Leveraging Sentence-Transformers to Overcome Query-Document Vocabulary Mismatch in Information Retrieval

Saber Zahhar, Christophe Rodrigues, Nedra Mellouli

Leveraging Sentence-Transformers to Overcome Query-Document Vocabulary Mismatch in Information Retrieval

Saber Zahhar[1](✉)[0009−0007−6562−9060], Nédra Mellouli[2][0000−0001−8858−9902], and Christophe Rodrigues[2][0000−0002−9039−4570]

[1] Devoteam – Paris, France
saber.zahhar@devoteam.com
[2] De Vinci Higher Education, DVRC Research Center – Paris, France
{firstname.lastname}@devinci.fr

Abstract. Query-document vocabulary mismatch represents the gap between a query's terms and the index terms used for document retrieval. It is a significant challenge that affects severely the performance of search algorithms. Our Ph.D. focuses on building a semantic layer that can be shared by both document index terms as well as query terms in order to overcome this problem. In this paper we focus on expanding queries using aligned keyphrases. We show that state-of-the-art keyphrase generation models do improve retrieval but at the cost of an increased vocabulary mismatch. To reduce this effect, we project, using sentence-transformers, the generated keyphrases to their closest representative term from the indexed vocabulary. However, the original set consists of author-assigned annotations which may suffer from issues such as duplication and misspelling. Through the processing of these annotations, we are able to reduce the search space for query-document alignment. We repeat this experiment on keyphrases extracted by tf-idf and demonstrate significant improvements over the author keyphrases, effectively bridging the vocabulary gap and enhancing search relevance.

Keywords: Abstractive Keyphrase Generation · Information Retrieval (IR) · Query Expansion · Thesaurus Generation · Sentence Transformers

1 Introduction

With the democratization of Internet, digital libraries have become the main conduit for accessing, preserving and organizing knowledge. One of its main challenges relates to user experience, most specifically the retrieval effectiveness of their search engine and to that extent the underlying indexing process. Due to copyright and computation constraints, indexing relies on information compression to index its documents. Broad approaches such as text summarization [11] as well as granular ones like keyphrase assignment [4] have been observed.

Keyphrases, also known as *index terms*, are phrases that encapsulate core concepts found within a document. Together with text summaries, they have proven effective in easing downstream tasks such as information retrieval [3].

However, most keyphrase annotation tools have remained since the 1990s *extractive*, i.e. keyphrases are found in the source text. On the one hand, this fails to expand neither the document nor the query given that the keyphrase only highlights already present terms, which possesses less retrieval power than if the keyphrase were absent from the text [3]. On the other hand, extractive annotation does not reduce query-document vocabulary mismatch, this phenomenon corresponds to the gap between terms used by a query and the index terms of relevant documents. This discrepancy heavily impacts the retrieval effectiveness of search engines as they struggle to match semantically similar yet lexically different terms.

To facilitate both reader and author experience in query-document alignment, publishing companines have made efforts to constrain the domain of keyphrases to a professionally curated set of keyphrases, usually called *thesaurus*. Still, such thesaurus are time-consuming to produce and require highly specialized knowledge, limiting the pool of qualified users.

To address these challenges and limitations, we investigate how can a semantic layer be used to bridge the gap between documents and queries. By matching complex, domain-specific and diverse terminology from documents with simple, generic and familiar terms from queries, a semantic layer may greatly reduce the vocabulary mismatch in the retrieval process.

In this paper, we provide an overview on both "information retrieval" and "keyphrase generation" (Section 2), we then explain how can the two intersect in a symbiotic way (Section 3) with an experimental setup (Section 4). We conclude by discussing future avenues for this thesis (Section 5)

2 Literature Review

2.1 Controlled Vocabularies

Controlled vocabularies find their roots in the fields of library science and information science, emerging as a response to the need for systematic organization of knowledge to facilitate retrieval and use. Given the sparse availability of keyphrase-annotated documents, early approaches for generating keyphrases absent from text relied on a thesaurus [15].

