On the Graph-Based Extension of the Set Based Model: A New Approach on Graph Representation and Model Ensemble.

Nikitas Rigas Kalogeropoulos[1*], Nikolaos Skamnelos[1†] and Christos Makris[1†]

[1*]Computer Engineering and Informatics Department, Computer Engineering and Informatics Department, University Campus, Rio, Patras, 26504, Achaia, Greece.

*Corresponding author(s). E-mail(s): kalogeropo@ceid.upatras.gr;
Contributing authors: nskamnelos94@yahoo.gr; makri@ceid.upatras.gr;
†These authors contributed equally to this work.

Abstract

The purpose of this paper is to present a set of algorithmic improvements upon an extension of the Set-Based Model (Kalogeropoulos et al (2020)) that focuses on the dependence among document terms employing graph representations. A graph-based approach to document representation can positively affect the information retrieval process due to the ability of graphs to capture the syntactic notion and, in some cases, the semantic relationship among document terms. The aforementioned model generates complete graphs; thus, each document term will be interdependent with the rest. Consequently, an interdependent segment or multiple parts of a document called a window is defined, in which the graph creation algorithms are applied. The proposed methods aim to approximate the window size by exploiting the document length. Moreover, an attempt to create multiple windows is made, considering the relationship between a sentence and a paragraph, which is reflected in the semantic importance of nodes and edges. An attempt to tackle the stop-word detection problem on bridge nodes is made by implementing algorithmic schemes that use core decomposition (Seidman (1983a)) to identify the importance of such nodes in a sample of the corpus collection. Finally, a simple reranking scheme and an ensemble voting technique are implemented to enhance model performance on queries where the proposed approach lacks performance. The experimental analysis made on multiple document collections exhibits performance improvements and in some queries, our

approach outperforms even state-of-the-art models such as BM25 (Robertson et al (2004)) and ColBERT (Khattab and Zaharia (2020)).

Keywords: Information Retrieval Models, Set-Based Model, Graphical Document Representation, Stopword Detection, Model Ensemble, Borda Count

1 Introduction

The main focus of information retrieval is the creation of effective and efficient models that comply with the user's information-seeking needs, which are usually expressed in the form of unstructured queries.

Graph theoretic models on information retrieval emerged around 1957 (Firth (1957)). Since then, several graphical formalisms have been implemented to express textual data as such graphs. According to, Medeiros Soares et al (2005) the data is separated into syllables of words, creating a graph in which each node depicts a syllable, and an edge is created between nodes (syllables) that exist in the document. Bordag et al (2003) noticed the semantic relationship of terms in a well-formed text segment, thus suggesting a similarity measure for collocations inside sentences. The proposed graph is bidirectional. Each term is a node on the graph and edges stand for the collocation probability of those terms, which is expressed by the thickness of the edge. Furthermore, Ferrer i Cancho et al (2007) and Ferrer i Cancho et al (2004) focused on terms of syntactic relationships, when creating the graph. Finally, given a document collection, Blanco and Lioma (2012) created two different aspects of co-occurrence text graphs. The first one is undirected and an edge between two nodes—terms is formed if they exist within a window of terms. The term weights are computed similarly to TextRank (Mihalcea and Tarau (2004)), which is a variation of PageRank (Page et al (1999)) or with a degree-based metric called TextLink. Secondly, they implemented a directed co-occurrence graph with grammatical constraints; these constraints are expressed by employing part-of-speech (PoS) tagging. In such a graphical approach, the direction of every edge defines a grammatical dependency between the respective terms.

Rousseau and Vazirgiannis (2013) proposed Graph of Words, a textual representation and a degree-based weighting scheme, which implements a w-sized sliding window, where w is the number of terms that the window consists of. The graph is undirected, even though, a directed case was implemented and found useful in applications such as document summarization or phrasal indexing. Terms are represented as nodes. An edge is formed between two nodes if they exist in the same window. The edge weight reflects the number of nodes co-existing in a window. Finally, for each term in a document, a weight is calculated, which depends on normalized node weight in the respective graph, as well as, the term's inverse document frequency in the collection, having many similarities with the BM25 model (Robertson and Sparck Jones (1977), Robertson and Walker (1994),Robertson et al (2004),Sparck Jones et al (2000)).

In further research, Rousseau and Vazirgiannis (2015) approached the problem of keyword detection using core decomposition (Seidman (1983a)). By retaining nodes

on the main core of each textual graph, they detected keywords of the respective document. In a later step, they introduced methods of estimating important nodes using subgraphs other than the main core. As a first attempt, they focus on dense cores or trusses (Cohen (2008)), ultimately proposing a node ranking that depends on the sum of core numbers in which each neighbor belongs for a given node (Tixier et al (2016)).

In recent days the work of Mongiovì and Gangemi (2024) employ graph structures to enhance the retrieval of relevant textual evidence from extensive document collections. They construct a comprehensive graph that encapsulates co-occurrences of entity mentions across the corpus, enabling the identification of pertinent information as subgraphs within this larger structure. This graph-based methodology is particularly effective when evidence is dispersed across multiple documents, such as in claim verification and open-domain question answering. By representing the relationships between entities in a graph format, the approach facilitates the discovery of interconnected pieces of evidence that collectively support or refute a given claim.

Moreover, another use case of graphs lies again in capturing structural and semantic information, a limitation of traditional keyword-based search systems, which often struggle to capture complex, domain-specific relationships within data. Recent work on the area implements a semantic graph-based approach that maps entities and their intricate interdependencies, enabling more advanced querying capabilities and contextual data retrieval (De Filippis et al (2024)). It is common in traditional information Retrieval tasks, where graphs are implemented to tackle the matching as a graph matching or as an embedding similarity problem. The work of Wang et al (2024) attempts to combine those tasks leveraging the advantages of both by proposing Graph Retrieval framework via Knowledge Distillation.

Additionally, another extension of the set-based model exists (Kalogeropoulos et al (2025)), where terms are represented as nodes and edges capture their relationships. However, that approach, in order to optimize retrieval, applies spectral clustering to group terms into semantic clusters, allowing for pruning redundant edges, making the graph sparser and more efficient while maintaining retrieval accuracy. By leveraging precomputed embeddings via spectral clustering and a k-Nearest Neighbors (K-NN) approach a query expansion can be employed to dynamically incorporate relevant terms with minimal computational cost.

Finally, in the context of Retrieval-Augmented Generation (RAG) and Large Language Models (LLMs), graphs play a pivotal role in enhancing information retrieval and generation processes. Traditional RAG systems often rely on unstructured text retrieval, which may overlook the intricate relationships between entities. To address this limitation, Graph Retrieval-Augmented Generation (GraphRAG) methodologies have been developed, leveraging the structured nature of graphs to capture complex inter-entity relationships. By incorporating graph-based indexing and retrieval, these systems can access and utilize relational knowledge more effectively, leading to more accurate and contextually relevant responses. For instance, in GraphRAG workflows, graph-based indexing organizes information into nodes and edges representing entities and their relationships. During retrieval, graph-guided methods navigate these structures to extract pertinent subgraphs that inform the generation phase, ensuring that

the output is enriched with relational context. This integration of graph structures into RAG frameworks enhances the system's ability to understand and generate nuanced information, thereby improving performance in tasks that require comprehension of complex relationships (Peng et al (2024)).

One important aspect and method of performance enhancement in not only information retrieval models but also clustering or classification is *the model ensemble*. Such approaches focus on the combination/fusion of multiple base models resulting in an outcome. Some of the oldest, yet not deprecated methods are associated with election processes. Voting methods, such as Borda (de Borda (1781)) or Condorcet (Young (1988), Tyomkin and Kurland (2023)) counts are implemented to merge lists of rankings derived from information retrieval models, considering such models as voters and documents as candidates. Fusion models can capture the diversity of base estimators' characteristics, exhibit robustness on models' failures, and are capable of amplifying retrieval precision without many compromises on performance. In this paper, we will employ the Borda count voting method, although in the bibliography, the Condorcet is implemented as well (Markovits et al (2012), Anava et al (2016), Rabinovich et al (2014)).

