smallest, and average distances between all query-word pairs in a document) term proximities. In order to obtain a term proximity score, the term proximity value p was placed into $\log(\lambda+\exp(-p))$.

Cross term [39] is a pseudo term made from two neighboring query terms in a document. The influence of a term t is modelled by a bell-shaped curve: cross term t_{12} occurs where the curves of t_1 and t_2 intersect. The strength of t_{12} is the curves' function value at the intersection point. Cross-term frequency $\mathrm{tf}(t_{12})$ in a document is calculated as follows: first, all the instances of t_{12} are identified, then the sum of all the instances' strength values becomes $\mathrm{tf}(t_{12})$. The number of documents that contain t_{12} is calculated as follows: first, an appearance value of t_{12} in a document is calculated as $\mathrm{tf}(t_{12})$ divided by the number of instances of t_{12} in the document, then the number of documents that contain t_{12} is the sum of t_{12} appearance values across all the documents. The cross-term frequencies and number of documents are included into BM25 and the language modeling framework to calculate the cross-term text similarity score.

Sometimes, term proximity is not calculated explicitly. Instead, terms are required to be near each other. One of the most cited works on term dependencies [40] combined unigrams, n-grams, and groups of unordered terms within the language modeling framework for text similarity calculation. The weight of a query unigram t was calculated as $\lambda\cdot\mathrm{tf}(t,\mathrm{D})/|\mathrm{D}|+(1-\lambda)\cdot\mathrm{tf}(t,\mathrm{C})/|\mathrm{C}|$, where $\mathrm{tf}(t,\mathrm{D})$ and $\mathrm{tf}(t,\mathrm{C})$ are term frequency in the document and the entire document collection, $|\mathrm{D}|$ and $|\mathrm{C}|$ are the number of terms in the document and the entire document collection. Likewise, the weight of a query n-gram (an ordered sequence of two or more terms) and an unordered group of query terms within a fixed-size context window were calculated.

Another method [41] identified noun phrases in a query: a proper name, a dictionary phrase from WordNet, a simple phrase 2–4 words long with at least two "content words", a complex phrase that had one or more dictionary phrases or simple phrases embedded. During query-document matching, the words from a query noun phrase were matched to fixed-size context windows in the document; the matching rules varied for different types of noun phrases. If there was a match, the weight of the noun phrase was its IDF. If the noun phrase was a complex phrase, IDFs of the sub-phrases were added. The noun-phrase-based query-document similarity score was the sum of all the above IDFs.

While a query term may not appear in a document directly, it may still be relevant to the document topic-wise, "topic" as in topic modeling [42]. In a topic-mediated term weight, such relevance is expressed as $\mathrm{P}(t|z)\cdot\mathrm{P}(z|\mathrm{D})$ – probability that term t belongs to topic z multiplied by the probability that topic z belongs to document D.

Once the above weights of term proximity features were calculated, they were added to unigram-based text similarity scores, which resulted in the final query-document similarity score.

2.2.2 Term Proximity Embedded in BM25 or the Language Modeling Framework

Song et al. [43] introduced term relevance contribution $rc(t)$ that would replace the number of term occurrences in a document in BM25. The value of $rc(t)$ was calculated as follows: first, the document was split into non-overlapping spans where each span contained as many unique query terms as possible, then $rc(t,span) = n(span)^{\lambda}/width(span)^{\Upsilon}$, where t is a query term, $span$ is one of the spans that cover t, $n(span)$ is the number of query terms that occur in the $span$, λ and Υ are tuning parameters. The final $rc(t)$ was the sum of $rc(t,span)$ across all the spans that covered t.

Pseudo-frequency [44] was used the same way as the above relevance contribution. For any appearance of query term t in a document sentence, $span$ covered the distance from t to the closest other query term in the sentence. Pseudo-frequency was calculated as $pf(t,span) = 1+1/width(span)^{\lambda}$. If there was no span, then $pf(t)=1$.

The proximity language model [45] measured $d(t)$ – distance score of a query term t in a document – as either (i) the number of words from t to the closest other query term in the document or (ii) the average, or (iii) the sum of the number of words from t to every other query term in the

document. The term's proximate centrality was calculated as $1/\lambda^{d(t)}$, and worked into a query-document similarity score by the language modeling framework.

TextRank [46] has copied the ideas of PageRank [47] and applied them to terms in a document, where co-occurrence between two terms in a context window was defined as a link. TextRank has been used for document retrieval with a fixed-size context window [48] and a dynamic sentence-size context window [49], where query-document similarity was calculated by BM25, and TextRank replaced the usual term frequency formula.

2.2.3 Machine-Learning Enhanced Text Similarity Calculation

ChatGPT took the world by surprise at the end of 2022 and demonstrated that IR is undergoing a paradigm shift. LLMs is a big step forward in natural language understanding by computers. Some researchers view GPT-4 "as an early (yet still incomplete) version of an artificial general intelligence" [50].

LLMs are deep neural networks that are based on the highly scalable transformer architecture [51] and have been pre-trained on massive amounts of data to learn representations of language (or other data modalities). Pre-training LLMs involves self-supervision on tasks constructed from unlabeled corpora, e.g., predicting a masked word based on its context (masked language modeling) or predicting the next word in a sentence (next word prediction). Once pre-trained, LLMs can either learn to perform a specific downstream task in a process, i.e., fine-tuning, whereby some or all of its parameters are updated based on a relatively small set of labeled task-specific examples, or through in-context learning [52], whereby a natural language description of the task is provided in the LLM prompt itself, either with a few examples of input-output pairs (few-shot learning) or without any such examples (zero-shot learning).

Given the size of recent LLMs, typically consisting of billions of parameters, and the intractability of fully fine-tuning these for specific tasks, efforts have primarily focused on parameter-efficient fine-tuning [53], [54], whereby a small set of additional parameters are trained, or by improving the in-context learning capabilities of LLMs, whereby none of the LLM's parameters are updated, e.g., through chain-of-thought prompting [55] and combining reasoning with action plan generation, e.g., ReAct [56]. In fact, recent LLMs, such as PaLM [57] and GPT-4, have been shown to exceed the capabilities of smaller transformers like BERT, even without fine-tuning. Recently, there has been considerable interest in improving the capabilities of LLMs by allowing them to leverage external knowledge bases, i.e., retrieval-augmented LLMs, and tools, i.e., tool-augmented LLMs. Some of these efforts fall within the scope of LLM-based autonomous agents [58].

BERT is an example of a masked language model and arguably the most popular transformer model before GPT revealed its capabilities. Similar to an LLM, a BERT language model is pre-trained on a large amount of text and then fine-tuned for the particular downstream task. The BERT's ability to disambiguate a term by encoding the context of the term – the sequences of neighboring terms to the left and to the right – into the embedding of the term has proved superior. Sentence-BERT [59], alias SBERT, is optimized for sentence embeddings and is likely to work with paragraphs as well.

BERT has been applied in automated question answering. Because transformers have fixed-length input, e.g., 512 text tokens with BERT, a larger document is split into passages. After the initial retrieval of passages by, for instance, BM25, the passages are re-ranked by BERT. The query-passage similarity score may be the cosine similarity of query-passage text embeddings [60] or the probability that the query and the passage are relevant, calculated by BERT as a binary classification model [61], or the sum of selected cosine similarities for term embeddings in a query and a passage [62].

For document retrieval, the document is split into overlapping passages, paragraphs, or sentences. The aggregated score may be a linear combination of the initial query-document similarity score (e.g., BM25) and the BERT-generated query-passage similarity scores [7], or only BERT-generated first-passage score, or best-passage score, or the sum of all passage scores [63].

A more advanced approach, different from score aggregation, is query-passage representation aggregation [64]. The BERT embedding of a query-passage pair is a vector. The vectors are aggregated across all the passages in the document, and then the query-document similarity score is calculated from the aggregation result.

3 Text Similarity by Pairwise Term Co-occurrences

Term pair is short for a pairwise term co-occurrence. A term pair is unordered; the co-existence relationship has no direction. *Bag of word pairs* designates a collection of term pairs the same way as bag of words designates a collection of terms. In Section 3.1, term co-occurrences are established, and a bag of word pairs is created given a query and a document. In Sections 3.2 and 3.3, the bag of word pairs is used to calculate the numeric query-document similarity score.

3.1 Creating Term Pairs

Both the query and the document are split into text chunks, where a text chunk is a bag of words. Hence, the query and the document each are a set of bags of words. In the literature, similar to our text chunk is "context window". The size of the window can be fixed, or depend on linguistic features of the co-occurring terms [41], or have natural boundaries such as a sentence or a paragraph [49]. The exact scopes of the text chunks for our experiments are defined in Section 4.3.

Given a query *q* and a document *d*, a bag of word pairs *bowp* is created as follows:

```
for each text chunk qch ∈ q :
     for each text chunk dch ∈ d :
          for each term t₁ ∈ dch & t₁ ∈ qch :
               for each term t₂ ∈ dch & t₂ ∈ qch & t₂ ≠ t₁ :
                    if {t₁,t₂} ∉ bowp then add {t₁,t₂} to bowp
```

This basic algorithm contains embedded "for each" loops, which makes it impractically slow in case of too many text chunks in too many documents to process. Therefore, the document collection should be indexed for speedy retrieval of a limited number of potentially relevant text chunks and subsequent query-document matching. Indexing of the text chunks lies outside the scope of this article.

If a text chunk contains *n* words, then the number of possible unordered term pairs in the chunk is $n \cdot (n\text{-}1)/2$. In the literature, *pruning of pairwise term co-occurrences* has been done by setting a threshold for a co-occurrence weight calculated as

- mutual information within the boundaries of a sentence ([28], [65], [66]) or a document [67];
- the number of documents where a particular co-occurrence appears [36];
- Jaccard similarity coefficient [29].

In our bag of word pairs, the number of term pairs is reduced right from the start. Only the terms that co-occur in both a document chunk and a query chunk are selected; term pairs appear during a query-document matching process and disappear when the bag of word pairs is disposed of. Furthermore, we reduce the number of eligible terms and term pairs by filtering out stop words. We manually inspected a list of most frequent words in our experiment data and subjectively selected the stop words. A complete list of the stop words is available in Appendix A.

3.2 Text Similarity by a Bag of Word Pairs

If a query-document matching instance generates a sufficient number of term pairs, we may calculate text similarity using the term pairs only. The numeric value of the text similarity depends on (i) how representative the co-occurrence in the document collection is and (ii) the weights of the co-occurring terms.

3.2.1 Term Loyalty and Inverse Chunk Frequency

We measure the importance of a pairwise term co-occurrence by two features: (i) how likely the two terms are to appear in the company of each other as opposed to appearing independently of each other, and (ii) how common the co-occurrence is in the collection.

Term loyalty measures the first feature of pairwise term co-occurrence. If n($\{t_1,t_2\}$) is the number of text chunks in the document collection where both terms t_1 and t_2 appear, n(t_1) and n(t_2) are the numbers of text chunks where t_1 and t_2 appear either together or separately, then term loyalty is calculated as

$$\text{loyalty}(\{t_1,\ t_2\}) = \frac{n(\{t_1,\ t_2\})}{\max\left(n(t_1), n(t_2)\right)} \tag{1}$$

Standard Jaccard similarity coefficient has the denominator n(t_1)+n(t_2)-n($\{t_1,t_2\}$), where both terms are equal. We tried it, yet our denominator yielded slightly better retrieval precision. The more frequent of the two terms turned out slightly more influential.

A representative for text similarity is a not-too-common co-occurrence with good IDF. Because we have text chunks instead of documents, we measure inverse chunk frequency (ICF):

$$\text{icf}(\{t_1, t_2\}) = \log_2 \frac{N}{n(\{t_1,\ t_2\})} \tag{2}$$

where N is the total number of text chunks in the document collection (or in the collection of text chunks). All our logarithms have the base 2, which was decided experimentally.

Shirakawa et al. [68] argue that the traditional IDF formula, if applied to phrases, assigns disproportionally high weight to uncommon phrases. Shirakawa et al. propose more general IDF formulas for n-grams where unigrams are a special case. If we replace n-grams in [68] Shirakawa et al.'s formulas with our term pairs, we obtain formula (2). Therefore, we consider our ICF being good enough.