The primary purpose of controlled vocabularies is to improve the retrieval effectiveness. By standardizing terms used to describe content, controlled vocabularies can reduce ambiguity and variability in language, making it easier to find related information. In practice, most digital libraries do not possess a professionaly curated controlled vocabulary, especially when they do not cover regulated fields such as legislature and biology. This opens the path to investigate what can be used to serve as index terms.

2.2 Keyphrase Assignment

Keyphrase extraction methods used to be the main focus with two separate schools of thought, one based on statistical unsupervised techniques [9,21] and

the other on graph-based properties [23,18], the latter ended up becoming the mainstream approach these last two decades.

Thanks to the recent advancements in computational capabilities, deep learning has found optimal conditions for its development and applications, allowing researchers to take full advantage of the increasing corpora published. Specifically, neural network architectures paved the way for generative models capable of abstractive keyphrase annotation.

Due to its capacity to generate synonyms which may align with controlled index terms, abstractive keyphrase generation may represent the solution to overcome vocabulary mismatch between keyphrases from queries and index terms of documents.

Pioneered by [17], deep learning generative models used to follow the *One2One* architecture; every training example is formed from a document with a singular keyphrase attached to it split from the targeted keyphrases. For prediction purposes, such models employ a technique known as beam search to generate phrases, choosing those with the top rankings as the candidate keyphrases. However, the architecture failed to consider the relationships among keyphrases, potentially limiting the efficiency of keyphrase generation models.

Addressing the previous limitation, *One2Seq* treats the generation of keyphrases as a task of creating a sequence. Here, the target keyphrases are organized in a specific sequence using delimiters, with present keyphrases arranged by their appearance order in the text and absent keyphrases placed randomly afterwards as described by [16]. Nevertheless, its reliance on a predetermined keyphrase sequence can introduce bias during training phase, particularly if the generated keyphrases don't align with the sequence. Furthermore, production of repetitive keyphrases has been observed in [6].

2.3 Text representation

Introduced by [19], Word2Vec is a neural network architecture focused on capturing semantic relationships between words by representing them in a shared vector space. Doc2Vec [12] extends this principle by training a set of index-document pairs where indices can be n-gram words and documents a list of words. Nevertheless, embeddings remain static regardless of context, e.g. "orange" will have the same embedding when used as a color or a fruit. Overcoming the lack of uncontextualized word embeddings as well as using a more granular tokenization than words. Sentence-Transformers [20] are a recent architecture that leverages advancements made by BERT [7], Sentence-Transformers improves upon traditional embeddings by generating contextualized representations of words, capturing nuanced meaning based on the surrounding words in the text. Unlike Word2Vec and Doc2Vec, Sentence-Transformers are trained to understand the relationships between sentences, making them well-suited for tasks like semantic similarity.

Sentence-Transformers may be able to match keyphrases used as index terms for documents as well as keyphrases used for query expansion.

3 Problem Statement

In our context, we use Sentence-Transformers to act as an embedding system to map keyphrases generated for query expansion to the set of index terms used for documents. This approach has, to the best of our knowledge never been applied in the context of query-document keyphrase alignment, and could reduce the reliance on specific ontologies or vast amounts of training data while enhancing the precision of indexing and retrieval, potentially overcoming the vocabulary mismatch through a keyphrase shared semantic layer.

Methodology

In this section, we present our three-step approach: `Thesaurus Construction`, `Query Expansion` and `Query-Document Alignment`.

3.1 Thesaurus Construction

`Thesaurus Construction` focuses on generating a set of index terms that will serve as the semantic layer. Currently, we focus on simplifying keyphrase annotations from documents through a series of natural language processing techniques:

- Two keyphrases are clustered if they are similar when lowercased;
- Further clustering occurs when their respective stopwords are removed;
- Keyphrases are further grouped when their subwords are stemmed;
- Finally, clustering occurs if their stemmed subwords are the same when sorted alphabetically.

Though simple, these steps allow us to reduce redundancy in the search space. Indeed, annotations such as "Recommendation Systems" and "system recommender" cannot be matched directly due to their lexical distinction despite their clear semantic link. Through our processing steps we are able to cluster both keyphrases into a single keyphrase to be used by the semantic layer, in practice we replace the least common keyphrase of the two by the other. Thus, these processing steps are able to automatically reduce the size of our semantic layer whilst not degrading the performance of retrieval algorithms like BM25, given that it is insensitive to lower-casing and token order.