Similarly, a performance enhancement can be achieved by reranking, an approach that is mainly used in combination with neural retrieval models (Lin et al (2022)). The main idea behind the reranking process is to use a two-phase ranking scheme that employs two different ranking algorithms in series. The first ranking algorithm produces a ranking list, which will be processed by the second scheme. Note that we can repeat this process by employing more reranking phases.

The approach presented in Kalogeropoulos et al (2020) is founded on the assumption that every term within a document shares an equal, bidirectional relationship with every other term, thereby forming a fully connected graph for each document. In this representation, terms are depicted as nodes and their relationships as edges. However, from a linguistic perspective, this assumption is overly simplistic, necessitating methods to constrain intra-document relationships. Notably, this approach results in complete document graphs, which require additional processing to extract meaningful term associations, making it computationally demanding. Additionally, such graphs consist of n^2 edges, posing significant memory challenges.

The concept of sliding windows, apart from the work of Rousseau and Vazirgiannis (2013) is integral to various fields, including graph analysis, streaming data processing, and information retrieval. In graph analysis, sliding window models are utilized to monitor and analyse the most recent subsets of edges or nodes, effectively capturing the evolving structure of dynamic graphs. This approach is particularly useful for maintaining up-to-date information in rapidly changing networks. For instance, Zhang et al (2024) introduced an index-based processing method for connectivity queries within sliding windows on streaming graphs, enabling efficient real-time graph data processing. In streaming data processing, sliding window techniques are fundamental, allowing systems to handle continuous data flows by dividing them into manageable chunks or "windows." This method facilitates real-time analysis and decision-making by focusing on the most recent data within a specified window. Datar et al (2002) discussed maintaining aggregates and statistics over data streams concerning the last seen data

elements, referring to this model as the sliding window model. Khalid and Verberne (2008) examine document segmentation strategies, including disjoint and sliding passages, to enhance passage retrieval for question-answering tasks. The authors evaluate different retrieval models and segmentation methods to determine their effectiveness in retrieving relevant passages.

Here, we will try to embed the text segmentation notion upon the Graph-Based extension of the Set-Based model. We will try to estimate the effect of window sizes, following the logical linguistic partition of sentence, paragraph, and a combination of the above, implementing multiple windows.

Our Contribution. In this study, an effort is made to alleviate the connection among terms in the textual graphs that arise in their proposed extension of the Set-Based model (Possas et al (2002); Pôssas et al (2005); Kalogeropoulos et al (2020)) by implementing appropriate text partitions termed windows and combining appropriately the resulted scheme with state-of-the-art approaches. We introduce several algorithmic enhancements to the Graph-Based Extension of the Set-Based Model, aiming to optimize the document-term relationships through structured graph representations. First, we propose a window-based graph construction method that segments documents into variable-sized partitions (windows), reducing computational complexity while preserving both syntactic and semantic relationships. This mitigates the issue of fully connected document graphs, which introduce excessive edges and redundant dependencies. Additionally, we address the stopword detection problem in graph structures by extending core decomposition techniques to identify and prune high-degree bridge nodes. By ranking terms based on their structural importance within the union graph, we develop a refined filtering mechanism to improve retrieval accuracy. To further enhance performance, we incorporate ensemble and reranking strategies, leveraging the Borda count voting method to combine multiple ranking models and integrating ColBERT as a reranker to refine document rankings effectively. Our contributions are validated on multiple document collections showcasing improved retrieval performance while maintaining computational efficiency. The experimental results indicate that our approach competes with and, in some cases, outperforms state-of-the-art models such as BM25 and ColBERT, making it a robust and efficient alternative in graph-based information retrieval.

The remainder of this paper is organized as follows. In Section 2, the graph-based extension of the Set-based model is summarized to acknowledge the focal points and novelties, which will be discussed in Section 3. In that part, an alternative graph creation will be presented, as well as multiple Windows implementations. Furthermore, in that section, we will elaborate on key nodes, their importance, the implemented reranking technique and the implemented ensemble method, which will be tested in Section 4, considering multiple collections having a variety of document and query structures. Finally, in the last section, our conclusion will be drawn and we will present future contribution aspects and directions.

2 Preliminaries and methods

As mentioned before, Kalogeropoulos et al (2020) proposed an information retrieval model employing graphs. Each document is mapped to a complete graph, called *Rational Path Graph*. Each term is related to the rest as a consequence of the graph's completeness. Every term of a given document is represented as a node in the graph and a relationship between two nodes as a weighted edge (out-edge) having weight W_{out}. Additionally, each node is characterized by a self-loop weighted edge (in-edge) with weight W_{in}. The weights of each out-edge and in-edge are calculated concerning the term frequency of the respective nodes, taking into consideration the lemmas that follow.

2.1 Rational Path Graph

In their work Kalogeropoulos et al (2020), proposed a graph creation algorithm. As an input, they used an inverted list, which contains terms and their respective term frequencies in the documents thus forming a Rational Path Graph as an output. In our work, we will propose an alternative method of constructing the same graph, with different representations as an input.

Definition 1.

$$W_{in} = \frac{tf_i \cdot (tf_i + 1)}{2} \tag{1}$$

Definition 2.

$$W_{out} = tf_i \cdot tf_j, \text{ WHEN } N_i \neq N_j \tag{2}$$

2.2 Union Graph

After a document is processed and the rational path graph is completed, a collection-sized graph is created as the union of each separate document's graph (Sonawane and Kulkarni (2014)). The graph union is merging two or more graphs without any information loss to the initial corpus. If a document collection D with documents $\{d_1, d_2, \cdots, d_n\}$ is considered and their respective graph representatives are $\{G_1, G_2, \cdots, G_n\}$ then the graph union of the collection D is represented as

$$G_D = \bigcup_{i=1}^{n} G_i \tag{3}$$

In practice, as shown in figure 1 when merging separate graphs into one union graph G_D new nodes or edges are included in the new graph, and on co-existing edges. The weight of an edge on the union graph is calculated as the sum of individual weights. For example, in figure 1 the edge (A, B) exists in both document graphs, thus the weight of that edge is calculated as 6 in their union. Similarly, nodes C and D are included along with their associated edges.

At this point, following the work of Rousseau and Vazirgiannis (2015) and Tixier et al (2016) on keyword estimation, an attempt to augment the weight of such words is made, implementing in a similar manner core decomposition on document graphs.

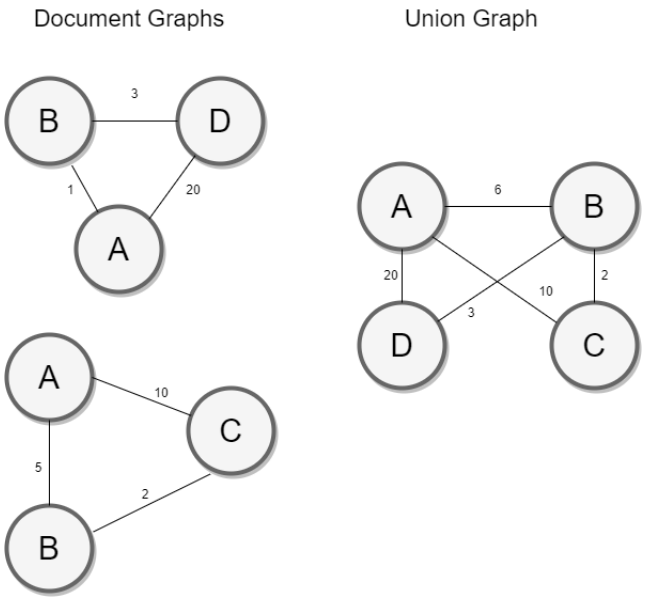


Fig. 1 Union Graph Example

In the context of graph theory, core decomposition of a graph involves partitioning the nodes into layers known as cores, where each layer represents a subset of nodes with a certain level of connectivity gauged by each node degree on that sub-graph (Seidman (1983b)). Due to the graph completeness, though, a pruning process is imperative, because on a complete graph, all nodes belong to the main core, by definition.

