UNIVERSITÄT KOBLENZ · LANDAU
Fachbereich 4: Informatik

Institute for Software Technology
Institut für Softwaretechnik

Sub-topic modeling in document retrieval systems

Proposal for the Master Thesis

submitted by

Gadiyaram, Sri Sai Praveen

December 20, 2022

Supervisor: Prof. Dr. Jan Jürjens
Institut für Softwaretechnik

Contents

1 Abstract 3

2 Introduction and Motivation 3

3 Background 4

4 Challenges 5

5 Idea 7

6 Related work 8

7 Research question 9

8 Proposed methodology 10

9 Dataset 13

10 Evaluation 14

11 Thesis outline 16

12 Schedule with important deadlines 17

References 17

1 Abstract

Retrieving highly relevant documents in the top results for a given user query is one of the challenging tasks in Information Retrieval (IR). This challenge is more amplified when the user has a very specific intention, and the search query lacks the context related to his intention. Current document retrieval systems based on BM-25 and Semantic search cannot fulfill the user intention, as they were designed to retrieve documents solely based on the user query. To fulfill the user intent and missing context, a novel unsupervised approach is proposed in this master thesis to extract highly coherent query-specific contexts (sub-topics) from search results, which helps the user immensely narrow down the search space. Furthermore, a new extrinsic clustering evaluation approach is introduced, and also the experiment will be evaluated using precision and survey analysis.

2 Introduction and Motivation

Most IR systems today focus on retrieving relevant results based on the query. However, some of the search engines shown in Figure 1 have a unique advantage in improving the search results by learning the user's intention or behavior based on the browsing history, location, cookies, etc. Current search engines are well-established with efficient indexing and retrieval algorithms, and also rely on different data sources such as Blogs, Wikipedia, News articles, Advertisements, cookies, etc.

Figure 1: Most popular search engines [1]

A recent analysis to understand the user search queries on 306 million keywords used in Google search, showed that user queries were comprised majorly of a relatively small number of keywords, and also the mean keyword length is 1.9 words and 8.5 characters[2]. Search queries are majorly classified into 3 categories as *Transactional, Navigational, and Informational queries* as shown in Table 1 on page 3. In this experiment, we are dealing specifically with Informational search queries, where the objective of the query is to request very specific information matching the user's intention.

Table 1: User search query types [2]

[1] https://www.webmarketersguide.com/seo/top-search-engines
[2] https://backlinko.com/google-keyword-study

S No.	Query type	Definition
1	Navigational queries	Search queries that aim to reach a particular website or URL e.g., YouTube.com, amazon.de
2	Transactional queries	Search queries that intend to perform certain transactional activity such as making an online-purchase or downloading research papers
3	Informational queries	Search queries that aim to retrieve very specific information to the user and also covers various domains and topics e.g., Robots, AI, How to make Tea?

Information search queries become very challenging when the query does not provide any context or search objective clearly. For example, let us consider two search queries: **"Robotics"** and **"What are the technological advancements in Robotics related to Unmanned Weapon Systems?"**. Obviously the results of these search queries are going to be different, because the first query is generally mapped to multiple domains and can lead to high amount of noise in the search results. On the other hand, the second query would provide high quality results to the user, as the user intention clearly specified.

Let us consider the case, when the user provides only one or two phrase queries and the user intention is defined to certain topics such as *Innovation in Technology and military*. Knowing the user intention partially cannot fulfill the missing context completely. The main challenge here is defining the user intention not only theoretically, but also programming with the current IR systems and this specific challenge is aimed to solve with this experiment. High variety of sources for context-less search queries can also result in high noise or false positives and the user is less likely to find the relevant documents in the top results. For this purpose, only news articles from diverse sources are considered in this experiment and the results can be easily transferred on other data sources as well.

3 Background

This experiment is realized using a retrieval pipeline containing three major components such as **Web scraper, Document filter**, and **Retriever** as shown in Figure 2. The first component *Web scraper* takes a list of URLs from a SQLite database and downloads the HTML page of news articles. Subsequently, the downloaded HTML text articles are cleaned from advertisements and noise. Each cleaned news article is considered as a single entity, namely a *Document*. The majority of downloaded documents are a mixture of topics such as Military, Technology, AI, etc., and also contains a small number of typical news topics, namely Politics, Sports, Advertisements etc. The documents consist of news articles from the languages German and English.

The second component *Document filter* is an SVM-based classifier that filters most of the irrelevant documents related to typical news topics. The documents are classified into three classes namely **Military, Technology**, and **Negative**. This component is carefully designed to have a high Recall for the positive documents related to technology and the military. Achieving a high precision is a huge challenge as the dataset is very small and highly imbalanced. The dataset used for training the classifier is custom-designed and collected through manual labeling. Three different labelers have labeled the documents in two stages. Mean noun-chunk vectors from pre-trained multi-lingual Universal Sentence Encoder(USE) [25] model