3.2 Query Expansion

`Query Expansion` consists in concatenating queries with generated keyphrases using state-of-the-art models. We use the query as text input for the model and add to the original query the output keyphrases from the model. In order to avoid bias from over-generating and length-penalty from retrieval systems, we reduced the maximum number of generated keyphrases to 5.

3.3 Query-Document Alignment

`Query-Document Alignment` is performed by computing Sentence-Transformers embeddings for both the index terms and the keyphrases generated for the queries. By applying a pairwise similarity metric between keyphrases and index terms, we can project each keyphrase into a corresponding index term, effectively bridging any possible vocabulary mismatch.

4 Results

4.1 Experimental setup

Dataset Our investigations centered around the `ACM-CR` [2] document retrieval dataset. It consists of 102,411 documents in English on topics related to information retrieval from the ACM Digital Library. 169 citation contexts were extracted from scientific papers from the 2020 proceedings of conferences centered on information retrieval which serve as queries, these citation contexts are sentences or whole paragraphs from human authors that contain one or more references to other scientific papers. We view these cited references as relevant documents for the citation context, that is also serve as a query. 481 query-document relevant pairs are made available.

We also compared two annotation methods to expand the documents:

- Using available author-assigned keyphrases from the documents.
- Using `scikit-learn`'s `TfIdfVectorizer` extraction algorithm with parameters `ngram_range = (1, 4)` and `nltk`'s stopword list and word tokenizer.

This two annotation methods will allow us to compare how index terms can differ and what is the impact on retrieval effectiveness.

	Unique Keyphrases	**Keyphrases w/ count $>$ 1**	**Keyphrase count**
	Before `Thesaurus Processing`		
Authors	112134	26.57%	2.87 ± 15.51
TF-IDF	330992	21.54%	1.44 ± 7.94
	After `Thesaurus Processing`		
Authors	89816	30.08%	3.58 ± 21.22
TF-IDF	266592	31.48%	1.79 ± 9.02

Table 1. Annotation statistics for both configurations. The last three columns are expressed under a "Mean ± Standard deviation" format.

Table 1 presents the varying distributions before and after applying `Thesaurus Processing` on both author-assigned and TF-IDF annotations.

Prior to processing, the author-assigned keyphrases showed a lower total number of unique keyphrases (112,134) compared to TF-IDF (330,992), with a higher percentage of keyphrases that appeared more than once (26.57% vs. 21.54%). First of, almost 30% of documents have no author-assigned keyphrases [3]. However, when applying tf-idf, we extracted the top 5 keyphrase candidates based on the 4.5 average keyphrase per document statistic given by the authors [3]. Moreover, tf-idf's purely extractive nature makes each author's writing style add unique keyphrases to the thesaurus. Still, we observe in both configurations a 20% drop of unique keyphrases after `Thesaurus Processing`'s step. Also, the number of keyphrases that appear more than once is greater after processing, revealing that tf-idf tend to paraphrase itself more often.

Overall, the number of appearance of each keyphrase increased after processing, though with greater variance. This skewed distribution is a sign of a poorly designed thesaurus. Indeed, thesauri are expected to be uniformly distributed and each index term should cover a concept as unique as possible from the others, like independent mathematical axes. Though keyphrases are on average more used, the standard deviation enable us to confirm that the appearance count isn't distributed equally. This may be in part due to the varying degree of specifity found in keyphrases, e.g. both configurations have entries very generic like "error" and "human" as well as very specific ones like "computer-assisted language learning".

Also, we found that current simple processing techniques cannot adequately tackle all redundancy in the dataset. For example, keyphrases like "360-video", "360° video", "360-degree-video" and "360-video mental health" were still separated after our processing step.