At this point, an average edge weight is calculated on the rational path graph, and edges that are less than a threshold are removed. That limit is calculated as the product $p \cdot Average\ edge\ weight$, where p is a heuristic percentage variable. Finally, every edge related to a keyword node is multiplied by an importance variable h.

After completion of processing the document collection, a derived of the union graph weight is calculated for each node k, as shown in the following equation.

$$nw_k = \log\left(1 + a\frac{W_{out_k}}{(W_{in_k}+1)(Cn_k+1)}\right)\log\left(1 + b\frac{1}{Cn_k+1}\right) \tag{4}$$

where, W_{out_k} is the sum of the weights of all out-edges of node k, W_{in_k} is the weight of node's k in-edge, Cn_k stands for the number of its neighbors, and a, b are the variables which affect the importance of graph-based extension against the Set-Based model original weights. Thereafter, a simple inverted index can be created, which contains the calculated weights, as well as, supplementary information used by the Set-Based model's weighting process.

2.3 Graphs and Set-Based model

The retrieval process is handled by the Set-Based model. As mentioned, the notion of termsets, related to query terms, contributes highly to complexity reduction, due to processing significantly lower data volume (Possas et al (2002), Pôssas et al (2005)). It is important to notice that the model implements association rule mining algorithms (Agrawal and Srikant (1994)), to combine two frequent termsets in a different element of each set, thus creating a new one, which if frequent, will be considered a termset of the model. The frequency threshold is user-defined and contributes to decreasing the complexity even more.

The set-based model uses a termset frequency (TS) and inverse document frequency (IDF) based metric for termset weights. The graph-based extension introduces a new measure, (tnw), to combine the graph-derived weights. Therefore, for a term set S_i consisting of k terms the weight is calculated as the product of those term weights as determined on the union graph.

$$tnw_{S_i} = \prod_{k \in S_i} nw_k \tag{5}$$

The new measure acts multiplicatively to the set-based model's weighting process, as shown in the following equation. In this equation, Sf_{ij} represents the term frequency of the set S_i in document j and $\frac{N}{dS_i}$ denotes the inverse document frequency of termset S_i, which will be used as the query scoring function.

$$W_{S_{ij}} = (1 + \log Sf_{ij}) \cdot \log\left(1 + \frac{N}{dS_i}\right) \cdot tnw_{S_i} \tag{6}$$

$$W_{S_{iq}} = \log\left(1 + \frac{N}{dS_i}\right) \tag{7}$$

Finally, the document ($\overrightarrow{d_j}$) and query ($\overrightarrow{Q}$) vectors are formed having as the size of at most 2^n elements, where n is the number of unique terms that the query consists of. It is imperative to notice that a collection-wise termset calculation would be non-realistic since it is computationally expensive because the lexicon size N is vastly larger than the number of query terms ($N \gg n$). As the set-based model dictates, the ranking scheme is expressed as the ordered cosine similarity between the collection documents and the query.

$$sim(Q, d_j) = \frac{\overrightarrow{d_j} \cdot \overrightarrow{Q}}{\|\overrightarrow{d_j}\| \times \|\overrightarrow{Q}\|} \tag{8}$$

2.4 Dense Retrieval

The current flow of ranking research in information retrieval (Lin et al (2022)) is on dense neural models, which use contextualized representations of terms and documents through deep machine learning techniques. The most important one is the BERT model of Devlin et al (2018). It is a large language model (LLM) exploiting the architecture and the attention mechanism proposed by Vaswani et al (2017). The

focal point of BERT is the training on a vast amount of data on two separate problems, the Masked Language and the Next Sentence Prediction. Diving a bit deeper into the model, it takes as input a sequence of document terms and offers as an output a vector of embeddings for each term, as well as a vector for the document, known as classification (CLS) token, since it is mainly used for classification purposes. The attention mechanism highlights the important parts or document terms, thus branching away from the recurrent architecture that was widely implemented in sequence models. Therefore, the model offers a well-formed, dense representation of terms and documents.

Such representations can be utilized alongside classical models on tasks like reranking (Nogueira et al (2019)), document and query expansion (Mallia et al (2021)), or even on creating new dense retrieval models, for instance, ColBERT (Khattab and Zaharia (2020)). The latter model exploits the BERT model and creates a scoring function that achieves improved results on various information retrieval problems. The main idea is to pass each document and query pair through two separate and independent (BERT) transformers, embedding into term vectors information from the model. For each query and document term, a cosine similarity is calculated creating a matrix of similarities for query and document terms. For each query term, the max similarity is calculated. To be more specific, for each row in the similarity matrix, the maximum value is located. Those max similarities are added to create a query-document relevance score. That method and scoring function are applied to each document in the collection and finally, by sorting documents by their score, a ranking emerges.

3 Proposed Methods

The purpose of this section is to examine the rational path graph completeness, in order to refine and model more appropriately the relationship among terms in a document. At first, the edge pruning methods proposed by Zhou et al (2012) were considered, but due to the fear of information loss, a windowed approach seemed linguistically more reasonable.

3.1 Windowed Rational Path Graph

In the initial formulation of the Graph-Based extension of the Set-Based model of Kalogeropoulos et al (2020), a graph construction algorithm was proposed having as input a set of unique terms and the respective document frequencies and calculating the edge weights using the equations in definitions 1 and 2. In our proposal, we follow an alternative approach, where the initial input is defined by the whole unprocessed document. This algorithm is preferable because it improves the time complexity of the document pre-processing compared to Kalogeropoulos et al (2020).

Let a document collection $D = \{d_1, d_2, \cdots d_n\}$, $n \in \mathbb{N}$ and suppose that each d_j consists of terms that exist in a dictionary $T = \{t_1, t_2, \cdots t_n\}$. For each document, the respective graph is constructed as described on algorithm 1. The algorithm is designed to create the graph, nodes, and edges with their weights, assuming that in the document, all terms are related to each other.

Algorithm 1: Rational Path Graph Construction Algorithm

Input: Document D_j
Output: Graph G_j

```
insert each term in a Stack S
while |S| ≠ 0 do
    t = S.pop()
    if t ∉ G_j then
        increase the weight of the edge {t, t} by 1
    else
        initialize an edge {t, t} with weight 1
    foreach x ∈ S do
    if edge {t, x} ∈ G_j then
        increase the weight of the edge {t, x} by 1
    else
        initialize an edge {t, x} with weight 1
    ;
```

More analytically, assuming the following sequence of document terms $t_1, t_2, \cdots t_i, \cdots t_j, \cdots t_n$. We aim to calculate the weight of the $\{t_i, t_j\}$ edge, as shown on the algorithm 1.

If i equals j, then each time the term t_i is taken out from the sequence, the $\{t_i, t_i\}$ edge's weight will be incremented by the number of t_i co-occurrences. Thus, assuming that tf_i is the term frequency of t_i, the first time t_i is popped, the edge will have as weight tf_i and on the last, the already calculated weight plus 1, which concludes to a simple arithmetic sequence. If we formalize:

$$W_{in} = tf_i + (tf_i - 1) + (tf_i - 2) + \cdots + 1 = \frac{tf_i \cdot (tf_i + 1)}{2} \tag{9}$$

If i does not equal j, then each time the terms t_i or t_j are poped the $\{t_i, t_j\}$ will have its weight incremented by 1, for each remaining t_i or t_j depending on the popped term. Due to the commutative law and the undirected edges of the graph, we can rearrange the term sequence to contain all the t_i terms followed by the t_j terms. Finally, if tf_i is the number of occurrences of t_i and tf_j that of t_j, then,

$$w_{out} = tf_j + \cdots + tf_j = tf_i \cdot tf_j \tag{10}$$

As mentioned previously, the graph-based approaches are usually combined with a window approach, where a window is equivalent to the notion of a text segment or a sentence. However, as shown on algorithm 2, the in-document relationships are altered when implementing text windows.