We calculate pairwise term co-occurrence weigh as

$$w(\{t_1, t_2\}) = \text{loyalty}(\{t_1, t_2\}) \cdot \text{icf}(\{t_1, t_2\}) \tag{3}$$

3.2.2 Query-Document Similarity Score

We calculate a numeric text similarity score as the sum of all the term-pair weights:

$$\text{score}(P) = \sum_{\{t_1,t_2\}\in P} \left(w(t_1) + w(t_2)\right) \cdot w(\{t_1, t_2\}) \tag{4}$$

where P is a bag of word pairs, the term weights $w(t_1)$ and $w(t_2)$ are not specified yet, and $w(\{t_1,t_2\})$ is formula (3). Because a term appears in an unordered term pair only once, we can rearrange monomials without changing the sum of the polynomial:

$$\text{score}(P) = \sum_{t\in T(P)} w(t) \cdot \sum_{\{t,*\}\in P} w(\{t,*\}) \tag{5}$$

where T(P) is the set of terms from all the term pairs in P. Because we will reuse the sum of term co-occurrence weights, we define it separately:

$$w(t, P) = \sum_{\{t,*\}\in P} w(\{t,*\}) \tag{6}$$

The text similarity score becomes

$$\mathrm{score}(\mathrm{P}) = \sum_{\mathrm{t}\in\mathrm{T}(\mathrm{P})} \mathrm{w}(\mathrm{t}) \cdot \mathrm{w}(\mathrm{t},\mathrm{P}) \tag{7}$$

In formula (7), $w(t)$ is affected by the co-occurrence of t with other terms in a number of text chunks in the entire document: well-connected t and t having important connections becomes more important itself. This is analogous to a contextual word embedding, except the context modifies the importance of t, not the meaning of t.

Formula (6) is biased towards larger documents because they are likely to generate a larger number of unique pairwise co-occurrences per term. A large non-relevant document may generate many pairwise co-occurrences with weak $w(\{t,*\})$ values and outcompete a small relevant document with just a few strong $w(\{t,*\})$ values. In formula (7), the bias is strengthened by more terms in T(P) and, therefore, more iterations. In our experiments, the size effect had a strong influence if the difference in document sizes was large, e.g., tenfold. In a collection of documents with comparable sizes, the size effect was not noticeable.

We are not aware of any research regarding the normalization of term associations in a document. We tested a few options on our own: normalization towards the density of term pairs per text unit, normalization towards the density of pairwise co-occurrences per unique term in a bag of word pairs, boosting documents shorter than average, and removing low-weight term co-occurrences. Any attempt of normalization leads to a considerable precision drop in ad-hoc retrieval. Most probably, straightforward normalization is not an option; for comparison, BM25 does normalization of term frequency by document length in a rather sophisticated way, see formula (10).

In formula (7), the format of text similarity calculation is that of BM25: the formula iterates through the terms shared by the query and the document and sums up their weights. Because BM25 is a state-of-the-art framework, we reuse BM25's document term frequency and IDF and calculate query-document similarity as

$$\mathrm{sim}_{\mathrm{TP}}(\mathrm{q},\mathrm{d}) = \mathrm{score}\big(\mathrm{P}(\mathrm{q},\mathrm{d})\big) = \sum_{\mathrm{t}\in\mathrm{T}(\mathrm{P})} \mathrm{idf}_{\mathrm{BM}}(\mathrm{t}) \cdot \mathrm{dtf}_{\mathrm{BM}}(\mathrm{t}) \cdot \mathrm{qtf}_{\mathrm{TP}}(\mathrm{t}) \cdot \mathrm{w}(\mathrm{t},\mathrm{P}) \tag{8}$$

$$\mathrm{idf}_{\mathrm{BM}}(\mathrm{t}) = \log_2 \frac{\mathrm{N} - \mathrm{df}(\mathrm{t}) + 0.5}{\mathrm{df}(\mathrm{t}) + 0.5} \tag{9}$$

$$\mathrm{dtf}_{\mathrm{BM}}(\mathrm{t}) = \frac{\mathrm{tf}_{\mathrm{d}}(\mathrm{t})}{\mathrm{tf}_{\mathrm{d}}(\mathrm{t}) + \mathrm{k}_1 \cdot \left(1 - \mathrm{b} + \mathrm{b} \cdot \frac{\mathrm{dl}}{\mathrm{avgdl}}\right)} \tag{10}$$

where N is the number of documents in the collection, df(t) is the number of documents that contain term t, $\mathrm{tf}_d(t)$ is the number of appearances of t in a document, dl is document length in words, $avgdl$ is the average document length in the collection. A good interval for k_1 is between 1.2 and 2 [69]; we experimented and set k_1=2. The value of b was set to 0.75. Query term frequency $\mathrm{qtf}_{TP}(t)$ is the total number of occurrences of t in those query chunks that contain term pairs $\{t,*\}\in$P; the version of query term frequency used by standard BM25 and calculated by formula (12) in Section 3.3.1 did not work well in formula (8).

In formulas (9) and (10), the scope of the text is the entire document, as required by BM25, regardless of chunks; an attempt to apply formulas (9) and (10) to text chunks instead of documents did not work well in our experiments.

Previous research ([39,] [43], [44]) has applied the term proximity within a document to modify or replace $\mathrm{tf}_d(t)$ in formula (10). In our experiments, modifying $\mathrm{tf}_d(t)$ by formula (6) and placing it back into formula (10) did not work. The reason is, arguably, the different scopes of $\mathrm{tf}_d(t)$ and formula (6): $\mathrm{tf}_d(t)$ has the scope of a document, whereas term loyalty and ICF embedded in formula

(6) have the scope of the text-chunk collection. We argue that collection-wide term co-occurrence features, such as term loyalty and ICF, should have the same level of influence on term weights as IDF has.

3.3 Bag of Word Pairs Combined with Bag of Words

BM25 and cosine similarity are two state-of-the-art bag-of-words matching methods. We introduce term co-occurrence weights into both methods with the aim of increasing retrieval precision without compromising recall.

3.3.1 BM25

BM25 has a number of similar alternatives introduced over the decades [18]. Our BM25 alternative is as follows:

$$\mathrm{sim}_{\mathrm{BM}}(\mathrm{q},\mathrm{d}) = \sum_{\mathrm{t}\in\mathrm{q}\cap\mathrm{d}} \mathrm{idf}_{\mathrm{BM}}(\mathrm{t}) \cdot \mathrm{dtf}_{\mathrm{BM}}(\mathrm{t}) \cdot \mathrm{qtf}_{\mathrm{BM}}(\mathrm{t}) \tag{11}$$

$$\mathrm{qtf}_{\mathrm{BM}}(\mathrm{t}) = \frac{(\mathrm{k}_3 + 1) \cdot \mathrm{tf}_{\mathrm{q}}(\mathrm{t})}{\mathrm{k}_3 + \mathrm{tf}_{\mathrm{q}}(\mathrm{t})} \tag{12}$$

where $\mathrm{idf}_{\mathrm{BM}}(t)$ and $\mathrm{dtf}_{\mathrm{BM}}(t)$ are formulas (9) and (10), $\mathrm{tf}_{\mathrm{q}}(\mathrm{t})$ is the number of appearances of term t in a query. For k_3, there are several options in the literature: k_3=1000 ([70], [71]), k_3=8 ([72]), k_3=1.2 ([73]), and k_3=0 ([74]). Because a larger k_3 value worked better for us, we set k_3=1000.

Term co-occurrence weights enter BM25 as

$$\mathrm{sim}_{\mathrm{BMTP}}(\mathrm{q},\mathrm{d}) = \sum_{\mathrm{t}\in\mathrm{q}\cap\mathrm{d}} \mathrm{idf}_{\mathrm{BM}}(\mathrm{t}) \cdot \mathrm{dtf}_{\mathrm{BM}}(\mathrm{t}) \cdot \mathrm{qtf}_{\mathrm{BM}}(\mathrm{t}) \cdot \big(1 + 0.5 \cdot \mathrm{w}(\mathrm{t},\mathrm{P})\big) \tag{13}$$

The reader may observe that formulas (8) and (13) are similar but different: (i) formula (13) iterates through the terms shared by the query and the document, whereas formula (8) narrows down the set of terms to only those in the bag of word pairs; (ii) $\mathrm{qtf}_{\mathrm{BM}}(t)$ and $\mathrm{qtf}_{\mathrm{TP}}(t)$ are calculated differently; (iii) formula (13) has "1+0.5" whereas formula (8) has not. The differences come from the design of the formulas: in formula (8), term weights strengthen term co-occurrence weights, whereas in formula (13), the term co-occurrence weights have been included to strengthen the term weights.

3.3.2 Half of the Co-occurrence Weight

In formula (13), the multiplier 0.5 was initially established as an empirical constant with its best value between 0.4 and 0.6. The value 0.5 also has a rational explanation. Let $\mathrm{w}(t)$ be $\mathrm{idf}_{\mathrm{BM}}(t)\cdot\mathrm{dtf}_{\mathrm{BM}}(t)\cdot\mathrm{qtf}_{\mathrm{BM}}(t)$, then formula (13) can be expanded as

$$\mathrm{w}(t_1) + \underline{0.5\cdot\mathrm{w}(t_1)\cdot(\mathrm{w}(t_1,t_2)} +\ldots) + \mathrm{w}(t_2) + \underline{0.5\cdot\mathrm{w}(t_2)\cdot(\mathrm{w}(t_1,t_2)} +\ldots) +\ldots =$$
$$\mathrm{w}(t_1) + \mathrm{w}(t_2) + \underline{0.5\cdot\mathrm{w}(t_1,t_2)}\cdot(\mathrm{w}(t_1) + \mathrm{w}(t_2)) +\ldots$$

Either half of the co-occurrence weight is applied to each of the two terms in the co-occurrence, which is $\underline{0.5\cdot\mathrm{w}(t_1,t_2)}\cdot(\mathrm{w}(t_1) + \mathrm{w}(t_2))$, or the entire co-occurrence weight is applied to the average of the two-term weights, which would be $\mathrm{w}(t_1,t_2)\cdot\underline{0.5\cdot(\mathrm{w}(t_1) + \mathrm{w}(t_2))}$.

Sordoni et al. [75] expressed concern that combining scores obtained separately from matching single terms and from matching compound dependencies may count a dependency twice, as a compound and as a component. Indeed, our experiments confirm Sordoni et al.'s [75] concern about the excess impact of term dependencies: applying the entire co-occurrence weight to each term in the co-occurrence lowered retrieval precision. Therefore, formula (13) applies only half of the co-occurrence weight to each of the two terms in the co-occurrence.

3.3.3 Cosine Similarity

Cosine similarity is calculated as

$$\mathrm{sim}_{\mathrm{CS}}(q,d) = \frac{\sum_{i=1}^{N} q_i \cdot d_i}{\sqrt{\sum_{i=1}^{N} {q_i}^2} \cdot \sqrt{\sum_{i=1}^{N} {d_i}^2}} \tag{14}$$

where q_i and d_i are term weights calculated as

$$w_{\mathrm{CS}}(t) = \mathrm{tf}(t) \cdot \mathrm{idf}(t) \tag{15}$$

where the scope of term frequency is the entire query or document, and the scope of IDF is documents in the document collection. We include pairwise term co-occurrence in cosine similarity as term weight

$$w_{\mathrm{CSTP}}(t) = w_{\mathrm{CS}}(t) \cdot \big(1 + 0.5 \cdot w(t,P)\big) \tag{16}$$

It is difficult to trace how the multiplier 0.5 impacts formula (14), but our experiments confirm that the logic explained in Section 3.3.2 holds here as well: each of the two terms in a co-occurrence gets a fair half of its co-occurrence weights.

4 Experimental Setup

The text similarity calculation methods (listed in Table 1) were tested in ad-hoc retrieval. The test target methods – TP, BMTP, and CSTP – make use of pairwise term co-occurrence. Two baseline methods – BM and CS – are still state-of-the-art unigram-based text similarity calculation methods. Two other baselines – KIM and YANG – also use pairwise term co-occurrences, but of a different kind. Finally, SBERT uses a pre-trained language model to generate sentence/paragraph embeddings. SBERT is arguably the most popular sentence/paragraph transformer, with over 6000 language models released on HuggingFace [76].