Query Expansion To inquire about our mapping on different annotations, we assigned keyphrases to the queries using;

- **MultipartiteRank** [1]: A graph-based extraction method that evaluates the relationships between words and their co-occurrences. By constructing a weighted graph representation of the text, it identifies the most relevant terms based on their structural importance and connectivity.
- **GPT-3.5**: Leveraging the capabilities of a large language model, we used OpenAI's prompt template[3] to facilitate keyphrase extraction.
- **KeyBART** [10]: KeyBART is a pre-trained model based on the BART (Bidirectional and Auto-Regressive Transformers) architecture [14], designed specifically for keyphrase generation following the One2Seq format.

Query-Document Alignment We generate embeddings for index terms from one of the two thesauri after processing using Mixedbread's embedder that appear in the top-30 MTEB leaderboard [**?**][13]. We do the same with keyphrases from queries. Then, for each query's keyphrase, we project it to its closest index term using cosine similarity.

[3] https://platform.openai.com/docs/examples/default-keywords

Evaluation Experiments are carried out by indexing documents using annotations from either the author's or tf-idf thesaurus after processing. We perform ranking of queries and their expansions through the standard Lucene's BM25 model implemented in the BM25-Sparse open-source Python module[4]. Stemming is always carried out, since it is a standard procedure in information retrieval, using `nltk`'s implementation of `PorterStemmer`. Stopwords are usually removed to ease BM25's calculation as per the `bm25s` documentation. For all configurations, we use `bm25s`' default parameters `k_1=1.5, b=0.75`. We evaluate retrieval effectiveness in terms of mean Average Precision (mAP) on the top 10 retrieved documents as per [8]. We also quantify query-document vocabulary mismatching as the ratio of a query's tokens not found in the set of index terms.

	Author indexing		**TF-IDF indexing**	
Query expansion	**mAP@10**	**Mismatch**	**mAP@10**	**Mismatch**
	`Queries with no keyphrases`			
`None (default query)`	0.113	12.70%	0.147	10.20%
	`Queries with raw keyphrases`			
`KeyBART`	0.116	13.60%	0.155	11.70%
`MultipartiteRank`	0.121	13.40%	0.148	10.20%
`GPT-3.5`	0.119	12.70%	0.160	10.80%
	`Queries with projected keyphrases`			
`KeyBART`	0.124	10.70%	0.161	8.70%
`MultipartiteRank`	0.121	11.70%	0.166	9.20%
`GPT-3.5`	0.135	11.20%	0.198	8.40%

Table 2. Experimental results of document retrieval using our configurations, with a comparison between indexing using author-assigned and TF-IDF-based keyphrases.

Results Table 2 shows our experimental results. Across all query expansion techniques and regardless of projecting keyphrases to their respective controlled index terms, TF-IDF provide a better indexing basis for retrieval effectiveness as seen with the mAP score, despite indexing three times more keyphrases (Table 1). We may attribute this to the inherent nature of BM25 that is heavily based off TF-IDF's principle. We observe on average an increase of 3–4 percentage points when switching from authors to tf-idf indices.

Across all configurations, using query expansion techniques improves retrieval performance, with GPT-3.5 having the best increase in retrieval power except against MultipartiteRank in the first unprojected author-assigned thesaurus, which again may be due to poor indexing grounds.

Projecting keyphrases into the controlled thesaurus significantly reduces the vocabulary mismatch, showing that this method is effective in improving query-

[4] https://github.com/xhluca/bm25s

document alignment. Each configuration consistently performed better in retrieval than its non-projected counterpart. The largest score difference in mAP is over 8.5 percentage points and just under 4.5 percentage points in mismatch.

5 Future Directions

5.1 Improving Thesaurus Processing

While we unraveled the potential in building a semantic layer based on TF-IDF annotations, we still fail to cluster many keyphrase pairs. For example, the author-assigned processed thesaurus still left out keyphrases "360-degree video", "360 degree video" and "360° video" as separate index terms. This results in extreme redundancy and poor performance of our keyphrase alignemnt approach.

HDBSCAN [5] is density-based hierarchical clustering technique that allow better clustering of keyphrases by progressively merging index terms based on their semantic similarity, e.g. generic terms like "video" can be linked to specific phrases like "360-degree video".