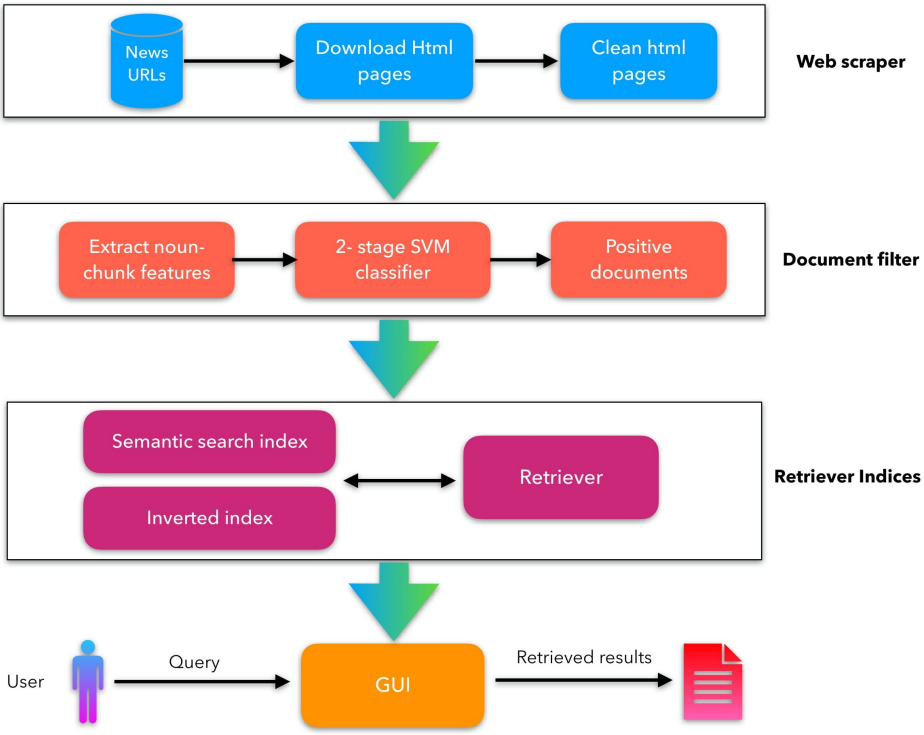

Figure 2: High-level architecture of IR system designed in this experiment

from Tensorflow Hub[3] namely have shown best than the other features such as keyword vectors, verbs, adjectives, etc. Fine-tuning pre-trained transformer models such BERT is also tested and showed bad performance results. In this experiment, Bag of words approaches and word vector features such as word2vec [18], GloVe [22] are completely out of question as they generally pre-trained in a single language and it is very hard to extend the models to vocabulary and stopwords in multiple languages.

Positive documents after the *Document filter* are stored in two different indices namely Inverted index and Semantic search index supported by Elastic search[4] and FAISS[5] respectively. Finally, the component *Retriever* uses both the indices and retrieves documents according to the user request through the web-based GUI as shown in Figure 3. Retriever supports two typical retrieval algorithms such as *BM-25 and Semantic search*, and also retrieve a *Candidate Pool*, which is a combination of retrieval results from both algorithms. User has a choice to select appropriate language and desired document count of retrieval.

4 Challenges

A major challenge in both BM-25 and Semantic search results is high false positives. The algorithms are complementary and no system performs better than the other in all possible user queries, as shown in Table 2 on page 6. Furthermore, below are a few critical problems related to the thesis experiment.

[3] https://tfhub.dev/google/universal-sentence-encoder-multilingual-large/3
[4] https://www.elastic.co/
[5] https://github.com/facebookresearch/faiss

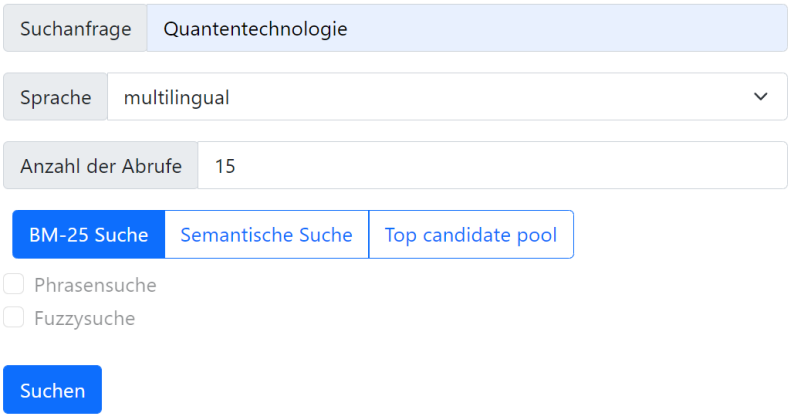

Figure 3: Search UI interface developed for this experiment

1. **Low positive document distribution:** Information related to Innovation and technological breakthroughs is rarely published in news articles. However, the probability of finding such information is not zero, as there are some positive news articles, which are gathered during labeling data for classification and retrieval datasets. But their low distribution makes the retrieval very challenging. Moreover, it is very difficult to rely on a fixed retrieval document count or it can be frustrating for the user to navigate through the long results until he reaches some positive documents according to his intent.

2. **Lack of context in the Query:** In most cases, a well-formed sentence query easily improves the retrieval performance (at least in the case of semantic search) and eases the user to access the positive documents. The query patterns which are focused on in this thesis experiment are majorly keywords, which are phrases containing one to three words. Initial results from the BM-25 and Semantic search look relevant, as they contain query keywords, but not all the results match the user intention. The query *Robotics* is discussed in many fields such as Agriculture, Medicine, Automobile, Military, etc. Without any extra information from the user, it is very challenging to integrate the user's intention to re-rank the retrieval results.