Table 1. Tested text similarity calculation methods

Acronym	Method	Description
Test target methods		
TP	Formula (8)	Term-pair-based text similarity without unigrams
BMTP	Formula (13)	BM25 with term co-occurrence weights in term weights
CSTP	Formulas (14)+(16)	Cosine similarity with term co-occurrence weights in term weights
Baselines		
BM	Formula (11)	BM25, unigrams only
CS	Formulas (14)+(15)	Cosine similarity, unigrams only
KIM	[29]; Section 2.1	Unidirectional term associations transfer influence of one term to the other term; normalization by document length; $\lambda = 0.5$
YANG	[27]; Section 2.1	Jaccard distance between two terms by co-occurring terms in short texts, then distance between term pairs, then distance between texts
SBERT	Cosine similarity between text embeddings	Off-the-shelf SBERT for sentence/paragraph embeddings without fine-tuning the BERT language model; more details in Section 4.4

4.1 Document Collection

The experiment data was two document collections. One document collection was email messages sent by citizens to handling officers of the Swedish Social Insurance Agency. The other collection was documents from the website of the same agency. All the texts were in Swedish, which belongs to the Germanic branch of Indo-European languages.

The email messages were released to a group of researchers within an earlier research project. After cleaning the data, we got 8630 plain-text messages with no metadata. Table 2 shows the eight text categories of the messages, the number of messages per category, and the average number of words per message after compound splitting, without stop words. The sum of the category messages is more than the total because 43 messages belong to two categories. Development of the email collection is covered in [77].

Table 2. Email categories, the number of messages per category, and the average number of terms per message with standard deviation

Identifier	Category title	Number of messages	Avg message size
Cat1	Please send me a fill-in form	390	17.94 ± 14.91
Cat2	When will you decide my housing allowance?	290	16.16 ± 9.83
Cat3	How many days of parental benefits do remain for my child?	181	17.31 ± 13.89
Cat4	How much will I get in my future pension?	148	19.92 ± 13.73
Cat5	When do I get my money?	1285	16.07 ± 11.93
Cat6	Questions regarding child allowance	192	31.17 ± 25.78
Cat7	Please send me a European Health Insurance Card	59	17.03 ± 13.46
Rest	All other messages	6128	28.22 ± 38.70
	Total	8630	25.26 ± 33.95

The email messages contained many unusual abbreviations whose meanings one could guess from the context; therefore, we expanded as many abbreviations as we could while doing spelling corrections. We removed greetings and pleasantries, such as “dear sir/madam”, “sincerely yours” and similar, as well as advertisements attached at the end of the messages by free email accounts. In matching email queries to email documents, such duplicate pieces of text would generate term co-occurrences that falsely signal text similarity.

The website documents were references by an XML sitemap for search engines. Table 3 outlines the structure of that document collection. 154 web articles, 70 fill-in forms, and 27 brochures were linked as relevant documents to at least one query according to the relevance criteria defined in Table 5. The other documents merely filled the collection.

Table 3. Documents from the website, the entire collection

Web articles				PDF documents						Total num
Relevant		Other		Relevant fill-in forms		Relevant brochures		Other		
Num	Size	Num	Size	Num	Size	Num	Size	Num	Size	
154	806 ± 857	437	392 ± 689	70	342 ± 256	27	31628 ± 31800	1387	9048 ± 11823	2075

Because formula (6) is biased towards larger documents, we created a reduced-size document collection outlined in Table 4. We kept all the web articles and fill-in forms, calculated their average size and the standard deviation, and removed PDF documents larger than the just-calculated average size plus standard deviation.

Table 4. Documents from the website, the reduced-size collection

Web articles		PDF documents			Total num
Num relevant	Num other	Num fill-in forms	Num brochures	Num other	
154	437	70	1	454	1116

In all HTML documents, navigation blocks, headers, and footers were removed. Fill-in forms share captions to identify a person (name and surname, social security number, town and zip-code, etc.), the organization, and bureaucratic procedures; such duplicate pieces of text were removed from fill-in form queries but remained in the documents to be retrieved.

For bag of words and bag of word pairs, compound words were split into components, all words were lemmatized.

4.2 Linking Documents to Queries

The relevance criterion for matching email queries to email documents is straightforward: if both belong to the same text category, Cat1–Cat7, the document is relevant to the query.

Five email categories – Cat1, Cat2, Cat3, Cat6, and Cat7 – were selected as the source of email queries that may have relevant website documents. Cat4 had no relevant website documents because the social insurance agency had discontinued handling pensions. Cat5 was a too-large category with diverse, complex, and personal inquiries; therefore, these messages were not linked to website documents. An annotator processed each candidate query individually and linked the documents by binary relevance judgments according to the criteria in Table 5. Once the email messages were linked to the documents, another round of query-document relevance judgments was carried out: the web articles and fill-in forms linked to the email queries became candidate queries themselves linked to web articles, fill-in forms, and brochures.

The relevance criteria in Table 5 were defined by an annotator who had previous experience with the texts from the Swedish Social Insurance Agency and assessed by a reviewer. Both researchers were in agreement regarding the relevance criteria.

Table 5. Query-document relevance criteria

Query	Relevant document		
	Web article	**Fill-in form**	**Brochure**
Email	The article answers the issue posed in the query. However, the answer is generic and does not address every nuance found in the query. Think FAQ.	The form is explicitly requested, or the query suggests that the author would like to apply for a public service that requires filling the application form.	The brochure explicitly mentions the issue posed in the query. A brochure does not normally answer a specific question.
Web article	(i) The same information either summarized or more detailed. (ii) A subtopic of the query; e.g., the query addresses subsidies for health-related work aids, the linked article addresses subsidized hearing aids. (iii) An equivalent service for a different group of recipients; e.g., one articles addresses sickness benefit for those studying in Sweden, the other one – for those studying abroad.	The query addresses a public service which requires an application by filling the form. Usually the form, or its online equivalent, is linked from the query text.	The brochure explicitly mentions the issues addressed in the query. If there are no issues, the brochure describes the public service addressed in the query.
Fill-in form	The article addresses a public service that makes use of the fill-in form. Usually the article has a link to the form or its online equivalent.	(i) Both forms address the same service but with different details. (ii) The main and supplementary application forms are treated as mutually relevant.	The brochure describes parts of a public service that make use of the fill-in form. Typically, the query is an application form.

Table 6 shows the number of queries that have at least one relevant web article, fill-in form, or brochure.

Because we had more than a thousand queries, and each query required individual research, we could not afford to have another annotator repeat the process in order to calculate inter-annotator agreement, nor there were any annotators with domain experience readily available. As a means of quality control, we randomly selected 20% of the queries per email category (at least 20 messages), 20 web-article queries, and 20 fill-in-form queries to be inspected by a reviewer. The inspection addressed precision-related, not recall-related, judgments because our goal was to compare the precision of text-matching methods at the existing recall values. The review did not lead to any changes in the query-document relevance judgments. The review ensured that the query-document relevance judgments maintained a high quality and that the relevance criteria had been consistently applied.

Table 6. Number of queries that have at least one relevant web article, fill-in form, or brochure

Query	Relevant document		
	Web article	Fill-in form	Brochure
Email	725	230	564
Web article	82	54	64
Fill-in form	41	34	30

4.3 Scope of Pairwise Term Co-occurrence

Previously, we have worked with sentence-wide and paragraph-wide text-pattern matching [78]. In this research, we proceeded with natural text-chunk boundaries and experimentally selected two scopes of pairwise term co-occurrence: a sentence or the entire query text. Table 7 defines the use of the scopes.

Table 7. Profiles of pairwise term co-occurrence context

Acronym	Min size of a bag of word pairs	Query		Document	
		Text type	Scope of a text chunk	Text type	Scope of a text chunk
TC1	2	Email message	Entire query text	Any	Sentence
TC2	0				
TC3	2	Web article	Sentence		
TC4	0				
TC5	2	Fill-in form	Entire query text		
TC6	0				

We have organized the scopes of text chunks into six profiles of pairwise term co-occurrence context. Profiles TC1, TC3, and TC5 require that a query-document matching instance generates at least two term pairs for text similarity calculation by formula (8). This makes the density of relevant documents higher and inflates precision while lowering recall because some relevant documents are lost.

Profiles TC2, TC4, and TC6 allow text similarity calculation with zero or more term pairs, which guarantees 100% retrieval recall and non-inflated precision.

4.4 Text Segmentation for SBERT Embeddings

We used the Python's SentenceTransformer framework and a pre-trained language model [79] for generating SBERT text embeddings. The pre-trained language model was not fine-tuned for our text corpus in order to allow for a fair comparison with the TP, BMTP, and CSTP text similarity calculation methods that are not based on supervised machine learning. Neither did our use case have much fine-tuning data.

Because an SBERT embedding limits a piece of input text to 512 BERT-tokens, web articles and brochures were split into paragraphs (pieces of text typically focused on a single issue), each paragraph got an embedding, and mean pooling was used to obtain the document embedding. The

overwhelming majority of email messages were less than 512 BERT-tokens long; therefore, almost every message got its own embedding. The few longer texts were split into a maximum 400-token spans; each span got an embedding, and mean pooling was used to obtain the document embedding. Fill-in forms contain abrupt captions, splitting the text into paragraphs did not work well, therefore fill-in forms were processed the same way as email messages.

Table 8 defines the profiles of text segmentation for SBERT text encoding before any mean pooling of the embeddings is applied.

Table 8. Profiles of text segmentation for SBERT text encoding

Acronym	Query		Document	
	Text type	Scope of a text embedding	Text type	Scope of a text embedding
TC7	Email message or fill-in form	Entire query text or a 400-token span	Email message or fill-in form	Entire document text or a 400-token span
TC8			Web article or brochure	Paragraph
TC9	Web article	Paragraph	Fill-in form	Entire document text or a 400-token span
TC10			Web article or brochure	Paragraph

4.5 Performance Measures

We used *mean average precision* (MAP) and *mean precision at one* (mean P@1) to compare the text similarity calculation methods listed in Table 1.

Precision is the share of retrieved documents relevant to the query among all the retrieved documents. Precision at *k* (P@k) is precision for the top *k* retrieved documents. P@1 has two possible values: 1 if the top retrieved document is relevant to the query and 0 otherwise. Mean P@1 across all the queries is calculated as

$$\text{Mean P@1} = \frac{1}{|\text{Q}|}\sum_{\text{q}\in\text{Q}} \text{P@1}_\text{q} \tag{17}$$

where Q is the set of queries. Average precision for query q – AP_q, and MAP across all the queries are calculated as

$$\text{AP}_\text{q} = \frac{1}{\text{N}}\sum_{\text{k}=1}^{\text{n}} \text{P@k}\cdot\text{rel(k)} \tag{18}$$

$$\text{MAP} = \frac{1}{|\text{Q}|}\sum_{\text{q}\in\text{Q}} \text{AP}_\text{q} \tag{19}$$

where n is the number of retrieved documents, N is the number of retrieved relevant documents, rel(k) is 1 if the k^{th} document is relevant to the query, and 0 otherwise.

For precision of email retrieval by email queries, we have two averages – *micro average* and *macro average*. For the micro average, we calculated MAP and mean P@1 across category borders. Our macro average was calculated from the seven precision values of Cat1–Cat7. Macro average reflects the average across topics, disregarding the number of documents per topic.

If an average value is shown with a *confidence interval*, the confidence level is 95%.

In order to demonstrate *MAP gain* by one method over another one, we subtracted the two MAP values. If the confidence intervals of the two MAP values did not overlap, the MAP gain was *statistically significant*. *Mean-P@1 gain* was calculated the same way.

Recall is the share of retrieved documents relevant to the query among all the relevant documents in the collection. Because precision was always measured for all the retrieved documents at maximum recall, we did not consider any precision-recall trade-off and did not measure the F-score.