It is of high importance to notice that the proposed algorithms do not implement overlapping windows. Moreover, on most windowed approaches, the size is a given constant W, thus as an alternative option, experiments were made by calculating the

Algorithm 2: Windowed Rational Path Graph Construction Algorithm

Input: Document D_j, window size W
Output: Graph G_j

Split D_j into n parts P, with $n = \frac{\|D_j\|}{W}$
for *i in range* $[0, n-1]$ **do**
    insert each term of P_i in a Stack S
    while $|S| \neq 0$ **do**
        t = S.pop()
        if $t \notin G_j$ **then**
            increase the weight of the edge $\{t, t\}$ by 1
        else
            initialize an edge $\{t, t\}$ with weight 1
        foreach $x \in S$ **do**
        if *edge* $\{t, x\} \in G_j$ **then**
            increase the weight of the edge $\{t, x\}$ by 1
        else
            initialize an edge $\{t, x\}$ with weight 1
        ;

window size w as a percentage of the document size (eq. 11).

$$W = |d_j| \cdot p, \text{ where p is the user defined percentage} \tag{11}$$

The main reason the algorithms 1 and 2 are presented is to better understand that the terms are connected as an undirected graph, and how their connection is formed. The time complexity of the algorithm (alg.1) is $\frac{1}{2} \cdot m(m+1) = O(m^2)$, for every document, where m is the number of document terms. On the windowed approach 2, the complexity will be calculated as, $n \cdot (\frac{1}{2} \cdot W(W+1))$, where n stands for the number of document partitions and W is the window size. The window size is less than the document length, reducing the graph construction time and creating graphs containing cliques. Consequently, the windowed method deviates from completeness, providing graphs that capture the semantic, as well as, the syntactic notion of the document.

As an effort to attain a more accurate sentence perspective, an interchangeable window is proposed by using the dot symbol to locate sentences. That particular method aims to compare, in terms of performance, the perfect sentence window against the other windowed methods, that employ different rules on text segmentation. However, that showed no meaningful result to report.

Until this point, the sentence view is expressed as a single text window. As an effort to include in-paragraph relationships, a second window can be implemented similarly to that of a sentence but different in size. Finally, the importance of sentence-paragraph correlation can be manipulated by two parameters α, β, ultimately altering

the edge weight according to the following equation:

$$W_e = \alpha \cdot W_{e_{sentence}} + \beta \cdot W_{e_{sentence}} \tag{12}$$

3.2 Importance of Bridge Nodes

Following the work of Tixier et al (2016), Rousseau and Vazirgiannis (2015) as well as that of Kalogeropoulos et al (2020) an effort to elaborate on the main core nodes of each graph's core decomposition is made. At first, we try to amplify important nodes as shown in the above-mentioned related work. However, we noticed that in our proposed method, the main core consisted of not only keywords but also stopwords. That acted as a motivation for further research on the stopword detection problem.

For each Rational Path document graph, we implemented the core decomposition scheme and noticed that on average 30% - 40% of nodes that consisted of the main core were stopwords. That of course is a strong indication, that the main core contains mainly stopwords, keywords, and a few words of random unknown importance. Implementing the same method on the union graph, the percentage of such words was 67% on average.

Taking all the above into consideration, we proposed a stop-word detection algorithm as described in algorithm 3, In particular, we employed a sample of the document collection, plus its respective graph representations, and we proposed a scoring function S_i for each graph node i:

$$S_i = \frac{\sum_j^N e_{i,j}}{W_{size}} + \alpha \times W_{p_i} \tag{13}$$

where, $e_{i,j}$ is the edge weight for every edge that is related to node i, w_{size} is the window size that normalises the edge sum and W_{P_i} is a binary sum that expresses the existence of node i in a document graph. By sorting accordingly, a ranking emerges locating the top stop-words. As sampling we implemented in our experiments (see subsection 4.4) a simple hashing of the document IDs function into buckets. Thereafter we will rank each term in a sample - bucket using the equation 13.

Algorithm 3: Stop - Word Detection Algorithm

Input: Collection C
Output: Word Ranking List L

Sampling (i.e hashing by Document ID)
for $d_i \in C$ **do**
    Assign d_j to bucket: $Y = ID_{d_j}$ mod X
for $d_k \in BucketY$ **do**
    for *term* $t_i \in d_K$ **do**
        $S_i = S_i + \frac{\sum_j^N e_{i,j}}{W_{size}} + 1 \times \alpha$

3.3 Ensemble and Reranking Methods

To maximise the effectiveness of our proposed approach and exploit its benefits, we considered its combination with other models, like BM25 (Robertson et al (2004)) and ColBERT (Khattab and Zaharia (2020)). We employ separately the approaches of ensembling (fig. 2) and reranking (fig. 3), as we describe in the text that follows.

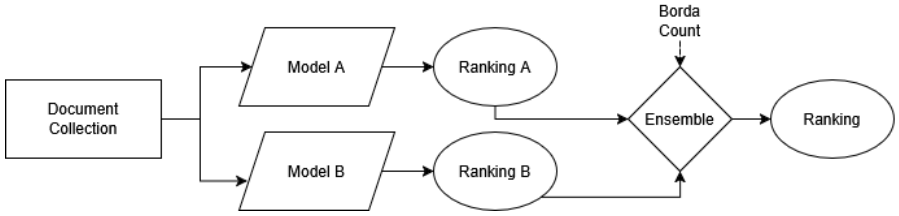


Fig. 2 Borda Count Schema.

To exploit the advantages of some implementations and cover some of their shortcomings, an ensemble technique was used to fuse model results into one final model. In continued-valued cases, the most popular ensemble option is the average or even weighted one. We employ a well-known voting method, termed Borda Count (de Borda (1781)) as described in the following algorithm:

Algorithm 4: Borda Count Algorithm

Input: Ranking A, Ranking B
Output: Final Ranking
candidate set = Ranking A ∪ Ranking B
for *candidate* ∈ *candidate set* **do**
    candidate scores[candidate][score] = 0
Points of each candidate will be determined by the respective index;
for *candidate* ∈ *candidate scores* **do**
    for *can* ∈ *Ranking A* **do**
        candidate scores[can][score] += |*RankingA*| - index of *can* in *Rankin A*
    for *can* ∈ *Ranking B* **do**
        candidate scores[can][score] += |*RankingB*| - index of *can* in *Rankin B*
Sort by candidate score by score on a **Final Ranking**

We consider models as voters and documents as candidates. The algorithm gets as input the base model rankings and outputs the final result list. The algorithm exploits a two-voter case for simplicity but it can be generalized to accommodate more models. The candidate set is created as the union of every ranking, handling any case of different candidates in the ranking list; then the score for each candidate is initialized with a 0 value. Finally, if we consider that the most important candidate will be on

the left side of the list indexed by 0, the score for each document in the ranking will be calculated as the difference between the list's size with the candidate's index. That score will be added to the total candidate score. When all documents and rankings are processed, the final result will be presented, by sorting the candidate list by score.