3. **Lack of labeled data:** As mentioned above, low distribution of positive documents results only in a highly imbalanced dataset, which hinders the experimentation of supervised re-ranking approaches. There is also a lack of keyword-based retrieval dataset in data repositories, which is labeled with the user intention of *Innovation in Technology and Military*. This is one of the main reasons to explore unsupervised approaches in this experiment.

4. **Lack of User-feedback:** Current retrieval systems specified above retrieve documents only based on the user query. A keyword-based query without any extra information from the user is insufficient to fulfill the user's intention.

Table 2: Current retrieval algorithm comparison on different query types

S No.	Query type	IR system
1	Simple single word	BM-25
2	Phrase query	BM-25
3	Abbreviations	BM-25
4	Prefix/Suffix queries	BM-25
5	No meaning	BM-25
6	Multi-lingual queries	Semantic search
7	German composite words	Semantic search
8	Spelling mistakes	Semantic search
9	Polysemy	Semantic search
10	Sentence/long phrase queries	Semantic search

5 Idea

After considering the challenges with positive documents for the user intention *Innovation and Technology*, a supervised solution is hard to achieve, in order to match the performance of a proper sentence query. And not only it is difficult to acquire a large dataset, but also the solution is very generic to only one user's intention. Any new changes or additions to presumed user intention will require again a huge effort. Real-time user feedback and continuous reinforcing algorithms can greatly fulfill the lack of labeled datasets, but they need feedback from diverse users regularly. Otherwise, the search results can be highly inclined to a particular user and can lead to biased results.

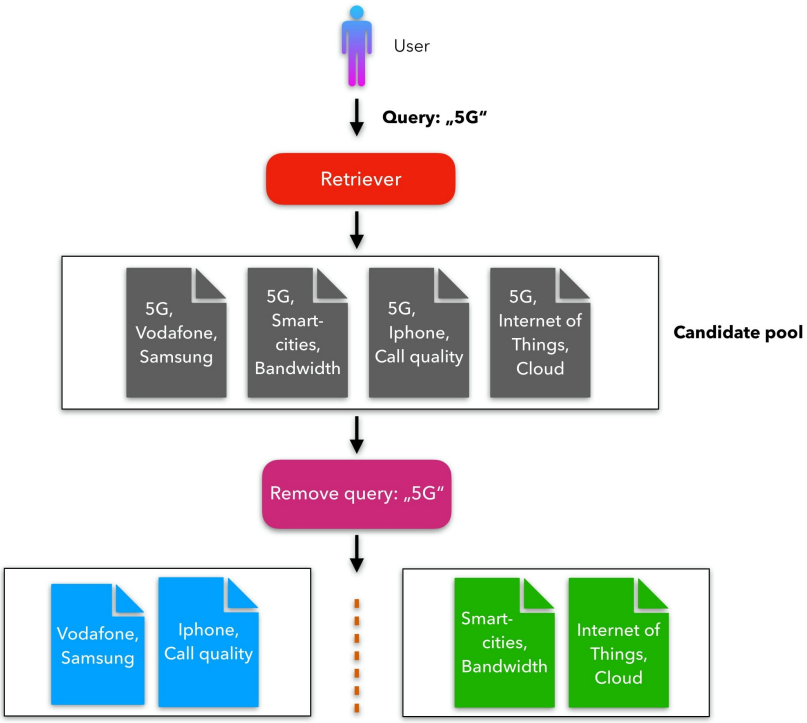

Figure 4: Idea to address the problem statement

A template-based search query is a great option to improve the context of a search query. For example, we have a pre-defined template such as *Innovations in XXX related to the Military*. Later when the user provides a query: *Robotics*, we replace the *XXX* with the user query. This

results in the final query *Innovations in Robotics related to Military*. This approach restricts the user to have only a few sets of templates and is also not efficient when a new template needs to be added or an existing template needs to be updated.

After considering various approaches to fulfill the missing context, extracting the contexts from top results to the user query in an unsupervised way and presenting it to the user seems to be more effective. This would not only help the user to have insights in the results pool, but also reduce the efforts to reach the highly relevant documents. These contexts are described in this experiment as *sub-topics*.

6 Related work

As information is growing at an exponential pace in today's world, the current retrieval systems are challenged with not only the performance and efficiency of the retrieved results but also with the variety of search queries such as text, image, voice, etc. Document retrieval is one of the oldest information retrieval techniques and is still actively researched today. Lexical matching based on *Tf-IDf* or *BM-25* is widely used in many applications and recently semantic search is gaining popularity with promising results. In a sample experiment comparing BM-25 and semantic search, both these popular retrieval systems lack in filtering false positives and has unique advantages and disadvantages over each other, as shown in Table 2 on page 6.

Based on the above retrieval methods, new approaches are tested to improve the quality of retrieved results, and minimize the false positives and researchers have also considered different techniques from Machine Learning(ML) to improve the re-ranking based on the availability of labeled data. In the early 2000's "learning-to-rank" approaches were very popular and are based on pairwise techniques. A set of queries and documents are collected and labeled according to the relevance between the query and the document. ML algorithms are further tuned with special loss functions based on relevance. Some of the popular pairwise ranking methods are RankBoost [7], RankNet [5], Rank-SVM [9] (using click-through data).