5 Results of Ad-hoc Retrieval

Two rounds of tests were carried out. First, we matched email queries to email documents. After that, we tested retrieval of website documents according to the relevance criteria defined in Table 5.

5.1 Retrieval of Email Messages

The document collection was all the messages in Cat1–Cat7 and Rest; the queries were all the messages from Cat1–Cat7. We tested the text similarity calculation methods defined in Table 1, except TP. TP was excluded because an average email-email matching instance generated 2.76 pairwise term co-occurrences, too few for text similarity calculation based solely on term co-occurrences.

Table B1 in Appendix B shows all the MAP and P@1 values; Table 9 summarizes the winners. There is close competition between BMTP and SBERT, and they tend to outperform the other text similarity calculation methods.

Table 9. Winning precision values. Source: Table B1 in Appendix B

	Cat1	Cat2	Cat3	Cat4	Cat5	Cat6	Cat7	Micro avg	Macro avg
Recall	100	100	100	100	100	100	100	100	100
MAP	17.87	62.83	71.06	36.31	38.97	21.31	39.34	35.70	35.05
Method	BMTP	SBERT	BMTP	SBERT	SBERT	BMTP	SBERT	SBERT	BMTP
Mean P@1	51.03	85.52	86.19	61.49	72.45	47.92	66.10	67.39	63.39
Method	SBERT	SBERT	CSTP	SBERT	SBERT	BMTP	BMTP	SBERT	SBERT

Figure 1 shows the performance trends across email categories. One method – YANG – stands out as a poor performer. The method is ingenious for retrieval of microblog posts, on average 8.56 words long. Email messages, on average 25 words long (see Table 2), have a too large scope of co-occurrence for the YANG method.

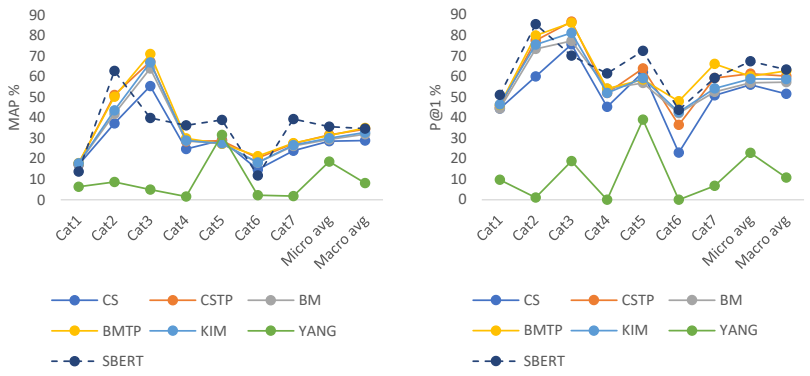

Figure 1. Overview of precision values per text similarity calculation method. Source: Table B1 in Appendix B

The other method that does not follow the trends is SBERT. With SBERT, two embeddings are similar if both email texts adhere to a common language model; the similarity is not overly focused on the presence of common subject-relevant words.

Table 10 shows precision gains. CSTP and BMTP perform almost always better than CS and BM. BMTP vs. SBERT, the two best-performing text similarity calculation methods, succeed and fail with large margins in different text categories, and all the MAP differences are statistically significant.

Table 10. Precision gains. Source: Table B1 in Appendix B. Statistically significant gains (confidence intervals of the two average-precision values do not overlap) have bold typeface.

	Cat1	Cat2	Cat3	Cat4	Cat5	Cat6	Cat7	Micro avg	Macro avg
MAP gain									
CSTP-CS	0.46	**13.81**	**11.88**	**3.79**	-0.33	**5.60**	3.48	**3.04**	5.53
BMTP-BM	0.16	**8.35**	**7.14**	1.15	0.01	**3.15**	1.37	**1.83**	3.05
BMTP- SBERT	**4.10**	**-12.55**	**31.15**	**-6.36**	**-11.69**	**9.45**	**-11.73**	**-4.41**	0.34
BMTP-KIM	0.11	**6.74**	4.18	1.10	-0.26	**3.05**	0.92	1.27	2.26
Mean-P@1 gain									
CSTP-CS	1.02	**17.59**	11.05	6.76	1.71	**13.54**	8.47	**5.43**	8.59
BMTP-BM	1.79	6.55	8.84	0.00	1.71	5.73	13.56	3.26	5.46
BMTP- SBERT	-4.88	-5.51	**16.02**	-7.44	**-13.93**	4.17	6.78	**-7.23**	-0.68
BMTP-KIM	-0.26	4.48	4.97	2.02	-0.62	5.21	11.86	1.30	3.96

Precision gains in Cat5 are dominated by negative numbers. The success of SBERT in Cat5 suggests that the relevance criterion of Cat5 prioritizes similar background stories without category-representative keywords.

5.2 Retrieval of Website Documents

We did two retrieval precision measurements: one with the profiles of pairwise term co-occurrence context TC1, TC3, and TC5, which may filter out some query-relevant documents and thus lower recall; the other one at 100% recall.

5.2.1 Retrieval at Close to 100% Recall

Table B2 in Appendix B shows retrieval precision from the entire web document collection. While term co-occurrences improve retrieval of brochures, cosine similarity remains superior for retrieval of web articles and fill-in forms. Table B2 demonstrates the bias of formula (6) towards large PDF documents (see Section 3.2.2), which are often non-relevant but overshadow much smaller relevant web articles and fill-in forms.

In order to lessen that bias, we tested the retrieval of web articles and fill-in forms from the reduced-size document collection without the large PDF documents. Table B3 in Appendix B shows the results. Table 11 summarizes the winners.

Table 11. Winning precision values for website document retrieval. Source: Table B2 and Table B3 in Appendix B. Precision values by a method that utilizes pairwise term co-occurrence have bold typeface. "M" stands for email messages.

Relevant documents	**Reduced-size document collection**						**Entire document collection**		
	A (web articles)			**F (fill-in forms)**			**B (brochures)**		
Query	M	A	F	M	A	F	M	A	F
Recall	90.26	99.09	100	84.80	100	98.53	97.30	100	100
MAP	**44.89**	51.46	**47.25**	**25.21**	23.06	55.90	**35.68**	**36.04**	**46.53**
Method	TP	CS	TP	BMTP	BM	BM	CSTP	TP	BMTP
Mean P@1	**40.74**	**51.22**	**48.78**	**12.75**	11.11	47.06	**19.54**	**34.38**	**36.67**
Method	TP	TP	TP	BMTP	BM	BM	CSTP	TP	BMTP

In Table B2 and Table B3 in Appendix B, we observe that KIM and BM have very close, often identical, precision values. KIM has a BM25 component and a unidirectional term association

component, where the latter is normalized by document length. With large documents, the normalization reduces the value of the unidirectional term association component towards zero, and KIM becomes BM.

5.2.2 Retrieval at 100% Recall

Table B4 in Appendix B shows the precision values; Table 12 summarizes the winners. The influence of pairwise term co-occurrence is similar to that in Table 11, except for F–A retrieval.

Table 12. Winning precision values for website document retrieval at 100% recall. Source: Table B4 in Appendix B. Precision values by a method that utilizes pairwise term co-occurrence have bold typeface. "M" stands for email messages.

Relevant documents	**Reduced-size document collection**						**Entire document collection**		
	A (web articles)			**F (fill-in forms)**			**B (brochures)**		
Query	M	A	F	M	A	F	M	A	F
Recall	100	100	100	100	100	100	100	100	100
MAP	**39.15**	51.06	28.59	**21.97**	23.04	63.71	**33.49**	**35.60**	**34.47**
Method	BMTP	CS	CS	BMTP	BM	BM	CSTP	BMTP	BMTP
Mean P@1	**32.07**	**51.22**	19.57	**11.34**	11.11	58.82	**17.94**	**29.69**	**16.67**
Method	BMTP	BMTP	SBERT	BMTP	BM	BM	CSTP	BMTP	BMTP

Figure 2 illustrates the precision trends:

- For brochures as relevant documents, the solid line of BMTP dominates the top of the pile, although CSTP is the top performer with email queries.
- For web articles as relevant documents, most curves form a compact cluster with similar precision values. CS and BMTP are the top performers, while SBERT has slightly better P@1 with fill-in form queries.
- For fill-in forms as relevant documents, the thin dotted lines of BM and CS – bag-of-words text similarity calculation methods – dominate the top of the pile; still, BMTP is best with email queries.

SBERT did not turn out to be a top performer, and there are several reasons for that:

- Brochures are large documents, and the mean pooling of paragraph embeddings does not produce a good document embedding.
- SBERT performs on par with the other text similarity calculation methods when articles are the relevant documents and performs worse than the others when fill-in forms are the relevant documents. SBERT has been trained on a large natural language corpus; semi-structured text requires a differently trained language model.
- SBERT did not repeat the success of email retrieval by email queries. While some email text categories grouped messages with similar wordings, web-document retrieval had individual query-document relevance criteria defined in Table 5, and those relevance criteria did not assume similar wordings.
- A language model is trained assuming a certain human language style. Transformer-based NLP is sensitive to nuances: "stopwords and punctuation, which are often ignored by traditional IR approaches, play a key role in understanding natural language queries by defining grammar structures and word dependencies" [63]. The colloquial language style of email messages apparently does not match the formal language style of website documents; therefore, SBERT has the lowest precision with email queries. Polignano et al. [80] were aware of the matter when they trained an Italian BERT model "from scratch on social network language, Twitter, in particular, because many of the classic tasks of content analysis are oriented to data extracted from the digital sphere of users".

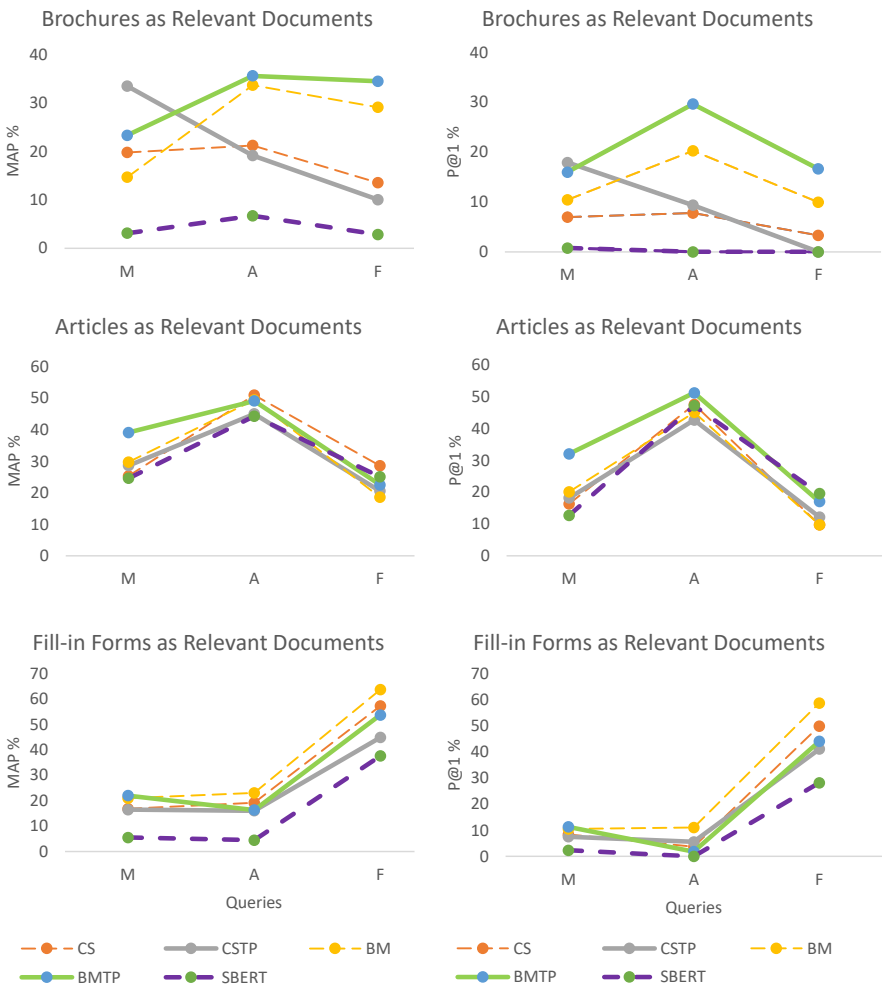

Figure 2. Overview of precision values per text similarity calculation method. Source: Table B4 in Appendix B.