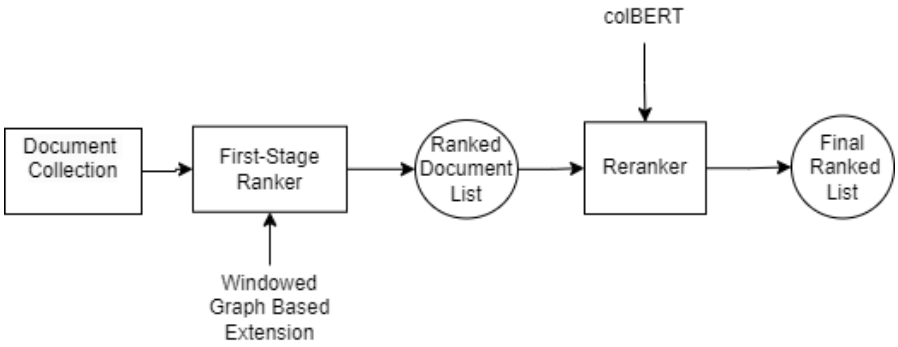


Fig. 3 Reranker Schema.

In the case of neural models, such as ColBERT, another approach is usually used, that of reranking. As the most simple and effective approach, initially, a base model handles the ranking process as a first-stage ranker and thereafter the reranker, which in our case is the ColBERT model, selects and rearranges the upper part of the ranking list accordingly. That process is depicted in figure 3. Hence in our evaluation, we will try to estimate the benefits of our approach, when used as a first-stage ranker.

4 Experiments

To evaluate the effectiveness and efficiency of our proposed methods, we conduct a series of experiments designed as an ablation study to systematically assess the impact of different graph construction techniques, ranking models, stopword detection mechanisms, and ensemble methods. Our experiments aim to quantify the trade-offs between retrieval accuracy and computational efficiency while identifying the contributions of each algorithmic enhancement.

We begin by analysing graph-based retrieval performance, comparing fully connected graphs to windowed approaches with various segment sizes to determine the effect of localized term relationships on retrieval effectiveness. Next, we explore stopword detection and pruning by evaluating how bridge node removal influences ranking and retrieval precision. Additionally, we assess the role of ensemble ranking strategies by implementing Borda count fusion and reranking with ColBERT to examine whether these techniques improve retrieval performance over individual models. Lastly, we conduct experiments on computational trade-offs, measuring the impact of graph sparsification, edge pruning, and model ensembles on retrieval speed and accuracy. The BM25, the set-based model, the simple graph based extension of the set based model (GSB) and the ColBERT will be regarded as our baselines for comparison.

Each experiment is performed across multiple benchmark collections, including Cystic Fibrosis, Cranfield, and Time, to ensure robustness across different query-document structures.

4.1 Collections and Experimental Setup

This section focuses on the experiments[1] on multiple collections (University of Glasgow (2011), Shaw et al (1991)).

Name	Acronym	Docs	Queries	Size (in MB)
Cystic Fibrosis	CFC	1.239	100	1.47
Cranfield	CRAN	1400	225	1.71
Time	TC	423	83	1.5

Table 1 Collections and types of experiments used on each one.

For each query in the collection, the Average Precision (A.P) is calculated. In each experimental case, the mean of average precision (MAP) is depicted, as shown in the following figures. The system that was used consists of a Ryzen 7 4800H 8 Cores (16 Logical) with clock speed at 2.9 GHz, and 16GB of RAM on a Windows operating system.

The Cystic Fibrosis (CFC), Cranfield (CRAN) (Shaw et al (1991)), and Time (TC) University of Glasgow (2011) collections were used for the evaluation of the retrieval ranking problem, while the Time collection alone was utilized for stopword detection. These collections provide queries and expert-judged relevant documents, enabling precise performance evaluation through standard information retrieval metrics. The Time collection also includes an expert-derived stopword list, facilitating validation of our stopword detection method. The Cystic Fibrosis collection consists of 1,239 well-formed abstracts with short to medium-length queries, making it suitable for evaluating retrieval models on concise document structures. The Cranfield collection, containing 1,400 documents and 225 queries, serves as a benchmark for classical information retrieval tasks, particularly in experimental evaluation of ranking methods. Finally, the Time collection, with 423 documents and 83 queries, comprises larger documents and medium-to-large queries, providing a more complex retrieval scenario that challenges ranking models with longer textual contexts.

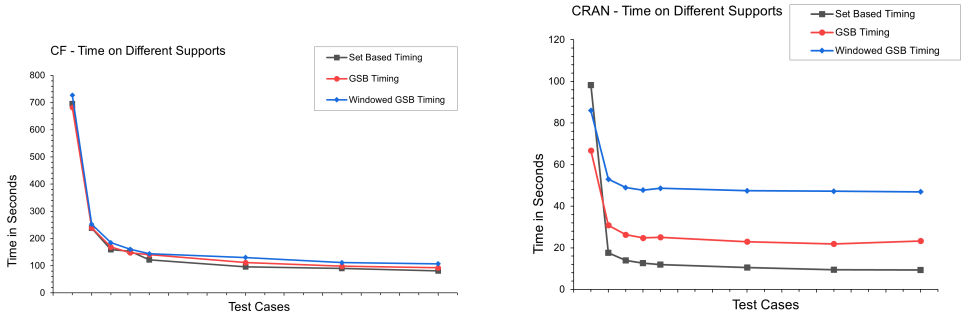
Our choice to focus on this collection choice was made by taking into consideration the variety of offered documents. Therefore, our models are evaluated on collections with different characteristics (see Table 1).

4.2 Trade-off between Processing time and Precision

In this section, we will evaluate the effect and importance of user-specified support on frequent set extraction, a technique that is borrowed from association rules algorithms. Figure 4 depicts the time needed to index the Cystic Fibrosis and Cranfield collection, and to answer the respective queries. It is reasonable to notice that

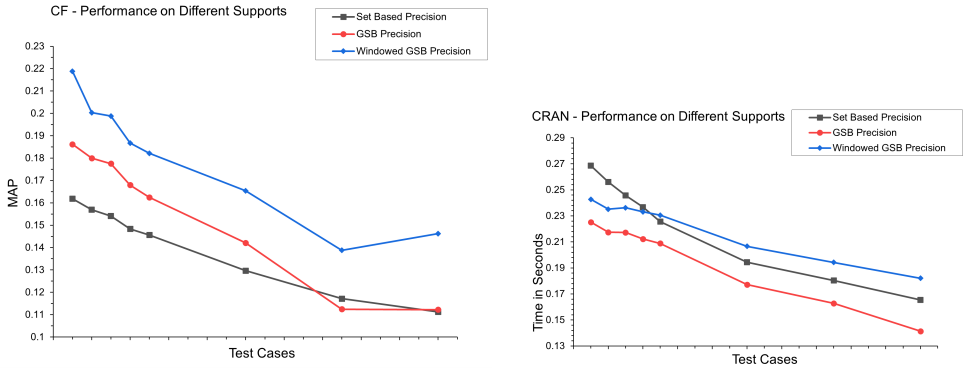
[1] Our source code is freely available at: Github GSB models and Github wGSB

as the support gets lower, the higher becomes the processing time (in seconds); however, more information is engulfed in the document vectors, since they become larger. This implies that all models based on the set-based approach will perform better in precision on small support values, but with a price in time.



(a) Time Difference on Supports on CFC (b) Time Difference on Supports on CRAN

Fig. 4 Effect of Apriori Support on Execution Time



(a) Performance Difference on Supports on CFC (b) Performance Difference on Supports on CRAN

Fig. 5 Effect of Apriori Support on Performance

In figure 5, our intuition concerning the improved precision for small support values is validated. An observation can be made that the performance loss is close to linear as the support increases, with the windowed approach having the best performance throughout the graph.