Recent state-of-the-art supervised approaches are Neural re-ranking methods and are based on complex Deep Learning(DL) architectures. Distributed word embeddings combined with the performance of non-linear neural networks have shown great results in improving the performance of retrieval systems and also considering semantics [19, 8]. Later the performance is drastically improved by the Transformer architectures, which are distributed and uses special attention mechanisms. One example of such a transformer-based contextualized re-ranking approach is "Passage re-ranking with BERT" presented in [20] has improved the state of the art by almost 27%.

On the other hand, unsupervised approaches are also showing impressive re-ranking of retrieval results and mainly focused on the user query and top retrieved documents [23]. One common challenge in these approaches is the user query, which is mostly comprised of only a few keywords [2, 10]. To tackle this problem, many researchers have tested Query Expansion(QE) approaches that fill the missing meaning and context in the query partially. QE methods include not only fixing spelling mistakes and finding synonyms but also weighting terms in the query and considering vocabulary from external sources [10]. QE based on a co-occurrence matrix is often not sufficient to retrieve relevant results and in this case, external resources like WordNet, Wikipedia, and custom dictionaries designed for particular domains are very practical to use.

QE techniques also include search results clustering, query filtering, word sense disambiguation, relevance feedback, etc [2]. Relevance Feedback is a method of retrieving search

results using the original query given by the user and then using the top-k documents for query expansion. The top results are generally highly relevant and sometimes can also give bad results, especially in the case of semantic search. Researchers have clustered search results in many different ways such as at the document level, keyphrases, query-specific clustering, etc [4, 11, 27, 21, 12, 13]. Typical distance-based clustering algorithms such as k-means are used in some research and also Hierarchical clustering is also tested [4, 17, 26], as it is flexible to change the threshold level for cutting the clustering dendrogram in a bottom-up approach.

Clustering top search results to similar groups segment the results into meaningful search spaces and help the user navigate to an appropriate sub-cluster depending on his search intention. However, the user intention can be inclined to a particular domain or theme, which is not generally extracted through simple clustering at the document or word level. In [14], the authors have used a special candidate selection stage to filter some phrases from the keyword extraction and a specific noun-chunks selection. This pipeline is used specifically for the extraction of innovation insights from research projects. As the user intention is related to *Innovation* in this master thesis experiment, a special query-specific candidate keyword selection is designed and tested.

A common drawback in most clustering approaches is mapping a document to a single cluster, which is not logically true, as a document can contain keywords from different domains. The approaches based on clustering at the word level [4, 17] consider only a single language of retrieval results or corpus and hence cannot be directly implemented on multilingual corpus.

With the advantage of contextual embeddings from sentence encoders, the authors in [1] made a break-through in document clustering with an efficient and explainable topic-modeling approach. Moreover, the documents can be contextually mapped to a certain topic and multiple languages can be easily modeled using a multi-lingual pre-trained encoder. In this experiment, keywords are used as an entity and clustered into semantically similar groups, rather than documents. This gives us the unique advantage of expressing each document as a combination of clusters(*sub-topics*).

7 Research question

RQ: Do sub-topics and it's ranking from the retrieved results of a given query help the user to reach the relevant documents in fewer steps and overcome the lack of context ?

This master thesis experiment aims to provide the user with high-quality query-specific clusters from a large retrieval pool of the original query. An unsupervised soft clustering approach is proposed to model documents (from multiple languages) as a mixture of sub-topics, which are extracted using the deep inherent information from keywords. These highly distinctive and informative sub-topics further help the user to limit the search space and reduce efforts to find the positive documents.

Any external input from the user helps to fulfill the context partially. However, testing whether sub-topics helped the user is quite complex and can be user-dependent. For this reason, the quality of clustering and their ranking is chosen as the absolute way to evaluate the experiment.

8 Proposed methodology

One way to extract different contexts from the candidate pool is to simply perform any clustering algorithm on the retrieved documents. This results in very generic clusters which are closely related to a given query and are not very useful to the user. To generate diverse and distinctive clusters, we need to use the latent information at the word or phrase level rather than at the document level. As the documents contain multiple occurrences of the query and are also highly similar in semantic space, we need to reduce the impact of the given user query to generate a clear distinction between the documents. Figure 5 illustrates the proposed pipeline on an abstract level.