5.2.3 Precision Gains

Table 13 demonstrates that pairwise term co-occurrence is certainly beneficial for M–A and M–B retrieval, which has only positive precision gains, often statistically significant. Retrieval of brochures has mostly positive precision gains. Retrieval of fill-in forms has mostly negative precision gains.

The main conclusion from Table 13 is that pairwise term co-occurrence is beneficial if (i) a query is considerably smaller than the relevant documents and (ii) the relevant documents tell stories as opposed to the structured "artificial" text of fill-in forms. "Considerably smaller" would be, on average, 32 (M–A), 39 (A–B), 92 (F–B), and 1259 (M–B) times smaller.

We would like to emphasize the difference between two notions of "influence of document size":

- Formula (6) is biased towards larger documents in the collection. In order to counteract this size influence, we have two test collections of website documents (see Table 3 and Table 4).
- Pairwise term co-occurrence improves text similarity score if a query is much smaller than the relevant documents. The size difference makes two bags of words unequal, and the meanings of the co-occurrences on top of the meanings of both terms compensate for this inequality, thus improving text-matching accuracy. We do not counteract this size influence; it yields a better text similarity score.

Bag of words still stands strong if (i) the relevant documents have structured text, e.g., fill-in forms; or (ii) the query and the document have equivalent sizes and rich vocabulary, e.g., both are webpages.

Table 13. Precision gains. Source: Table B2, Table B3, and Table B4 in Appendix B. Statistically significant gains have bold typeface. "M" stands for email messages.

Relevant documents	Reduced-size document collection						Entire document collection		
	A (web articles)			F (fill-in forms)			B (brochures)		
Query	M	A	F	M	A	F	M	A	F
MAP gain									
Recall	90.26	99.09	100	84.80	100	98.53	97.30	100	100
TP-CS	**14.18**	-2.43	10.53	0.87	-4.71	-11.58	**5.82**	**14.8**	**29.44**
TP-BM	**11.34**	-0.81	12.65	-3.21	-8.6	-22.8	**14.35**	2.36	10.83
Recall	100	100	100	100	100	100	100	100	100
CSTP-CS	3.35	-5.99	-8.00	-0.37	-3.12	-12.43	**13.73**	-2.06	-3.52
BMTP-BM	**9.39**	-0.35	3.95	1.03	-6.73	-10.03	**8.62**	1.92	5.36
Mean-P@1 gain									
Recall	90.26	99.09	100	84.80	100	98.53	97.30	100	100
TP-CS	**21.03**	3.66	24.39	1.47	-1.85	-11.76	4.79	**26.57**	**20.58**
TP-BM	**18.09**	6.10	**31.71**	-0.98	-9.26	-29.41	3.55	14.07	7.24
Recall	100	100	100	100	100	100	100	100	100
CSTP-CS	1.91	-4.88	2.44	-0.84	1.86	-8.82	**10.97**	1.57	-3.33
BMTP-BM	**11.96**	6.10	7.31	0.84	-9.26	-14.70	**5.58**	9.38	6.67

6 Key Findings

There is no universally best text similarity calculation method. Section 6.1 summarizes the cases where pairwise term co-occurrence is likely to perform better than the baselines defined in Table 1, whereas Section 6.2 summarizes the design principles of text similarity calculation methods that use pairwise term co-occurrence.

6.1 Applicability of Pairwise Term Co-occurrence in Text Similarity Calculation

So far, we have discovered two situations where pairwise term co-occurrence is expected to be beneficial:

- Both the query and the documents are compact yet informative pieces of text, such as email messages (see Section 5.1). With compact queries and documents, small bags of words may not have enough terms for a conclusive text similarity score, and pairwise term co-occurrence weights give the score a boost in the right direction. MAP and P@1 gains were as high as 5.53 and 8.59 percentage points, respectively (see Table 10).
  - In email retrieval by email queries, BMTP and SBERT have performed on par but had large precision differences for different relevance criteria. SBERT is likely to perform better if the relevance criteria rely on the same overall story and detailed relationships between all the words. BMTP is likely to perform better if the overall stories vary but refer to subject-relevant words, which enables subject-relevant term co-occurrences.
- The query is much smaller than the relevant documents in the collection. The size difference makes two bags of words unequal, and pairwise term co-occurrence steps in to compensate for this inequality (see Section 5.2.3). MAP and P@1 gains were as high as 29.44 and 31.71 percentage points, respectively (see Table 13).

In two situations, pairwise term co-occurrence is not likely to be beneficial:

- The queries and the relevant documents are larger than the aforementioned compact pieces of text and have equivalent sizes. Their bags of words are sufficient for text similarity calculation; pairwise term co-occurrence here is more of a distraction than a help.
- The relevant documents in the collection lack natural language flow, e.g., fill-in forms have captions as the major information carriers.

6.2 Design and Implementation Principles of Pairwise Term Co-occurrence in Text Similarity Calculation

While doing this research, we came across and tested a few design and implementation principles for calculating text similarity enhanced by pairwise term co-occurrence.

Half of the co-occurrence weight. If unigram-based text similarity calculation is enhanced by pairwise term co-occurrences in monomials such as $w(t_1)\cdot w(\{t_1,t_2\})$ and $w(t_2)\cdot w(\{t_1,t_2\})$, the monomials should be $w(t_1)\cdot(0.5\cdot w(\{t_1,t_2\}))$ and $w(t_2)\cdot(0.5\cdot w(\{t_1,t_2\}))$ instead (see Section 3.3.2).

Normalization of term co-occurrence weights. While normalization of term frequency by document size is common in unigram language models, normalization failed when we tried to normalize our term co-occurrence weights by document size or an equivalent score if document sizes in the collection were large or different (see Section 3.2.2). We believe that normalization of term co-occurrence weights by a single document-specific score is likely to be counter-productive.

Influence of in-document vs. collection-wide term features. Some previous research has applied in-document term proximity features to modify in-document term frequency in BM25 and the language modeling text-matching framework. In our experiments, modifying $tf_d(t)$ in formula (10) by term co-occurrence weights did not work. We argue that collection-wide term co-occurrence features, such as term loyalty and ICF, should have the same level of influence on term weights as IDF has (see Section 3.2.2).

Scope of pairwise term co-occurrence. Text similarity formulas do not define the scope of pairwise term co-occurrence. In Section 4.3, we have defined the scopes of pairwise term co-occurrence for different queries – email messages, web articles, and fill-in forms; the scopes were established experimentally. The scope of pairwise term co-occurrence influences precision; therefore, choosing the right scope matters.

Duplicate pieces of text. During text pre-processing, we removed non-relevant duplicate pieces of text, such as greetings and farewell pleasantries in email messages, as well as headers, footers, and navigation aids in web articles. If a query and a document contain the same non-relevant piece of text, that piece contaminates the bag of word pairs at the rate $n\cdot(n\text{-}1)/2$ pairs per n-word piece of text. Because of the same reason, plagiarized text, citations, and duplicate text generated by a content management system, present in both the query and the document, may over-dominate the text similarity score.

7 Conclusions

Our research demonstrates that a lightweight text similarity calculation method, equivalent to cosine similarity and BM25 and augmented by pairwise term co-occurrences, can perform on par with an off-the-shelf language model not fine-tuned for the particular text collection.

We do not claim that pairwise term co-occurrences would outperform a fine-tuned language model or an LLM with sufficient domain knowledge. Rather, the advantage of our text similarity calculation methods *depends on the use case*. Two types of use cases – public organizations, local governments in particular, and the bottom of the technology stack – have been introduced in Section 1.

7.1 Novelty, Contributions, Limitations

The *novelty* of this work revolves around the development of a method for measuring the strength of pairwise term co-occurrence and using this strength to modify the importance of a term (see Section 3.2.2). The method is analogous to a contextual word embedding, except in our case the context of a term modifies the importance, not the meaning of the term.

The *contributions* of this research are:

- Text similarity calculation formulas (8), (13), and (14) together with (16).
- Applicability criteria for pairwise term co-occurrence in text similarity calculation, stated in Section 6.1.

- Testing text similarity calculation by pairwise term co-occurrence vs. an off-the-shelf language model not fine-tuned for the text collection. While both perform on par for email retrieval by email queries, the mean pooling of paragraph embeddings for long documents made the performance of the language model poor. Also poor was the performance of the language model for retrieval of structured texts of fill-in forms.
- Design and implementation principles for text similarity calculation enhanced by pairwise term co-occurrence; the principles are listed in Section 6.2.
- Unique data set of text types representative of public organizations: email messages, web articles, fill-in forms, and brochures. We have contributed labeled data for measuring retrieval precision that is difficult to obtain and is time-consuming and expensive to develop. Starting out with some labeled data (categorized email messages), we established new relevance links between email messages, web articles, fill-in forms, and brochures (see Section 4.2). The variety of the text types has made it possible to establish the applicability criteria stated in Section 6.1. We will be happy to share our data with interested researchers.

We have tested our text similarity calculation methods using a variety of public-organization text types. This is a stronghold and also a *limitation* of the work. It is the stronghold because the use case of public organizations has been poorly addressed by the IR research community (since the rise of the large web-based and social-media-based text collections), which opens a niche for us. It is also a limitation because we are not aware of similar collections of labeled data representative for public organizations, which leads to difficulty in better verifying the use case.

7.2 Future Research

Although we have tested formulas (8), (13), and (14)+(16) for retrieval of documents representative of public organizations, we believe that the formulas are applicable to other kinds of documents as well as long as (i) text is the main information carrier and (ii) the requirements to the queries and the documents stated in Section 6.1 are fulfilled. Nevertheless, the other kinds of documents remain to be tested.

Text matching using terms, not the meanings of the terms modeled by word embeddings, is "old school". We did experiment with clustered BERT word vectors as latent concepts; however, the concepts did not work out. With narrow clusters, each term instance in the text became its own concept that did not match any other instance of the same term, whereas wider clusters encompassed loosely related terms and made concept co-occurrences unusable. Timkey and van Schijndel [81] have explored this kind of problem and found that the culprits are one to five "rogue dimensions". Although these dimensions are not relevant for human-judged word similarity, they dominate vector representation of contextual word embeddings and interfere with cosine similarity that measures the semantic similarity of two words. For us, identifying and correcting those "rogue dimensions" would enable pairwise concept co-occurrence instead of term co-occurrence.

Because formula (6) is biased towards larger documents (see Section 3.2.2), passage retrieval is an obvious next step for applying pairwise term co-occurrence in similarity calculation for large texts. Passage retrieval "can improve the accuracy of document retrieval when documents are long or span different subject areas" [82]. Furthermore, we can aggregate passage-level query-document relevance representations into a document-level representation before calculating query-document similarity [64].

BM25 has been used in passage retrieval to provide input to BERT and LLM re-rankers ([83], [84], [85]). Instead of BM25, formula (13) could be used in order to do the initial retrieval and filtering of the passages. So far, we have tested formula (13) with email-style messages. It remains to be tested with text passages in various text re-ranking tasks.

References

[1] T. Mikolov, I Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," *Advances in Neural Information Processing Systems*, vol. 26, 2013.