However, it is important to define a support value for which the compromise between precision and processing time is optimal. We propose a simple weighted average metric as defined on equation 14. That support metric considers the total model

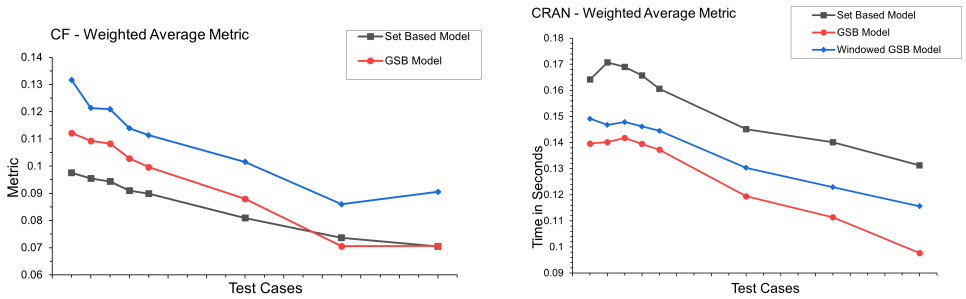
	Model	Reference	MAP on CF	MAP on CRAN
1	Set Based Model	Possas et al (2002)	0.161875265	0.19446149
2	Graphical Extension of SB	Kalogeropoulos et al (2020)	0.186186087	0.17721557
3	BM25	Robertson et al (2004)	0.222656198	**0.3532287**
4	Graph of Words	Rousseau and Vazirgiannis (2013)	0.117287827	0.104560648
5	Windowed Extension	Proposed Method	0.218826299	0.21731670
6	Windowed Model Boosted by Keywords	Proposed Method	0.211430520	0.208607620
7	Ensemble Windowed Model-ColBERT	Proposed Method	**0.23112741**	-

Table 2 Comparison of Models on Their Best Case at CFC and CRAN.

time (indexing and query answering), as well as, the MAP of the model.

$$Support\ metric_i = \alpha \cdot \frac{1}{total\ time} + \beta \cdot MAP\ of\ model_i \tag{14}$$

The parameters α and β are importance variables, that sum to 1. Intuitively, we consider the precision of the model as more important, thus β equals 0.7.



(a) Support Metric on CFC

(b) Support Metric on CRAN

Fig. 6 Support Choice Metric

Excluding the $Support = 1$, value, a proposed range for the support as depicted by the metrics graph 6 is $[3, 5]$. It is important to note that this section does not compare the performance among the models, but effect of the Apriori minimum frequency - support on each model. Since we wanted to perform a large in size set of experiments escaping large time delays, we exploited the similarity of the loss in performance of the set-based extensions, and we conducted the experiments with support equal to 10.

4.3 Models Performance on Ranking

As table 2 shows, in the best case, having fine-tuned the support and the window size the BM25 outperforms our proposed models by 0.003829899 on the Cystic Fibrosis collection. That means that our approach is on par with one of the state-of-the-art models on the sparse retrieval aspect. The percentage windowed extension performs almost the same as the constant windowed, thus it is omitted. Emphasizing the queries, by comparison of each average precision, the BM25 model shows greater results on 56

queries versus 44 of the graph-based extension. In every experiment conducted, only the title of each document and the text (i.e abstracts) were considered as valid information. Any other provided meta-data were excluded. Therefore we can conclude that our experiments on the Cystic Fibrosis collection are according to the reported data (Rivas et al (2014), Trotman (2004), Trotman (2005)) or even better, since the document collection is slightly preprocessed. Stemming-lemmatizing and stopword filtering operations were omitted in order to gauge the impact on the models' performance. As we will discuss in a later subsection stopwords are crucial for graph-based extensions, especially when windows are applied.

The notable difference between each model's precision is in well-formed queries with small relevant document lists. A well-formed query will be considered a query with a reasonably large amount of impactful terms. In such queries, the BM25 model shows a slight superiority on the Cystic Fibrosis collection and highlighted especially in the Cranfield collection (see table 2) which the majority of the queries have such characteristics. On the other hand, the graph-based extension shows a slightly improved performance on very small queries bloated with stopwords, or on very large queries with an abundance of relevant documents.

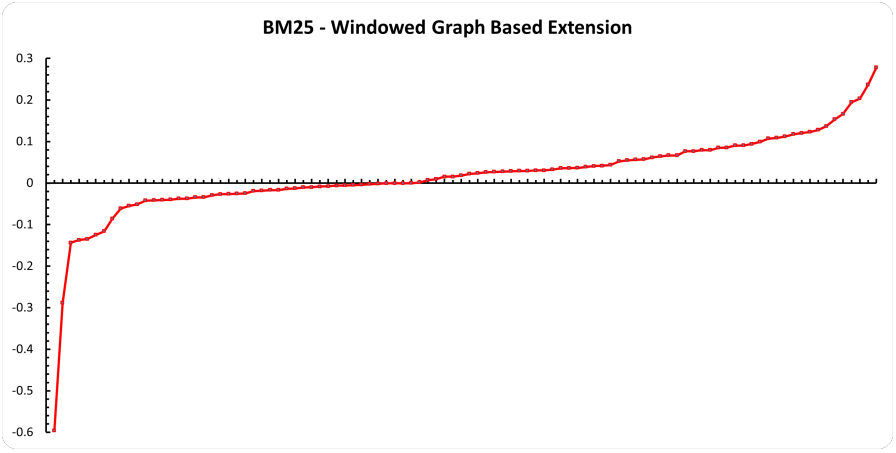
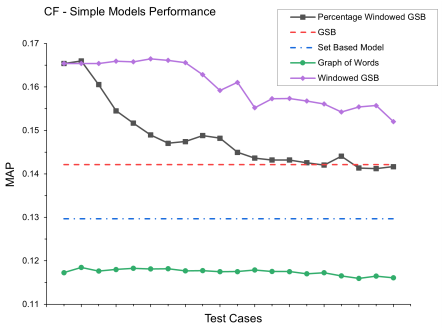


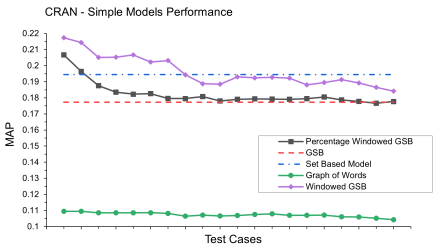
Fig. 7 Comparison of BM25 with the Graph-based Extension on CFC.

Furthermore, the graph-based extensions allow us to elaborate on other NLP tasks besides the retrieval, which is an advantage over the simple BM25. It is important to notice that the set-based model's performance increases when amplified by the aforementioned extension.

In figure 8(a) the MAP of each model is depicted on the Cystic Fibrosis collection. As we move away from the start of the x-axis the window size is increasing. Therefore for very small values on the percentage case, the constant and percentage window cases perform the same. On the other hand, on large values, the percentage model tends to perform as the complete graph-based extension of the set-based model (GSB) (Kalogeropoulos et al (2020)). Figure 8(b) demonstrates an enhancement in



(a) Precision of Proposed Models on CFC.



(b) Precision of Proposed Models on CRAN

Fig. 8 Precision of Proposed Models

the performance of the Set-Based Model compared to the basic Graph-Based Extension. However, windowed approaches, particularly with smaller window sizes, achieve the best performance.

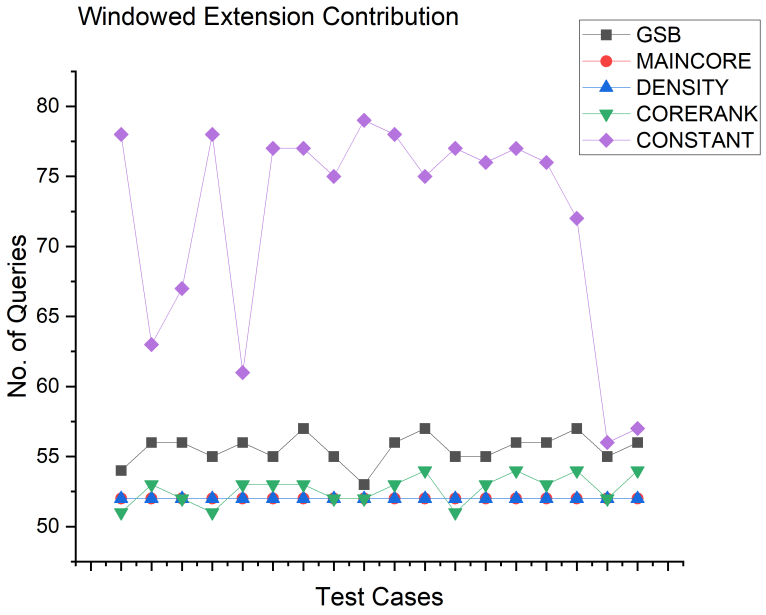


Fig. 9 Difference with the Set-Based Model of Proposed Models on CFC.