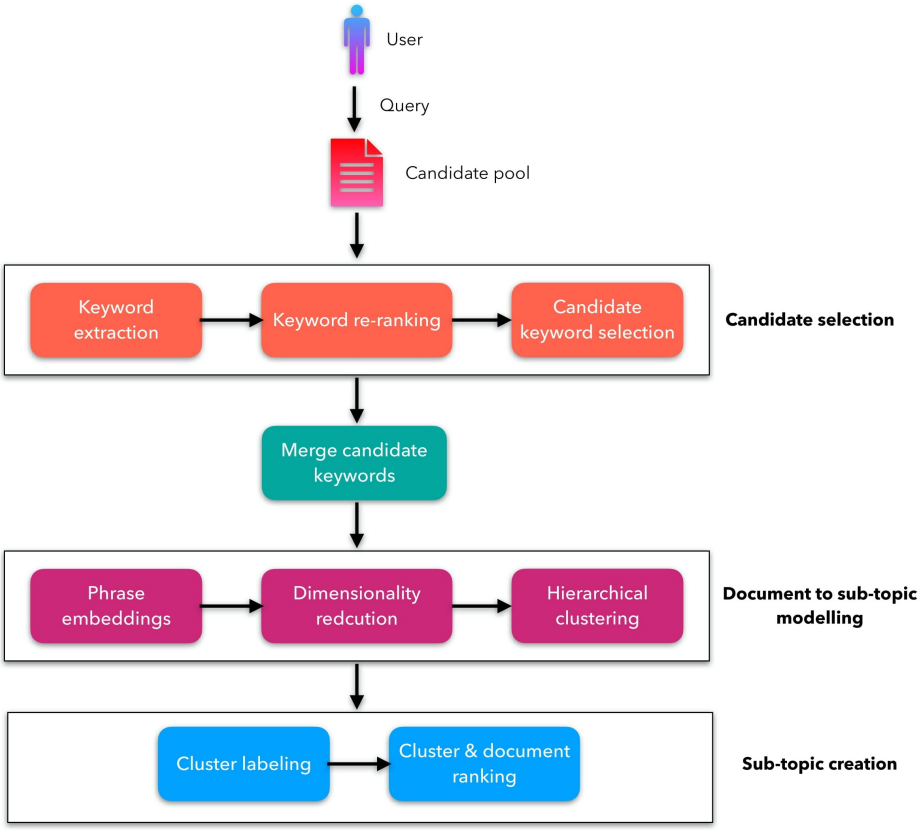

Figure 5: Proposed thesis methodology

The main goal of this experiment is the quality of clusters generated and this approach does not assume any fixed templates or specific user intentions. Major components in the pipeline are: *Candidate selection, Merging candidate keywords, Document to sub-topic modeling, and Sub-topic creation*.

The first step of this pipeline is to retrieve results from both BM-25 and semantic search for the original user query. The results are combined and duplicates are removed to create a candidate pool for the given query. To extract keywords with high diversity and low noise(unwanted or generic words), a Candidate selection module is proposed. This component consists of three major steps namely Keyword extraction, Keyword re-ranking, and Candidate keyword selection.

Keyword extraction is a process of extracting the most influential keywords in a text document and consists of three main stages as Noun-chunk extraction, Noun-chunk cleaning, Noun-chunk re-ranking and Candidate noun-chunk selection. Noun chunks are extracted from

each document with the help of the Spacy[6] library and the output contains a lot of wanted noise. To clean these noun chunks, a pipeline is proposed with the following tasks: *Stopword removal, N-gram range(1, 3), Remove punctuation, Remove determiners, Lemmatization, and Remove noun chunks containing numeric, Remove close duplicates using fuzzy matching*. Cleaned noun chunks are then re-ranked according to the cosine similarity to the original document. Top-k noun-chunks with highest cosine similarity is selected and further used in clustering. This process of selecting top noun chunks in a document is named Candidate noun-chunk selection and resulting phrases are referred to as keywords. This approach is inspired from the recent research related to using contextualized sentence embeddings for keyphrase extraction [3].

In the second stage Keyword re-ranking, cosine similarity is calculated using phrase embeddings between the keywords and the query. Using these similarity scores, keywords are then re-ordered in descending order. After this stage, it is observed that the keywords similar to the query have a high cosine similarity score compared to others and this information can be used to precisely extract some keywords, that are not similar to the query and have a high potential for sub-topics. Using a threshold cut-off, certain keywords are selected and further used for clustering. This process is referred to as Candidate keyword extraction and the resulting phrases after this stage are called Candidate keywords.

The second component in the pipeline *Merge candidate keywords* merges candidate keywords from each document in the candidate pool are merged and duplicates are carefully removed. These keywords are then clustered semantically and modeled with the documents again. This process has been named as Document to sub-topic modeling and is designed independently to the query results and to handle multiple languages. This component has three main steps namely Phrase embeddings, Dimensionality reduction, and Hierarchical clustering. To achieve semantic clustering, multi-lingual contextualized sentence encoders are considered to generate phrase embeddings for each candidate keyword. These densely distributed embeddings are usually highly dimensional(512 or 768) and the clustering in such high dimension is complex to capture patterns and can be resource intensive.

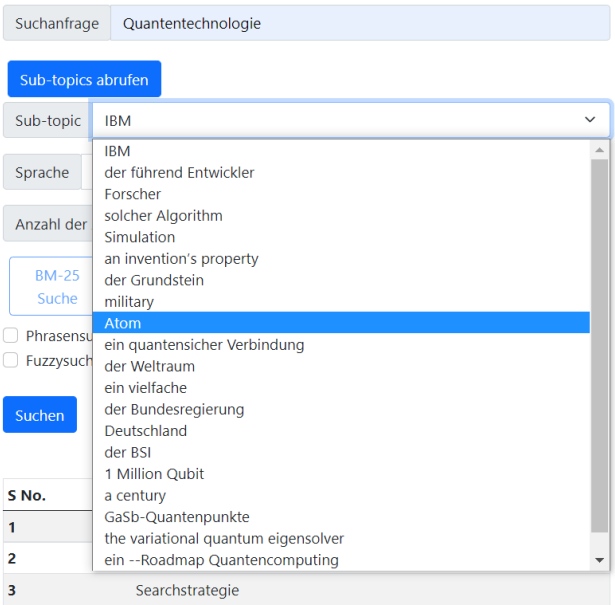

Figure 6: Sub-topic modeling output from initial implementation

Therefore, the dimensionality of the embeddings is reduced without losing underlying

[6] https://spacy.io/

information in the data using the UMAP algorithm [16] from the umap-learn library. These embeddings are further clustered in lower dimensions using a hierarchical clustering algorithm. Noise is expected in the candidate keyword extraction phrase and all the keywords are not important for modeling. The clustering algorithms such as k-means, Gaussian Mixture Models, etc., are not suggested in this case, as they consider all data-points while clustering. Consequently, HDBSCAN and DBSCAN clustering algorithms are preferred as they innately consider the noise in the data and avoid assigning a cluster for every data point. One major advantage of these algorithms is that the cluster count is not a parameter and the algorithm creates clusters effectively based on the data. HDBSCAN algorithm [15] with its varying epsilon and merging clusters has shown robust clustering results by finding varying density clusters and the same algorithm is considered in this experiment. This clustering pipeline is already tested and shown great results with documents in recent research [1].