[2] J. Devlin, M. W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," *arXiv:1810.04805,* 2018. Available: https://doi.org/10.48550/arXiv.1810.04805

[3] R. Gozalo-Brizuela and E. C. Garrido-Merchan, "ChatGPT is not all you need. A State-of-the-Art Review of large Generative AI models," *arXiv:2301.04655,* 2023. Available: https://doi.org/10.48550/arXiv.2301.04655

[4] S. R. Bowman, "Eight things to know about large language models," *arXiv:2304.00612*, 2023. Available: https://doi.org/10.48550/arXiv.2304.00612

[5] Y. Tan, D. Min, Y. Li, W. Li, N. Hu, Y. Chen, and G. Qi, "Evaluation of ChatGPT as a question answering system for answering complex questions," *arXiv:2303.07992*, 2023. Available: https://doi.org/10.48550/arXiv.2303.07992

[6] A. Choudhury and H. Shamszare, "Investigating the Impact of User Trust on the Adoption and Use of ChatGPT: Survey Analysis," *Journal of Medical Internet Research*, vol. 25, e47184, 2023. Available: https://doi.org/10.2196/47184

[7] W. Yang, H. Zhang, and J. Lin, "Simple applications of BERT for ad hoc document retrieval," *arXiv:1903.10972*, 2019. Available: https://doi.org/10.48550/arXiv.1903.10972

[8] A. Dhar, H. Mukherjee, N. S. Dash, and K. Roy, "Text categorization: past and present," *Artificial Intelligence Review*, vol. 54, pp. 3007–3054, 2020. Available: https://doi.org/10.1007/s10462-020-09919-1

[9] Q. Yaseen, "Spam email detection using deep learning techniques," *Procedia Computer Science*, vol. 184, pp. 853–858, 2021. Available: https://doi.org/10.1016/j.procs.2021.03.107

[10] W. Jiang, N. Synovic, M. Hyatt, T. R. Schorlemmer, K. Läufer, Y. Lu, G. K. Thiruvathukal, and J. C. Davis, "An Empirical Study of Pre-Trained Model Reuse in the Hugging Face Deep Learning Model Registry," in *2023 IEEE/ACM 45th International Conference on Software Engineering (ICSE)*, pp. 2463–2475, 2023. Available: https://doi.org/10.1109/ICSE48619.2023.00206

[11] T. Y. S. S. Santosh, R. Haddad, and M. Grabmair, "ECtHR-PCR: A Dataset for Precedent Understanding and Prior Case Retrieval in the European Court of Human Rights," in *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pp. 5473–5483, 2024. Available: https://aclanthology.org/2024.lrec-main.486

[12] B. B. Sookman, "Moffatt v. Air Canada: A Misrepresentation by an AI Chatbot," 2024. Available: https://www.mccarthy.ca/en/insights/blogs/techlex/moffatt-v-air-canada-misrepresentation-ai-chatbot. Accessed on May 17, 2024.

[13] R. Liu, J. Wei, F. Liu, C. Si, Y. Zhang, J. Rao, S. Zheng, D. Peng, D. Yang, D. Zhou, and A. M. Dai, "Best Practices and Lessons Learned on Synthetic Data for Language Models," *arXiv:2404.07503*, 2024. Available: https://doi.org/10.48550/arXiv.2404.07503

[14] C. Ling, X. Zhao, J. Lu, C. Deng, C. Zheng, J. Wang, T. Chowdhury, H. Cui, X. Zhang, T. Zhao, A. Panalkar, C. Wei, H. Wang, Y. Liu, Z. Chen, H. Chen, C. White, Q. Gu, J. Pei, C. Yang, and L. Zhao, "Domain Specialization as the Key to Make Large Language Models Disruptive: A Comprehensive Survey," *arXiv:2305.18703v7*, 2024. Available: https://doi.org/10.48550/arXiv.2305.18703

[15] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W. T. Yih, T. Rocktäschel, S. Riedel, and D. Kiela, D. "Retrieval-augmented generation for knowledge-intensive NLP tasks," in *NIPS'20: Proceedings of the 34th International Conference on Neural Information Processing Systems*, pp. 9459–9474, 2020.

[16] G. Xexéo, F. Braida, M. Parreiras, and P. Xavier, "The Economic Implications of Large Language Model Selection on Earnings and Return on Investment: A Decision Theoretic Model," *arXiv:2405.17637*, 2024. Available: https://doi.org/10.48550/arXiv.2405.17637

[17] J. Han, M. Kamber, and J. Pei, *Data Mining. Concepts and Techniques*. Third Edition, Morgan Kaufmann, 2012.

[18] C. Kamphuis, A. P. de Vries, L. Boytsov, and J. Lin, "Which BM25 do you mean? A large-scale reproducibility study of scoring variants," in *Advances in Information Retrieval. ECIR 2020. Lecture Notes in Computer Science*, vol. 12036, pp. 28–34, 2020. Available: https://doi.org/10.1007/978-3-030-45442-5_4

[19] SCB. Län och kommuner, 2024 (in Swedish). Available: https://www.scb.se/hitta-statistik/regional-statistik-och-kartor/regionala-indelningar/lan-och-kommuner/. Accessed on May 17, 2024.

[20] Statskontoret. Myndigheterna under regeringen, 2024 (in Swedish). Available: https://www.statskontoret.se/fokusomraden/fakta-om-statsforvaltningen/myndigheterna-under-regeringen/. Accessed on May 17, 2024.

[21] A. Smaldone and M. L. J. Wright, Local Governments in the U.S.: A Breakdown by Number and Type, Federal Reserve Bank of St. Louis, 2024. Available: https://www.stlouisfed.org/publications/regional-economist/2024/march/local-governments-us-number-type. Accessed on May 17, 2024.

[22] V. Karpukhin, B. Oguz, S. Min, P. A. Lewis, L. Wu, S. Edunov, D. Chen, and W. Yih, "Dense Passage Retrieval for Open-Domain Question Answering," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 6769–6781, 2020. Available: https://doi.org/10.18653/v1/2020.emnlp-main.550

[23] C. J. Van Rijsbergen, "A theoretical basis for the use of co-occurrence data in information retrieval," *Journal of Documentation*, vol. 33, no. 2, pp. 106–119, 1977. Available: https://doi.org/10.1108/eb026637

[24] H. Guo, L. Z. Zhou, and L. Feng, "Self-Switching Classification Framework for Titled Documents," *Journal of Computer Science and Technology*, vol. 24, pp. 615–625, 2009. Available: https://doi.org/10.1007/s11390-009-9262-z

[25] K. Soumya George and S. Joseph, "Text classification by augmenting bag of words (BOW) representation with co-occurrence feature," *IOSR Journal of Computer Engineering*, vol. 16, no. 1, pp. 34–38, 2014. Available: https://doi.org/10.9790/0661-16153438

[26] F. Figueiredo, L. Rocha, T. Couto, T. Salles, M. A. Gonçalves, and W. Meira Jr, "Word co-occurrence features for text classification," *Information Systems*, vol. 36, no. 5, pp. 843–858. Available: https://doi.org/10.1016/j.is.2011.02.002

[27] S Yang, G. Huang, and B. Ofoghi, "Short Text Similarity Measurement Using Context from Bag of Word Pairs and Word Co-occurrence," in *Data Science. ICDS 2019. Communications in Computer and Information Science*, vol. 1179, pp. 221–231, 2020. Available: https://doi.org/10.1007/978-981-15-2810-1_22

[28] M. Kaiser, R. Saha Roy, and G. Weikum, "Conversational question answering over passages by leveraging word proximity networks," in *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 2129–2132, 2020. Available: https://doi.org/10.1145/3397271.3401399

[29] H. S. Kim, I. Choi, and M. Kim, "Refining term weights of documents using term dependencies," in *Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 552–553, 2004. Available: https://doi.org/10.1145/1008992.1009116

[30] L. Shi and J. Y. Nie, "Using various term dependencies according to their utilities," in *Proceedings of the 19th ACM International Conference on Information and Knowledge Management*, pp. 1493–1496, 2010. Available: https://doi.org/10.1145/1871437.1871655

[31] J. Gao, J. Y. Nie, G. Wu, and G. Cao, "Dependence language model for information retrieval," in *Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 170–177, 2004. Available: https://doi.org/10.1145/1008992.1009024

[32] G. Mishne and M. De Rijke, "Boosting web retrieval through query operations," in *Advances in Information Retrieval. ECIR 2005. Lecture Notes in Computer Science*, vol. 3408, pp. 502–516, 2005. Available: https://doi.org/10.1007/978-3-540-31865-1_36

[33] Y. Rasolofo and J. Savoy, "Term proximity scoring for keyword-based retrieval systems," in *Advances in Information Retrieval. ECIR 2003. Lecture Notes in Computer Science*, vol. 2633, pp. 207–218, 2003. Available: https://doi.org/10.1007/3-540-36618-0_15

[34] S. Büttcher, C. L. Clarke, and B. Lushman, "Term proximity scoring for ad-hoc retrieval on very large text collections," in *Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 621–622, 2006. Available: https://doi.org/10.1145/1148170.1148285

[35] R. Schenkel, A. Broschart, S. Hwang, M. Theobald, and G. Weikum, "Efficient text proximity search," in *String Processing and Information Retrieval. SPIRE 2007. Lecture Notes in Computer Science*, vol. 4726, pp. 287–299, 2007. Available: https://doi.org/10.1007/978-3-540-75530-2_26

[36] K. M. Svore, P. H. Kanani, and N. Khan, "How good is a span of terms? Exploiting proximity to improve web retrieval," in *Proceedings of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 154–161, 2010. Available: https://doi.org/10.1145/1835449.1835477

[37] X. Lu, A. Moffat, and J. S. Culpepper, “Efficient and effective higher order proximity modeling,” in *Proceedings of the 2016 ACM International Conference on the Theory of Information Retrieval*, pp. 21–30, 2016. Available: https://doi.org/10.1145/2970398.2970404

[38] T. Tao and C. Zhai, “An exploration of proximity measures in information retrieval,” in *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 295–302, 2007. Available: https://doi.org/10.1145/1277741.1277794

[39] J. Zhao, J. X. Huang, and Z. Ye, “Modeling term associations for probabilistic information retrieval,” *ACM Transactions on Information Systems*, vol. 32, no. 2, pp. 1–47, 2014. Available: https://doi.org/10.1145/2590988

[40] D. Metzler and W. B. Croft, “A Markov random field model for term dependencies,” in *Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 472–479, 2005. Available: https://doi.org/10.1145/1076034.1076115

[41] S. Liu, F. Liu, C. Yu, and W. Meng, “An effective approach to document retrieval via utilizing WordNet and recognizing phrases,” in *Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 266–272, 2004. Available: https://doi.org/10.1145/1008992.1009039

[42] F. Jian, J. X. Huang, J. Zhao, T. He, and P. Hu, “A simple enhancement for ad-hoc information retrieval via topic modelling,” in *Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 733–736, 2016. Available: https://doi.org/10.1145/2911451.2914748

[43] R. Song, M. J. Taylor, J. R. Wen, H. W. Hon, and Y. Yu, “Viewing term proximity from a different perspective,” in *Advances in Information Retrieval. ECIR 2008. Lecture Notes in Computer Science*, vol. 4956, pp. 346–357, 2008. Available: https://doi.org/10.1007/978-3-540-78646-7_32

[44] O. Vechtomova and M. Karamuftuoglu, “Lexical cohesion and term proximity in document ranking,” *Information Processing & Management*, vol. 44, no. 4, pp. 1485–1502, 2008. Available: https://doi.org/10.1016/j.ipm.2008.01.003

[45] J. Zhao and Y. Yun, “A proximity language model for information retrieval,” in *Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 291–298, 2009. Available: https://doi.org/10.1145/1571941.1571993

[46] R. Mihalcea and P. Tarau, “TextRank: Bringing order into text,” in *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, Association for Computational Linguistics, pp. 404–411, 2004.

[47] S. Brin and L. Page, “The anatomy of a large-scale hypertextual web search engine,” *Computer Networks and ISDN Systems*, vol. 30, no. 1–7, pp. 107–117, 1998. Available: https://doi.org/10.1016/S0169-7552(98)00110-X

[48] R. Blanco and C. Lioma, “Graph-based term weighting for information retrieval,” *Information Retrieval*, vol. 15, pp. 54–92, 2012. Available: https://doi.org/10.1007/s10791-011-9172-x

[49] W. Lu, Q. Cheng, and C. Lioma, “Fixed versus dynamic co-occurrence windows in TextRank term weights for information retrieval,” in *Proceedings of the 35th international ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 1079–1080, 2012. Available: https://doi.org/10.1145/2348283.2348478

[50] S. Bubeck, V. Chandrasekaran, R. Eldan, J. Gehrke, E. Horvitz, E. Kamar, P. Lee, Y. T. Lee, Y. Li, S. Lundberg, H. Nori, H. Palangi, M. T. Ribeiro, Y. Zhang, “Sparks of Artificial General Intelligence: Early experiments with GPT-4,” *arXiv:2303.12712*, 2023. Available: https://doi.org/10.48550/arXiv.2303.12712

[51] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, “Attention is all you need,” *Advances in Neural Information Processing Systems*, vol. 30, pp. 5998–6008, 2017.