To further amplify our case, taking into consideration that the MAP metric could be misleading, we counted the number of queries that our proposed models outperformed the set-based model (Possas et al (2002)). That is expressed by the difference

in average precision for each query between the set-based model and the rest, as it is depicted in figure 9. The windowed case achieves improved results at 81/100 queries. The same figure depicts the behaviour of models implementing keyword highlight proposed by Kalogeropoulos et al (2020), Tixier et al (2016), and Rousseau and Vazirgiannis (2015).

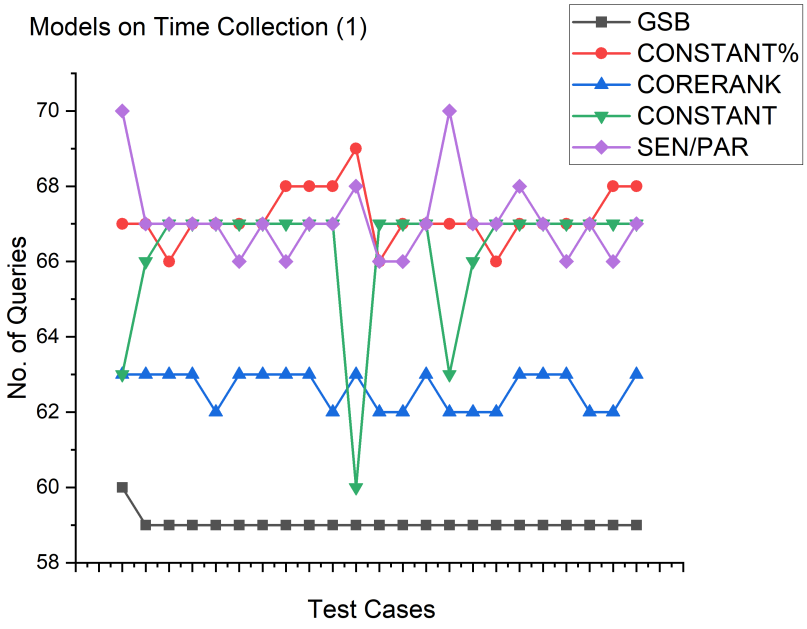


Fig. 10 Difference with the Set Based model of Proposed Models on TC.

For reproducibility, the same process is adapted to the Time collection (University of Glasgow (2011)). Figure 10 depicts the difference between the proposed models and the set-based. A noticeable observation of this collection stands on the sentence-paragraph extension, which allows us to conclude that in collections with large documents that model produces positive results. In any other case, due to the small size of the texts, the model's behaviour tends to be similar with the complete one.

Finally, in figure 11, we elaborate on the importance of windows and pruning edges on the Union Graph. We consider the complete case and then remove edges, with weight, less than a threshold for each experiment on the Cystic Fibrosis collection. That process allows for more coherent graphs, whose information can be implemented in various applications, such as node-terms conceptualization via graph clustering algorithms. The community detection problem on textual graphs can be transferred as a word semantic proximity detection issue.

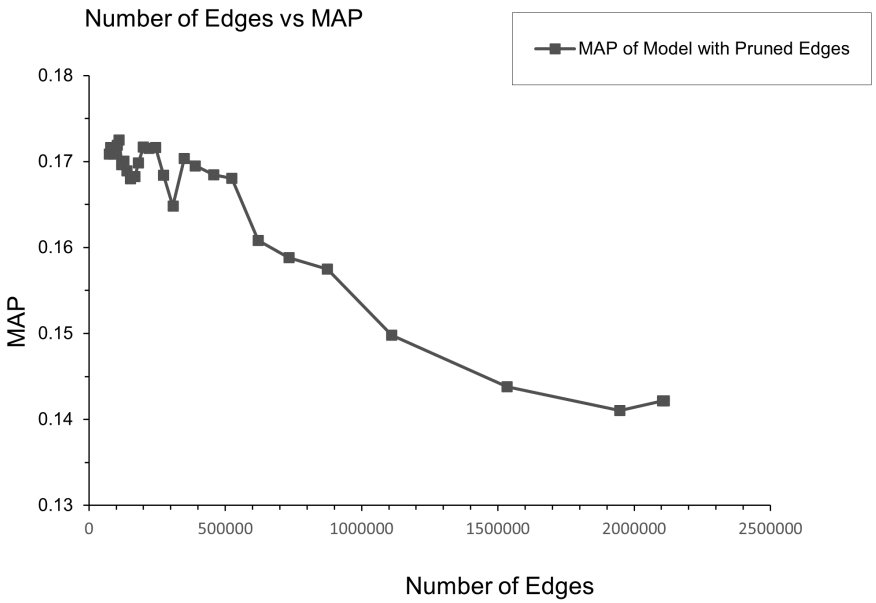


Fig. 11 Effect of Number of Edges on MAP.

The model performs better in capturing the relationship among terms in the collection, when the Union Graph consists of less than 500.000 edges, thus concluding that any form of meaningful edge removal is deemed necessary on the complete graph case.

4.4 Performance of Stopword Detection Algorithm

Before the stopwords detection solution, we conducted experiments, to gauge the impact of such words in our model. We concluded that they positively affect the process. The main and most probable reason is that graphs, that contain stopwords capture the syntactic notion of the text, but also act as a hub node, that belongs to many cliques, thus not allowing the rational path graph to be separated on complete subgraphs on a large document scale. Although, by removing stopwords on queries we can lower the computational complexity of the apriori algorithm at the retrieval phase. Such words appear frequently in text thus creating an unmeaningful amount of unnecessary termsets.

The Time collection offers a stopword list, which contains 340 expert-chosen words. We sampled that collection by hashing the documents' IDs into buckets of text. Thereafter, we implemented the ranking proposed by equation 13. The window size was chosen as 6 or 9, which yields the best results on the specific collection, and the sample size was 12%, 16%, 25%, and 33% of the total collection documents.

Table 3 depicts the precision of stopwords at the k top words of our ranking. In that particular case, the number of samples was 6, the window size was 9 and the parameter α was 5. It is important to notice that at the first 100 words, our method dictates 90% of the collection of stopwords. Moreover, the behaviour among the samples is

No of Words	Bucket 1	Bucket 2	Bucket 3	Bucket 4
10	0,9	0,9	0,9	0,9
25	0,96	0,96	0,96	0,96
50	0,96	0,94	0,94	0,96
100	0,9	0,89	0,87	0,87
200	0,72	0,73	0,71	0,725
340	0,61	0,63	0,57	0,59
450	0,52	0,54	0,50	0,508
1000	0,287	0,287	0,289	0,282

Table 3 Precision for Collection Samples on TC

similar. Therefore, we conclude that one sample can represent the collection on the stopword detection problem. The difference, or the lack of it, among the samples is definite enough in Fig. 12, where we increase the number of samples - buckets to 6.