After clustering, sub-topics are extracted using a centroid approach. A mean phrase vector(centroid vector) is calculated from all the keywords inside a cluster and the closest keyword vector to the centroid vector is considered a cluster label. This process is named cluster labeling and the cluster labels are considered sub-topics. Sub-topics and documents mapped to a sub-topic can be further ranked before showing to the user. The pipeline ends with this last component *Sub-topic creation*. Initial pipeline results are shown in Figure 6. Table 3 on page 12 shows the parameters which will be tuned during clustering evaluation.

Table 3: Parameters in the model for tuning

S No.	Hyperparameters	Range	Possibilities
1	Candidate keyword selection	[0.3, 0.4, 0.5]	3
2	Reduced dimensions (Umap)	[3, 5, 10]	3
3	MIn cluster size (Hdbscan)	[20, 30, 40, 50]	4
4	Min samples(Hdbscan)	[1, 3, 5, 7]	4

Modeling formalization: Each document can contain multiple sub-topics and each sub-topic is mapped to multiple documents. Let us consider a Corpus C that contains n documents and each document D is expressed as a set of candidate keywords (k). The number of keywords selected from each document depends on each document.

$$D = \{k_1, k_2, k_3, \dots\}$$

Once the user provides the retriever a search query q, a candidate pool CP of size m is generated, which is a document set.

$$CP_q = \{D_i, D_j, D_k, \dots, D_m\}$$

Subsequently, a merged candidate keyword set M of size l is extracted, which is a combination of keywords from documents in candidate pool CP.

$$M_q = \{k_i, k_j, k_k, \dots, k_l\}$$

Thereafter the keyword set M_q is clustered into r groups(s) and is defined as a sub-topic set S_q.

$$S_q = \{s_i, s_j, s_k, \dots, s_r\}$$

And each sub-topic (s) is again expressed as a set of keywords (k). The number of keywords in a sub-topic cluster can vary from cluster to cluster.

$$s_i = \{k_x, k_y, k_z, \dots\}$$

The mapping between the document and keywords is already known from above, now we can express each document as a set of sub-topics, and also each sub-topic is expressed as a set of documents.

$$D = \{s_i, s_j, s_k, \dots\}$$
$$s_i = \{D_x, D_y, D_z, \dots\}$$

9 Dataset

A dataset specific to this research problem is hard to find in the current IR data repositories due to two main reasons. Firstly, the search query needs to be a phrase rather than a sentence. Furthermore, the documents need to be labeled with a specific intention, rather than just coherence with the query. In this experiment, the intention is to retrieve the documents related to "Innovation and Technology", and a new dataset is designed for this purpose. Below are a few topics, that describe the user intention for this experiment: **Innovation, Technology Breakthroughs, Future products, Applied research, New procurement strategies, Artificial Intelligence patterns,** and so on. These topics are also described as positive document characteristics because a document is considered positive when it is related to any one of the above-mentioned topics.

Figure 7: Data collection methodology

The main strategy for the data collection is to consider documents from both lexical and semantic matching. Therefore, a candidate pool from search results of both elastic search and semantic search is taken into consideration. The below image shows the methodology followed for the data collection.

Table 4: Relevance label definitions

Label id	Label name	Label definition
1	Perfect	Document that strongly matches with one of the positive document characteristics and also contains a good coherent discussion about the user-given keyword throughout the document.
2	Partially relevant	Document that contains keywords and seems to be relevant, but still lacks innovation or novelty. However, the document shares information about some efficient or optimal way of doing things. Not be a clickbait!!
3	Irrelevant	Document that contains the given user keyword, but still lacks innovation and coherent discussion about the query. Some examples of these documents are click baits, advertisements, marketing blogs etc.. which contains a lot of relevant keywords at the beginning of the document, but yet not useful for this experiment.
4	Wrong/False	These are completely false documents and has nothing to do with the given user query. Eg: for the query "Combat cloud", the documents related to cloud computing are wrong documents.

Relevance labeling shown in Figure 8, is a process of assigning an appropriate label to the retrieval results inside the candidate pool. Every labeler has to assign a label not only concerning the query, but also consider the experiment intention i.e, coherence with positive document characteristics mentioned above. Table 4 on page 13 describes the label descriptions used in the data collection.