5.2 Addressing Biases in Abstractive Keyphrase Generation

[22] called the separation of extraction and abstraction keyphrase annotation. In this paper, we introduced a two-step approach that leverages the well-established strength of extractive methods and then refines it through an abstraction phase.

Still, we only computed surface-level representations of index terms, it would be interesting to levearage Large Language Models to build actual definitions of these index terms in order to build more robust embeddings.

References

1. Boudin, F.: Unsupervised keyphrase extraction with multipartite graphs. In: Walker, M., Ji, H., Stent, A. (eds.) Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers). pp. 667–672. Association for Computational Linguistics, New Orleans, Louisiana (Jun 2018). https://doi.org/10.18653/v1/N18-2105, https://aclanthology.org/N18-2105
2. Boudin, F.: Acm-cr: A manually annotated test collection for citation recommendation. In: Proceedings of the 2021 ACM/IEEE Joint Conference on Digital Libraries. p. 280–281. JCDL '21, IEEE Press (2024). https://doi.org/10.1109/JCDL52503.2021.00035, https://doi.org/10.1109/JCDL52503.2021.00035
3. Boudin, F., Gallina, Y.: Redefining absent keyphrases and their effect on retrieval effectiveness. In: Toutanova, K., Rumshisky, A., Zettlemoyer, L., Hakkani-Tur, D., Beltagy, I., Bethard, S., Cotterell, R., Chakraborty, T., Zhou, Y. (eds.) Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. pp. 4185–4193. Association for Computational Linguistics, Online (Jun 2021). https://doi.org/10.18653/v1/2021.naacl-main.330, https://aclanthology.org/2021.naacl-main.330

4. Boudin, F., Gallina, Y., Aizawa, A.: Keyphrase generation for scientific document retrieval. In: Jurafsky, D., Chai, J., Schluter, N., Tetreault, J. (eds.) Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. pp. 1118–1126. Association for Computational Linguistics, Online (Jul 2020). https://doi.org/10.18653/v1/2020.acl-main.105, https://aclanthology.org/2020.acl-main.105
5. Campello, R.J.G.B., Moulavi, D., Sander, J.: Density-based clustering based on hierarchical density estimates. In: Pei, J., Tseng, V.S., Cao, L., Motoda, H., Xu, G. (eds.) Advances in Knowledge Discovery and Data Mining. pp. 160–172. Springer Berlin Heidelberg, Berlin, Heidelberg (2013)
6. Chen, W., Chan, H.P., Li, P., King, I.: Exclusive hierarchical decoding for deep keyphrase generation. In: Jurafsky, D., Chai, J., Schluter, N., Tetreault, J. (eds.) Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. pp. 1095–1105. Association for Computational Linguistics, Online (Jul 2020). https://doi.org/10.18653/v1/2020.acl-main.103, https://aclanthology.org/2020.acl-main.103
7. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: BERT: Pre-training of deep bidirectional transformers for language understanding. In: Burstein, J., Doran, C., Solorio, T. (eds.) Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers). pp. 4171–4186. Association for Computational Linguistics, Minneapolis, Minnesota (Jun 2019). https://doi.org/10.18653/v1/N19-1423, https://aclanthology.org/N19-1423
8. Färber, M., Jatowt, A.: Citation recommendation: approaches and datasets. International Journal on Digital Libraries **21**, 375 – 405 (2020), https://api.semanticscholar.org/CorpusID:211132888
9. Hulth, A.: Improved automatic keyword extraction given more linguistic knowledge. In: Proceedings of the 2003 Conference on Empirical Methods in Natural Language Processing. pp. 216–223 (2003), https://aclanthology.org/W03-1028
10. Kulkarni, M., Mahata, D., Arora, R., Bhowmik, R.: Learning rich representation of keyphrases from text. In: Carpuat, M., de Marneffe, M.C., Meza Ruiz, I.V. (eds.) Findings of the Association for Computational Linguistics: NAACL 2022. pp. 891–906. Association for Computational Linguistics, Seattle, United States (Jul 2022). https://doi.org/10.18653/v1/2022.findings-naacl.67, https://aclanthology.org/2022.findings-naacl.67
11. Lam-Adesina, A.M., Jones, G.J.F.: Applying summarization techniques for term selection in relevance feedback. In: Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. p. 1–9. SIGIR '01, Association for Computing Machinery, New York, NY, USA (2001). https://doi.org/10.1145/383952.383953, https://doi.org/10.1145/383952.383953
12. Le, Q., Mikolov, T.: Distributed representations of sentences and documents. In: Proceedings of the 31st International Conference on International Conference on Machine Learning - Volume 32. p. II–1188–II–1196. ICML'14, JMLR.org (2014)
13. Lee, S., Shakir, A., Koenig, D., Lipp, J.: Open source strikes bread - new fluffy embeddings model (2024), https://www.mixedbread.ai/blog/mxbai-embed-large-v1
14. Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., Stoyanov, V., Zettlemoyer, L.: BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In: Jurafsky, D.,