[52] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan et al., “Language models are few-shot learners,” *Advances in Neural Information Processing Systems*, vol. 33, pp. 1877–1901, 2020.

[53] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen, “LoRA: Low-Rank Adaptation of Large Language Models,” *arXiv:2106.09685*, 2021. Available: https://doi.org/10.48550/arXiv.2106.09685

[54] T. Dettmers, A. Pagnoni, A. Holtzman, and L. Zettlemoyer, “QLoRA: Efficient finetuning of quantized LLMs,” *arXiv:2305.14314*, 2023. Available: https://doi.org/10.48550/arXiv.2305.14314

[55] J. Wei, X. Wang, D. Schuurmans, M. Bosma, B. Ichter, F. Xia, E. Chi, Q. V. Le, and D. Zhou, “Chain-of-Thought Prompting Elicits Reasoning in Large Language Models,” *Advances in Neural Information Processing Systems*, vol. 35, pp. 24824–24837, 2022.

[56] S. Yao, J. Zhao, D. Yu, N. Du, I. Shafran, K. Narasimhan, and Y. Cao, “ReAct: Synergizing reasoning and acting in language models,” *arXiv:2210.03629*, 2022. Available: https://doi.org/10.48550/arXiv.2210.03629

[57] A. Chowdhery, S. Narang, J. Devlin, M. Bosma, G. Mishra, A. Roberts et al., “PaLM: Scaling language modeling with pathways,” *arXiv:2204.02311*, 2022. Available: https://doi.org/10.48550/arXiv.2204.02311

[58] L. Wang, C. Ma, X. Feng, Z. Zhang, H. Yang, J. Zhang, et al., “A survey on large language model based autonomous agents,” *Frontiers of Computer Science*, vol. 18, 2024. Available: https://doi.org/10.1007/s11704-024-40231-1

[59] N. Reimers and I. Gurevych, “Sentence-BERT: Sentence embeddings using Siamese BERT-networks,” in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 3982–3992, 2019. Available: https://doi.org/10.18653/v1/D19-1410

[60] Y. Qiao, C. Xiong, Z. Liu, and Z. Liu, “Understanding the Behaviors of BERT in Ranking. *arXiv:1904.07531* 2019. Available: https://doi.org/10.48550/arXiv.1904.07531

[61] R. Nogueira and K. Cho, “Passage Re-ranking with BERT,” *arXiv:1901.04085*, 2019. Available: https://doi.org/10.48550/arXiv.1901.04085

[62] O. Khattab and M. Zaharia, M. “ColBERT: Efficient and effective passage search via contextualized late interaction over BERT,” in *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 39–48, 2020. Available: https://doi.org/10.1145/3397271.3401075

[63] Z. Dai and J. Callan, “Deeper text understanding for IR with contextual neural language modeling,” in *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 985–988, 2019. Available: https://doi.org/10.1145/3331184.3331303

[64] C. Li, A. Yates, S. MacAvaney, B. He, and Y. Sun, “PARADE: Passage representation aggregation for document reranking. *ACM Transactions on Information Systems*, vol. 42, no. 2, pp. 1–26, 2023. Available: https://doi.org/10.1145/3600088

[65] Y. Matsuo and M. Ishizuka, “Keyword extraction from a single document using word co-occurrence statistical information,” *International Journal on Artificial Intelligence Tools*, vol. 13, no. 1, pp. 157–169, 2004. Available: https://doi.org/10.1142/S0218213004001466

[66] X. Song, Y. Rui, and X. Hu, “Pairwise topic model and its application to topic transition and evolution,” in *2016 IEEE International Conference on Big Data (Big Data)*, IEEE, pp. 86–95, 2016. Available: https://doi.org/10.1109/BigData.2016.7840592

[67] L. Chen, K. K. Chin, and K. Knill, “Improved language modelling using bag of word pairs,” in *Interspeech 2009*, pp. 2671–2674, 2009. Available: https://doi.org/10.21437/Interspeech.2009-121

[68] M. Shirakawa, T. Hara, and S. Nishio, “IDF for Word N-grams,” *ACM Transactions on Information Systems*, vol. 36, no. 1, pp. 1–38, 2017. Available: https://doi.org/10.1145/3052775

[69] C. D. Manning, P. Raghavan, and H. Schütze, *Introduction to Information Retrieval*. Cambridge University Press, 2008. Available: https://doi.org/10.1017/CBO9780511809071

[70] B. He, J. X. Huang, and X. Zhou, “Modeling term proximity for probabilistic information retrieval models,” *Information Sciences*, vol. 181, no. 14, pp. 3017–3031, 2011. Available: https://doi.org/10.1016/j.ins.2011.03.007

[71] Y. Lv and C. Zhai, “When documents are very long, BM25 fails!” in *Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 1103–1104, 2011. Available: https://doi.org/10.1145/2009916.2010070

[72] S. E. Robertson, S. Walker, M. M. Beaulieu, M. Gatford, and A. Payne, “Okapi at TREC-4,” *Nist Special Publication*, pp. 73–96, 1996.

[73] T. Khatoon and A. Govardhan, “Query Expansion with Enhanced-BM25 Approach for Improving the Search Query Performance on Clustered Biomedical Literature Retrieval,” *Journal of Digital Information Management*, vol. 16, no. 2, pp. 85–98, 2018.

[74] A. Lipani, M. Lupu, A. Hanbury, and A. Aizawa, “Verboseness fission for BM25 document length normalization,” in *Proceedings of the 2015 International Conference on the Theory of Information Retrieval*, pp. 385–388, 2015. Available: https://doi.org/10.1145/2808194.2809486

[75] A. Sordoni, J. Y. Nie, and Y. Bengio, “Modeling term dependencies with quantum language models for IR,” in *Proceedings of the 36th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 653–662, 2013. Available: https://doi.org/10.1145/2484028.2484098

[76] Pretrained Models. SBERT.net, 2024. Available: https://sbert.net/docs/sentence_transformer/pretrained_models.html. Accessed on June 6, 2024.

[77] H. Dalianis, J. Sjöbergh, and E. Sneiders, "Comparing manual text patterns and machine learning for classification of e-mails for automatic answering by a government agency," in *Computational Linguistics and Intelligent Text Processing, CICLing 2011. Lecture Notes in Computer Science*, vol. 6609, pp. 234–243, 2011. Available: https://doi.org/10.1007/978-3-642-19437-5_19

[78] E. Sneiders, J. Sjöbergh, and A. Alfalahi, "Automated email answering by text-pattern matching: Performance and error analysis," *Expert Systems*, vol. 35, no. 1, 2018. Available: https://doi.org/10.1111/exsy.12251

[79] F. Rekathati, "Introducing a Swedish Sentence Transformer," *The KBLab Blog*, 2021. Available: https://kb-labb.github.io/posts/2021-08-23-a-swedish-sentence-transformer/. Accessed on June 6, 2024.

[80] M. Polignano, V. Basile, P. Basile, M. de Gemmis, and G. Semeraro, "ALBERTo: Modeling Italian Social Media Language with BERT," *IJCoL, Italian Journal of Computational Linguistics*, no. 5–2, pp. 11–31, 2019. Available: https://doi.org/10.4000/ijcol.472

[81] W. Timkey and M. van Schijndel, "All bark and no bite: Rogue dimensions in transformer language models obscure representational quality," in *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 4527–4546. Available: https://doi.org/10.18653/v1/2021.emnlp-main.372

[82] M. Wang and L. Si, "Discriminative probabilistic models for passage based retrieval," in *Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 419–426, 2008. Available: https://doi.org/10.1145/1390334.1390407

[83] S. Wang, S. Zhuang, and G. Zuccon, "BERT-based dense retrievers require interpolation with BM25 for effective passage retrieval," in *Proceedings of the 2021 ACM SIGIR International Conference on Theory of Information Retrieval*, pp. 317–324, 2021. Available: https://doi.org/10.1145/3471158.3472233

[84] A. Drozdov, H. Zhuang, Z. Dai, Z. Qin, R. Rahimi, X. Wang, D. Alon, M. Iyyer, A. McCallum, D. Metzler, and K. Hui, "PaRaDe: Passage Ranking using Demonstrations with LLMs," in *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 14242–14252, Available: https://doi.org/10.18653/v1/2023.findings-emnlp.950

[85] R. Pradeep, S. Sharifymoghaddam, and J. Lin, "RankVicuna: Zero-shot listwise document reranking with open-source large language models," *arXiv:2309.15088*, 2023. Available: https://doi.org/10.48550/arXiv.2309.15088

Appendix A. Stop Words

Following are the stop words, translated from Swedish to English, used in our experiments in a closed domain of Swedish social insurance. Although the experiments were conducted in a closed domain, the stop words are marginally domain specific.

a, about, above, according to, after, again, against, all, almost, already, also, always, among, and, anymore, anything, anyway, approximately, as soon as possible, as well, as well as, at, at least, be, because, because of, become, before, behind, believe, besides, between, big, both, but, by, bye, can, check (verb), close, different, directly, do, down, easily, easy, either, enough, even, every, everything, excuse me, far from, finally, following, for, for a while, from, go, good, good morning, great, happen, hardly, have, he, hello, hence, her, here, him, his, hope, hopefully, I, I mean, immediately, important, in, in comparison to, in front of, in the same time, instead, it, just, kind regards, know, late, like, little, long ago, many, maybe, me, more, mostly, much, must, my, my name is, necessary, need, neither, never, nevertheless, next, nice, nicely, no, not, now, of, of course, often, on, one (meaning a person), only, or, other, others, otherwise, ought to, our, out, outside, over, own, per, perhaps, please, possibly, previous, quite, rather, really, regarding, same, say, see, shall, she, simple, simply, since, sincerely, so, some, something, sometimes, sorry, still, such, such as, super, Sweden, Swedish Social Insurance Agency, take, than, thank you, thankful, that, the, their, theirs, them, then, there, therefore, these, they, this, those, through, to, today, together, tomorrow, under, unfortunately, until, up, us, want, we, very, what, where, wherefrom, which, while, whilst, who, whose, why, via, wish, with, within, without, wonder (would like to know), yes, yesterday, you, your, you’re welcome, yours

Appendix B. Retrieval Precision

The following tables show precision measurements as defined in Section 4.5 of the article. The acronyms of the text similarity calculation methods are defined in Table 1 of the article. The data sets are explained in Section 4.1. The profiles of pairwise term co-occurrence context are defined in Section 4.3. The profiles of SBERT text segmentation are defined in Section 4.4 of the article. "M" in Tables B2, B3, and B4 stands for email messages.

Table B1. Precision of email retrieval by email queries. Profile of pairwise term co-occurrence context is TC2; profile of text segmentation for SBERT is TC7.