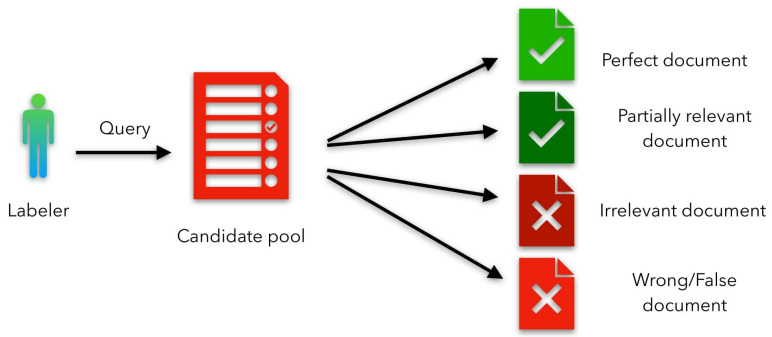

Figure 8: Relevance labeling of candidate pool

10 Evaluation

In most clustering evaluation techniques, an intrinsic evaluation approach without any label information is performed. As described above, a small dataset is collected and the same will

be used for the extrinsic evaluation. To evaluate the quality of clustering for this experiment, *Silhouette index* is considered for the intrinsic evaluation and a custom target function F is designed specifically to this experiment. All clustering algorithms use an objective function based on distance, density, etc., to create distinctive clusters and also often to estimate the cluster count [24]. Irrespective of the clustering algorithm, the output is more distinctive when the distance between the data points within the cluster is minimum and the distance between the clusters is maximum. Silhouette index tests the same criteria and is calculated by using intra-cluster and inter-cluster distances for each sample.

Target function on the other hand tests the quality of clusters against the relevance labels from the dataset, as shown in Table 5 on page 15. The objective is to test whether the relevance documents are clustered into a similar cluster and also the same case with irrelevant documents. Without any relation between clustering and relevance labeling, it can not be assumed that the positive documents and negative documents are clustered automatically because the documents cover a wide range of contexts in different domains and exist in multiple sub-topics. Therefore, it is more meaningful to evaluate the clustering for negative documents i.e., Irrelevant and Wrong labeled documents. A target function is designed to clearly address the number of negative documents isolated through sub-topic modeling.

Table 5: Relevance label distribution

Label id	Label name	Document count
1	Perfect	78
2	Partially relevant	147
3	Irrelevant	306
4	Wrong	98

The output of the sub-topic modeling pipeline is distinctive clusters having a unique context, which are independent of relevance to user intention. However, the clusters can be divided into relevant and irrelevant clusters according to the relevance labels in the dataset. Let us consider that N_1, N_2, N_3, N_4 represent functions to get the number of documents in a single cluster with label ids $1, 2, 3, 4$ respectively, as shown in Table 4 on page 13 and C represents the cluster set.

$$C = \{c_1, c_2, c_2, \dots\}$$

Relevant clusters C_r are clusters, that contain at least one label id 1 or a majority of label id 2. This can be determined using the below expression.

$$C_r = \{c_i \in C | (N_1(c_i) > 0) \vee (2 * N_2(c_i) >= (N_3(c_i) + N_4(c_i)))$$

With this expression, relevant clusters are differentiated from others and the focus is only on labels 1 and 2. The clusters that do not satisfy the above condition are logically considered irrelevant clusters.

$$C_i = \{c_j \in C \setminus C_r\}$$

The target function assesses the clustering with a ratio of documents in irrelevant documents to the documents in the candidate pool CP_q. Given N queries, the target function maps the score using the below equation. Clustering and pipeline parameters are optimized using this function.

$$F = \sum_{i=1}^{N} (|C_i|/|CP_i|) * 100$$

The output of clustering is hard to examine with the baseline IR systems because the order of documents is missing and the actual performance metrics related to false positives are not

addressed. For this purpose, we are extending the sub-topic creation with sub-topic ranking and document ranking. These two rankings help the existing pipeline to create a sequential order of documents and facilitate the evaluation of precision against the baselines. Therefore, this experiment proposes six different retrieval systems and evaluates the ranked results.

Table 6: Proposed IR systems for evaluation

S No.	System type	Sub-topic ranking	Document ranking
1	IR0	NA	Uniform distribution
2	IR1	NA	Query similarity
3	IR2	Query similarity	Query similarity
4	IR3	Template similarity	Template similarity
5	IR4	Document cardinality	Query similarity
6	IR5	Random combinations	Query similarity

The first system *IR0* is an arbitrary system, where the positive documents are distributed uniformly on the ranking order. *IR1* system is simple query re-ranked results based on cosine similarity between the query and documents. The systems *IR2, IR3, IR4* are results of sub-topic pipeline clustering and the clusters are first ranked and later the documents are re-ranked with certain criteria. These three systems simulate the user reading the results in a linear way or as a sequence. In *IR2*, the sub-topic clusters are ranked by the cosine similarity between the query and centroid vector of the cluster and similarly for document ranking.

The system *IR3* use a template similarity criteria, where the similarity is calculated between a template and centroid vector, rather than the query. For example, the template string can be "Innovation and Technology". In the same way, *IR4* clusters are ranked using the number of documents in the cluster. The last system *IR5* is an unreal system just like the *IR0*, but multiple combinations of random ranking clusters are considered to simulate the random selection of a sub-topic by the user and reading the documents in different sub-topics.