Chai, J., Schluter, N., Tetreault, J. (eds.) Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. pp. 7871–7880. Association for Computational Linguistics, Online (Jul 2020). https://doi.org/10.18653/v1/2020.acl-main.703, https://aclanthology.org/2020.acl-main.703

15. Medelyan, O., Witten, I.H.: Thesaurus based automatic keyphrase indexing. In: Proceedings of the 6th ACM/IEEE-CS Joint Conference on Digital Libraries. p. 296–297. JCDL '06, Association for Computing Machinery, New York, NY, USA (2006). https://doi.org/10.1145/1141753.1141819, https://doi.org/10.1145/1141753.1141819
16. Meng, R., Yuan, X., Wang, T., Brusilovsky, P., Trischler, A., He, D.: Does order matter? an empirical study on generating multiple keyphrases as a sequence. ArXiv **abs/1909.03590** (2019), https://api.semanticscholar.org/CorpusID:202540437
17. Meng, R., Zhao, S., Han, S., He, D., Brusilovsky, P., Chi, Y.: Deep keyphrase generation. In: Barzilay, R., Kan, M.Y. (eds.) Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). pp. 582–592. Association for Computational Linguistics, Vancouver, Canada (Jul 2017). https://doi.org/10.18653/v1/P17-1054, https://aclanthology.org/P17-1054
18. Mihalcea, R., Tarau, P.: TextRank: Bringing order into text. In: Lin, D., Wu, D. (eds.) Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing. pp. 404–411. Association for Computational Linguistics, Barcelona, Spain (Jul 2004), https://aclanthology.org/W04-3252
19. Mikolov, T., Chen, K., Corrado, G.S., Dean, J.: Efficient estimation of word representations in vector space. In: International Conference on Learning Representations (2013), https://api.semanticscholar.org/CorpusID:5959482
20. Reimers, N., Gurevych, I.: Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In: Inui, K., Jiang, J., Ng, V., Wan, X. (eds.) 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. Association for Computational Linguistics, Hong Kong, China (Nov 2019). https://doi.org/10.18653/v1/D19-1410, https://aclanthology.org/D19-1410
21. Tomokiyo, T., Hurst, M.: A language model approach to keyphrase extraction. In: Proceedings of the ACL 2003 Workshop on Multiword Expressions: Analysis, Acquisition and Treatment. pp. 33–40. Association for Computational Linguistics, Sapporo, Japan (Jul 2003). https://doi.org/10.3115/1119282.1119287, https://aclanthology.org/W03-1805
22. Xie, B., Song, J., Shao, L., Wu, S., Wei, X., Yang, B., Lin, H., Xie, J., Su, J.: From statistical methods to deep learning, automatic keyphrase prediction: A survey. Information Processing Management **60**(4), 103382 (2023). https://doi.org/https://doi.org/10.1016/j.ipm.2023.103382, https://www.sciencedirect.com/science/article/pii/S030645732300119X
23. Zha, H.: Generic summarization and keyphrase extraction using mutual reinforcement principle and sentence clustering. In: Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. p. 113–120. SIGIR '02, Association for Computing Machinery, New York, NY, USA (2002). https://doi.org/10.1145/564376.564398, https://doi.org/10.1145/564376.564398