	Cat1	**Cat2**	**Cat3**	**Cat4**	**Cat5**	**Cat6**	**Cat7**	**Micro avg**		**Macro avg**
Recall	100	100	100	100	100	100	100	100		100
MAP										
CS	17.01 ± 0.84	37.23 ± 1.29	55.45 ± 2.79	24.73 ± 1.60	29.12 ± 0.44	14.74 ± 0.96	23.91 ± 3.82	28.60 ± 0.54		28.88
CSTP	17.47 ± 1.02	51.04 ± 1.45	67.33 ± 3.21	28.52 ± 1.92	28.79 ± 0.46	20.34 ± 1.60	27.39 ± 3.88	31.64 ± 0.68		34.41
BM	17.71 ± 0.96	41.93 ± 1.61	63.92 ± 3.33	28.80 ± 1.90	27.27 ± 0.43	18.16 ± 1.26	26.24 ± 4.15	29.46 ± 0.63		32.00
BMTP	**17.87** ± 1.06	50.28 ± 1.59	**71.06** ± 3.32	29.95 ± 1.97	27.28 ± 0.44	**21.31** ± 1.64	27.61 ± 3.88	31.29 ± 0.71		**35.05**
KIM	17.76 ± 0.96	43.54 ± 1.64	66.88 ± 3.28	28.85 ± 1.91	27.54 ± 0.43	18.26 ± 1.27	26.69 ± 4.16	30.02 ± 0.65		32.79
YANG	6.38 ± 0.09	8.72 ± 0.15	4.97 ± 0.29	1.66 ± 0.05	31.65 ± 0.14	2.22 ± 0.04	1.82 ± 0.92	18.61 ± 0.52		8.20
SBERT	13.77 ± 0.70	**62.83 ± 2.15**	39.91 ± 3.98	**36.31 ± 3.02**	**38.97 ± 0.64**	11.86 ± 1.02	**39.34 ± 4.82**	**35.70 ± 0.79**		34.71
Mean P@1										
CS	44.36 ± 4.93	60.00 ± 5.64	75.69 ± 6.25	45.27 ± 8.02	62.18 ± 2.65	22.92 ± 5.95	50.85 ± 12.76	55.95 ± 1.93		51.61
CSTP	45.38 ± 4.94	77.59 ± 4.80	**86.74** ± 4.94	52.03 ± 8.05	63.89 ± 2.63	36.46 ± 6.81	59.32 ± 12.53	61.38 ± 1.89		60.20
BM	44.36 ± 4.93	73.45 ± 5.08	77.35 ± 6.10	54.05 ± 8.03	56.81 ± 2.71	42.19 ± 6.99	52.54 ± 12.74	56.90 ± 1.92		57.25
BMTP	46.15 ± 4.95	80.00 ± 4.60	86.19 ± 5.03	54.05 ± 8.03	58.52 ± 2.69	**47.92** ± 7.07	**66.10** ± 12.08	60.16 ± 1.90		62.71
KIM	46.41 ± 4.95	75.52 ± 4.95	81.22 ± 5.69	52.03 ± 8.05	59.14 ± 2.69	42.71 ± 7.00	54.24 ± 12.71	58.86 ± 1.91		58.75
YANG	9.74 ± 2.94	1.03 ± 1.16	18.78 ± 5.69	0.00 ± 0.00	38.99 ± 2.67	0.00 ± 0.00	6.78 ± 6.41	22.79 ± 1.63		10.76
SBERT	**51.03 ± 4.97**	**85.52 ± 4.06**	70.17 ± 6.68	**61.49 ± 7.87**	**72.45 ± 2.44**	43.75 ± 7.04	59.32 ± 12.64	**67.39 ± 1.82**		**63.39**

Table B2. Precision of website document retrieval from the entire document collection

Relevant documents	**Entire document collection**								
	A (web articles)			**F (fill-in forms)**			**B (brochures)**		
Query	M	A	F	M	A	F	M	A	F
Recall	90.26 ± 1.99	99.09 ± 1.07	100 ± 0.00	84.80 ± 4.49	100 ± 0.00	98.53 ± 2.84	97.30 ± 1.22	100 ± 0.00	100 ± 0.00
Profile	TC1	TC3	TC5	TC1	TC3	TC5	TC1	TC3	TC5
MAP									
TP	9.21 ± 1.47	10.29 ± 3.68	3.69 ± 1.82	7.72 ± 3.00	1.08 ± 0.42	1.69 ± 0.78	29.99 ± 1.94	**36.04** ± 7.67	44.99 ± 14.29
CS	**18.76** ± 2.06	**40.73** ± 7.03	**29.80** ± 7.45	**17.62** ± 3.73	**12.54** ± 3.42	**45.76** ± 12.69	24.17 ± 1.58	21.24 ± 5.53	15.55 ± 6.97
CSTP	17.11 ± 2.02	33.75 ± 6.69	23.64 ± 7.53	13.52 ± 3.43	7.20 ± 2.70	24.15 ± 9.48	**35.68** ± 2.53	19.18 ± 6.00	10.42 ± 6.94
BM	11.67 ± 1.58	15.11 ± 4.79	12.67 ± 5.78	9.39 ± 3.00	3.05 ± 1.16	27.64 ± 12.83	15.64 ± 1.64	33.68 ± 6.74	34.16 ± 10.68
BMTP	11.23 ± 1.56	11.90 ± 4.27	4.77 ± 2.21	8.56 ± 2.74	1.35 ± 0.55	6.73 ± 5.86	24.21 ± 1.86	35.60 ± 7.38	**46.53** ± 14.00
KIM	11.67 ± 1.58	15.14 ± 4.79	12.68 ± 5.78	9.31 ± 3.00	3.07 ± 1.16	27.67 ± 12.83	15.64 ± 1.64	33.68 ± 6.74	34.16 ± 10.68
Mean P@1									
TP	4.26 ± 1.52	9.76 ± 6.42	0.00 ± 0.00	4.90 ± 2.96	0.00 ± 0.00	0.00 ± 0.00	14.74 ± 2.93	**34.38** ± 11.64	23.91 ± 12.33
CS	**11.18** ± 2.37	**37.80** ± 10.50	**24.39** ± 13.15	**9.31** ± 3.99	0.00 ± 0.00	**32.35** ± 15.73	9.95 ± 2.47	7.81 ± 6.58	3.33 ± 6.42
CSTP	11.03 ± 2.35	31.71 ± 10.07	19.51 ± 12.13	7.84 ± 3.69	0.00 ± 0.00	14.71 ± 11.90	**19.54** ± 3.28	9.38 ± 7.14	3.33 ± 6.42
BM	6.62 ± 1.87	15.85 ± 7.91	7.32 ± 7.97	5.39 ± 3.10	0.00 ± 0.00	20.59 ± 13.59	11.19 ± 2.60	20.31 ± 9.86	16.67 ± 13.34
BMTP	5.59 ± 1.73	12.20 ± 7.08	0.00 ± 0.00	3.92 ± 2.66	0.00 ± 0.00	2.94 ± 5.68	16.70 ± 3.08	29.69 ± 11.19	**36.67** ± 17.24
KIM	6.62 ± 1.87	15.85 ± 7.91	7.32 ± 7.97	5.39 ± 3.10	0.00 ± 0.00	20.59 ± 13.59	11.19 ± 2.60	20.31 ± 9.86	16.67 ± 13.34

Table B3. Precision of website document retrieval from the reduced-size document collection

Relevant documents	**Reduced-size document collection**					
	A (web articles)			**F (fill-in forms)**		
Query	M	A	F	M	A	F
Recall	90.26 ± 1.99	99.09 ± 1.07	100 ± 0.00	84.80 ± 4.49	100 ± 0.00	98.53 ± 2.84
Profile	TC1	TC3	TC5	TC1	TC3	TC5
MAP						
TP	**44.89** ± 2.35	49.03 ± 7.01	**47.25** ± 9.59	21.25 ± 4.06	14.44 ± 4.49	33.09 ± 10.71
CS	30.71 ± 2.44	**51.46** ± 7.26	36.72 ± 7.91	20.38 ± 3.67	19.15 ± 4.73	44.67 ± 11.41
CSTP	32.25 ± 2.35	45.38 ± 6.95	32.99 ± 8.65	19.20 ± 3.59	16.07 ± 5.36	33.42 ± 11.56
BM	33.55 ± 2.17	49.84 ± 6.52	34.60 ± 8.32	24.46 ± 4.04	**23.04** ± 5.77	**55.89** ± 12.94
BMTP	42.88 ± 2.27	49.46 ± 6.87	47.09 ± 9.57	**25.21** ± 4.25	16.31 ± 4.58	40.61 ± 12.13
KIM	33.44 ± 2.16	49.83 ± 6.52	34.66 ± 8.33	24.51 ± 4.04	**23.06** ± 5.77	**55.90** ± 12.94
Mean P@1						
TP	**40.74** ± 3.69	**51.22** ± 10.82	**48.78** ± 15.30	11.27 ± 4.34	1.85 ± 3.60	17.65 ± 12.81
CS	19.71 ± 2.99	47.56 ± 10.81	24.39 ± 13.15	9.80 ± 4.08	3.70 ± 5.04	29.41 ± 15.32
CSTP	20.00 ± 3.01	42.68 ± 10.71	24.39 ± 13.15	8.82 ± 3.89	5.56 ± 6.11	20.59 ± 13.59
BM	22.65 ± 3.15	45.12 ± 10.77	17.07 ± 11.52	12.25 ± 4.50	**11.11** ± 8.38	**47.06** ± 16.78
BMTP	34.56 ± 3.57	**51.22** ± 10.82	43.90 ± 15.19	**12.75** ± 4.58	1.85 ± 3.60	26.47 ± 14.83
KIM	22.50 ± 3.14	45.12 ± 10.77	17.07 ± 11.52	12.25 ± 4.50	**11.11** ± 8.38	**47.06** ± 16.78

Table B4. Precision of website document retrieval at 100% recall, both document collections considered

Relevant documents	**Reduced-size document collection**						**Entire document collection**		
	A (web articles)			**F (fill-in forms)**			**B (brochures)**		
Query	M	A	F	M	A	F	M	A	F
Recall	100	100	100	100	100	100	100	100	100
MAP									
Profile	TC2	TC4	TC6	TC2	TC4	TC6	TC2	TC4	TC6
CS	25.34 ± 2.25	**51.06 ± 7.24**	**28.59 ± 6.28**	16.80 ± 3.27	19.14 ± 4.73	57.29 ± 12.73	19.76 ± 1.35	21.22 ± 5.53	13.53 ± 7.22
CSTP	28.69 ± 2.20	45.07 ± 6.94	20.59 ± 7.00	16.43 ± 3.14	16.02 ± 5.36	44.86 ± 13.58	**33.49 ± 2.50**	19.16 ± 6.01	10.01 ± 4.62
BM	29.76 ± 2.04	49.53 ± 6.53	18.60 ± 6.27	20.94 ± 3.53	**23.04 ± 5.77**	**63.71 ± 12.41**	14.69 ± 1.53	33.68 ± 6.74	29.11 ± 9.22
BMTP	**39.15 ± 2.22**	49.18 ± 6.88	22.55 ± 6.72	**21.97 ± 3.76**	16.31 ± 4.58	53.68 ± 13.09	23.31 ± 1.79	**35.60 ± 7.38**	**34.47 ± 11.19**
Profile	TC8	TC10	TC8	TC7	TC9	TC7	TC8	TC10	TC8
SBERT	24.63 ± 1.87	44.36 ± 6.74	25.09 ± 8.55	5.46 ± 1.17	4.45 ± 1.52	37.56 ± 10.63	3.1 ± 0.52	6.71 ± 1.75	2.83 ± 2.2
Mean P@1									
Profile	TC2	TC4	TC6	TC2	TC4	TC6	TC2	TC4	TC6
CS	16.30 ± 2.67	47.56 ± 10.81	9.76 ± 9.08	8.40 ± 3.52	3.70 ± 5.04	50.00 ± 16.81	6.97 ± 2.08	7.81 ± 6.58	3.33 ± 6.42
CSTP	18.21 ± 2.79	42.68 ± 10.71	12.20 ± 10.02	7.56 ± 3.36	5.56 ± 6.11	41.18 ± 16.54	**17.94 ± 3.14**	9.38 ± 7.14	0.00 ± 0.00
BM	20.11 ± 2.90	45.12 ± 10.77	9.76 ± 9.08	10.50 ± 3.90	**11.11 ± 8.38**	**58.82 ± 16.54**	10.45 ± 2.50	20.31 ± 9.86	10.00 ± 10.74
BMTP	**32.07 ± 3.37**	**51.22 ± 10.82**	17.07 ± 11.52	**11.34 ± 4.03**	1.85 ± 3.60	44.12 ± 16.69	16.03 ± 3.00	**29.69 ± 11.19**	**16.67 ± 13.34**
Profile	TC8	TC10	TC8	TC7	TC9	TC7	TC8	TC10	TC8
SBERT	12.73 ± 2.37	47.19 ± 10.43	**19.57 ± 11.59**	2.36 ± 1.08	0.0 ± 0.0	28.26 ± 13.16	0.78 ± 0.57	0.0 ± 0.0	0.0 ± 0.0