In [17], a new evaluation measure for IR systems named expectation score is introduced. The Expectation score (E) is similar to precision (P) but does not consider false positives. E_k represents the number of positive documents at the index k, whereas P_k represents the ratio of the number of positive documents at the index k to k. Furthermore, Mean Average Precision(MAP) [6] and a Survey are also used to evaluate the ranking performance. Mean average precision is one of the most used evaluation metrics in IR. MAP is calculated through the Average Precision(AP) metric, which is an average of precision scores only at the positive document indices. Let us consider G is a set of all positive document indices with size g.

$$AP = (\sum_{i=1}^{G} P_i)/g$$

$$MAP = (\sum_{i=1}^{N} AP_i)/N$$

The survey designed in this experiment is to evaluate the clustering output and also test the potential of new search query results. A new search query strategy is not yet designed and the number of survey inputs is also not finalized.

11 Thesis outline

Pending

12 Schedule with important deadlines

Pending

References

[1] Dimo Angelov. „Top2Vec: Distributed Representations of Topics“. In: *CoRR* abs/2008.09470 (2020). arXiv: 2008.09470. URL: https://arxiv.org/abs/2008.09470.

[2] Hiteshwar Kumar Azad and Akshay Deepak. „Query expansion techniques for information retrieval: a survey“. In: *Information Processing & Management* 56.5 (2019), pp. 1698–1735.

[3] Kamil Bennani-Smires et al. „Simple unsupervised keyphrase extraction using sentence embeddings“. In: *arXiv preprint arXiv:1801.04470* (2018).

[4] Andrea Bernardini, Claudio Carpineto, and Massimiliano D'Amico. „Full-subtopic retrieval with keyphrase-based search results clustering“. In: *2009 IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology*. Vol. 1. IEEE. 2009, pp. 206–213.

[5] Chris Burges et al. „Learning to rank using gradient descent“. In: *Proceedings of the 22nd international conference on Machine learning*. 2005, pp. 89–96.

[6] Gordon V Cormack and Thomas R Lynam. „Statistical precision of information retrieval evaluation“. In: *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*. 2006, pp. 533–540.

[7] Yoav Freund et al. „An efficient boosting algorithm for combining preferences“. In: *Journal of machine learning research* 4.Nov (2003), pp. 933–969.

[8] Jiafeng Guo et al. „A deep relevance matching model for ad-hoc retrieval“. In: *Proceedings of the 25th ACM international on conference on information and knowledge management*. 2016, pp. 55–64.

[9] Thorsten Joachims. „Optimizing search engines using clickthrough data“. In: *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*. 2002, pp. 133–142.

[10] Ashish Kankaria. „Query Expansion techniques“. In: 2015.

[11] Oren Kurland and Carmel Domshlak. „A rank-aggregation approach to searching for optimal query-specific clusters“. In: *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*. 2008, pp. 547–554.

[12] Xiaoyong Liu and W Bruce Croft. „Cluster-based retrieval using language models“. In: *Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*. 2004, pp. 186–193.

[13] Xiaoyong Liu and W Bruce Croft. „Representing clusters for retrieval“. In: *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*. 2006, pp. 671–672.

[14] Francesca Maridina Malloci et al. „A text mining approach to extract and rank innovation insights from research projects“. In: *International Conference on Web Information Systems Engineering*. Springer. 2020, pp. 143–154.

[15] Leland McInnes, John Healy, and Steve Astels. „hdbscan: Hierarchical density based clustering.“ In: *J. Open Source Softw.* 2.11 (2017), p. 205.

[16] Leland McInnes, John Healy, and James Melville. „Umap: Uniform manifold approximation and projection for dimension reduction“. In: *arXiv preprint arXiv:1802.03426* (2018).

[17] Martin Mehlitz et al. „A new evaluation measure for information retrieval systems“. In: *2007 IEEE International Conference on Systems, Man and Cybernetics*. IEEE. 2007, pp. 1200–1204.

[18] Tomas Mikolov et al. „Distributed representations of words and phrases and their compositionality“. In: *Advances in neural information processing systems* 26 (2013).

[19] Bhaskar Mitra, Fernando Diaz, and Nick Craswell. „Learning to match using local and distributed representations of text for web search“. In: *Proceedings of the 26th international conference on world wide web*. 2017, pp. 1291–1299.

[20] Rodrigo Nogueira and Kyunghyun Cho. „Passage Re-ranking with BERT“. In: *arXiv preprint arXiv:1901.04085* (2019).

[21] Stanislaw Osinski and Dawid Weiss. „A concept-driven algorithm for clustering search results“. In: *IEEE Intelligent Systems* 20.3 (2005), pp. 48–54.

[22] Jeffrey Pennington, Richard Socher, and Christopher D Manning. „Glove: Global vectors for word representation“. In: *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*. 2014, pp. 1532–1543.

[23] Shaurya Rohatgi, Jian Wu, and C Lee Giles. „PSU at CLEF-2020 ARQMath Track: Unsupervised Re-ranking using Pretraining.“ In: *CLEF (Working Notes)*. 2020.

[24] Meshal Shutaywi and Nezamoddin N Kachouie. „Silhouette analysis for performance evaluation in machine learning with applications to clustering“. In: *Entropy* 23.6 (2021), p. 759.

[25] Yinfei Yang et al. „Multilingual universal sentence encoder for semantic retrieval“. In: *arXiv preprint arXiv:1907.04307* (2019).

[26] Meng Yuan, Justin Zobel, and Pauline Lin. „Measurement of clustering effectiveness for document collections“. In: *Information Retrieval Journal* (2022), pp. 1–30.

[27] Oren Zamir and Oren Etzioni. „Web document clustering: A feasibility demonstration“. In: *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*. 1998, pp. 46–54.