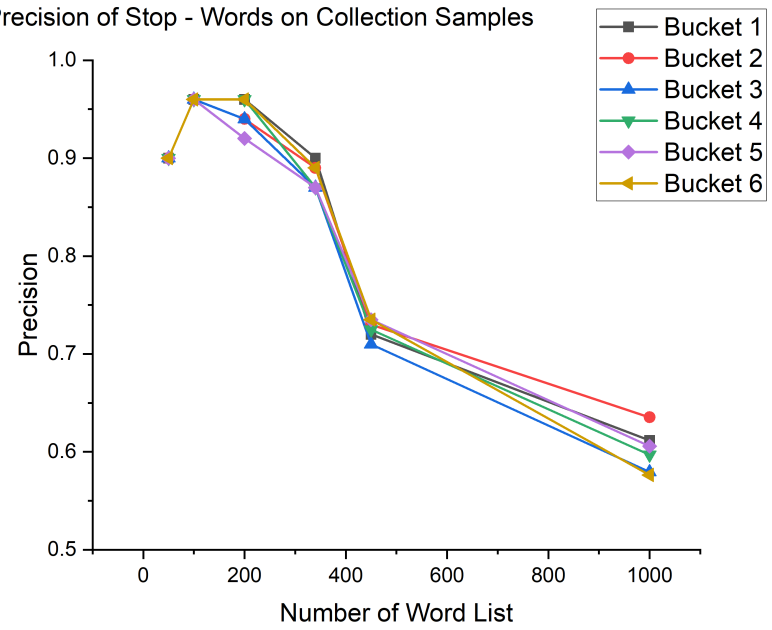


Fig. 12 Stopword Detection Precision on Collection Samples.

4.5 Performance of Ensemble and Reranking Methods

On the information retrieval model combinations, we made an effort to deviate from the reranking aspect of the current research tendency to model ensembles; hence we experimented with both alternatives. In section 3.3, a simple algorithm based on

Borda count was presented, which will be implemented in the following experiments, for ensembling.

The focal point of our attempt is to estimate empirically, the impact of fusion models implementing graph textual representations, as well as the effect of such models in approaches that are considered state-of-the-art such as the ColBERT (Khattab and Zaharia (2020)). To be more specific, our goal is not to propose an ensemble that will outperform the state-of-the-art, but a method that will restrain the model when it fails to perform in some queries.

Staring at the first issue at hand, the performance of the ensemble is depicted in figure 13. Every model that implements association rules mining for termset creation has support defined as 10. the simple unique models are considered as the baseline for comparison. An important observation is that the ensemble of the set-based model with the constant window has a smaller decline slope. That observation, though, is more important in the ensemble of models with different constant window sizes. That ensemble could be considered window agnostic, achieving the highest overall performance on larger window sizes, although, on small window size the simple window case is superior. Such observation allows our proposed models to be implemented on collections without the need for parameter fine-tuning.

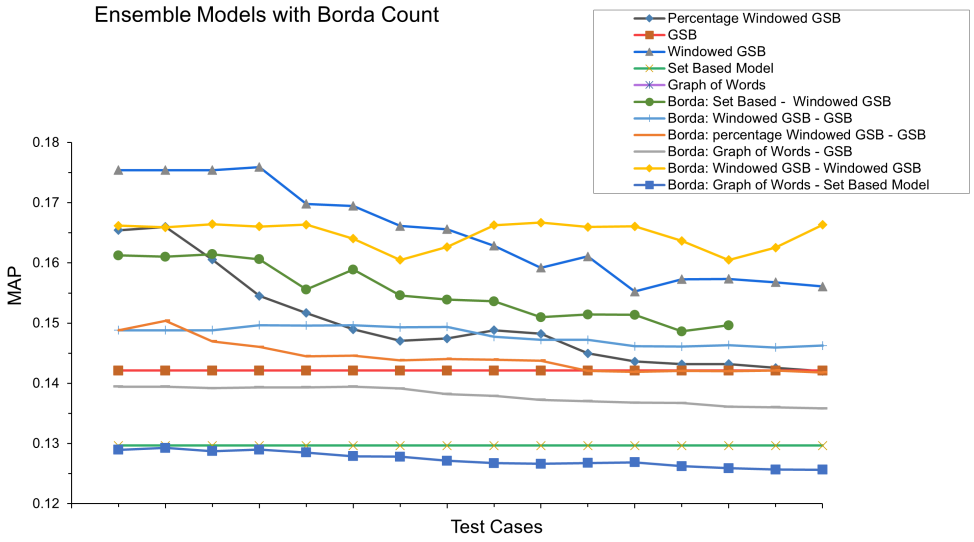


Fig. 13 Precision of Ensemble Models on CFC.

As mentioned before, the window size parameter is regarded as of crucial importance, and its fine-tuning can render a model obsolete, if the parameterization is hard. Therefore, the size defined by a percentage of a document size or a multiple-windowed ensemble approach can be considered as a solution to that issue.

Figure 14 elaborates on the performance of the fusion models. We clearly can validate the aforementioned observations about the newly found stability of the windowed

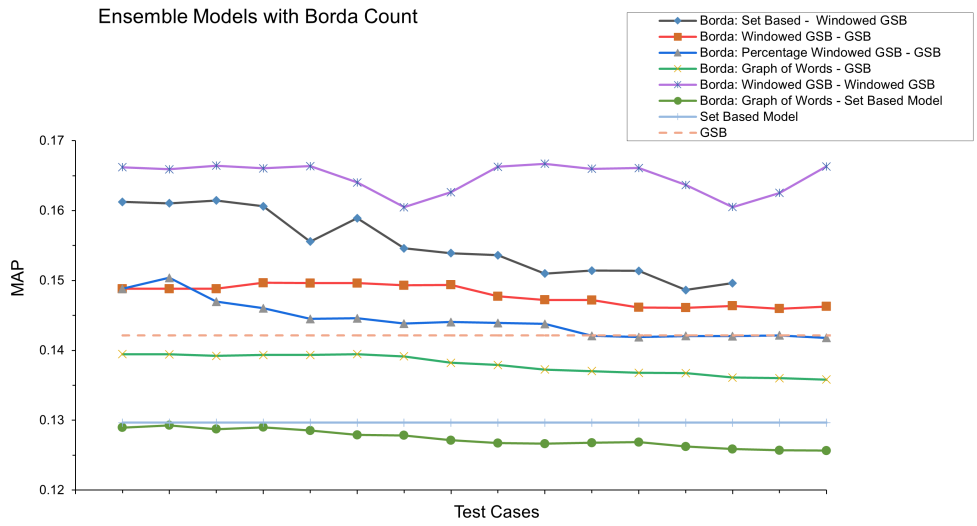


Fig. 14 Precision of Windowed Ensembles on CFC.

models. However, the rest of the fusion models tend to have similar performance to the inferior model.

In this paragraph, ColBERT's performance will be discussed in comparison with the BM25 model and our approach. We calculated the average precision difference between ColBERT and BM25 and our windowed approach, with window size of 7, on the Cystic Fibrosis collection. The BM25 model overall outperforms the ColBERT at 60 queries and our approach at 50 queries. As figure 15 depicts, the graph-based extension on the superior cases against the ColBERT is more effective than the BM25. Although the reverse statement has its merits. To simplify the above, when the BM25 is superior to the ColBERT, its performance is better than the windowed approach (wGSB), and when the windowed model is more effective than the ColBERT is more effective than the BM25 model as well.

On the reranking aspect of the experiment, we implemented as a fist-stage ranker the windowed graph-based extension of the set-based model, having window size at 7 terms defined. We consider that the first 400 documents in the sorted list are the most relevant for each query. Therefore, a new collection containing those documents was the input for the ColBERT model, handling the reranking process.

Figures 15 and 16 depict that the rerank and ensemble methods capture the positive impact of the ColBERT model, thus producing better results than the windowed approach in cases where the ColBERT outperforms the rest of the models. Therefore our proposed windowed method is capable of improving performance, especially when used in ensemble and reranking purposes, providing information crucial to the retrieval process.

To validate further our approach, we tested the statistical significance of our results by conducting 2-tailed paired t-tests; our tests produced small p-values, in the worst case at most 0.000952339.

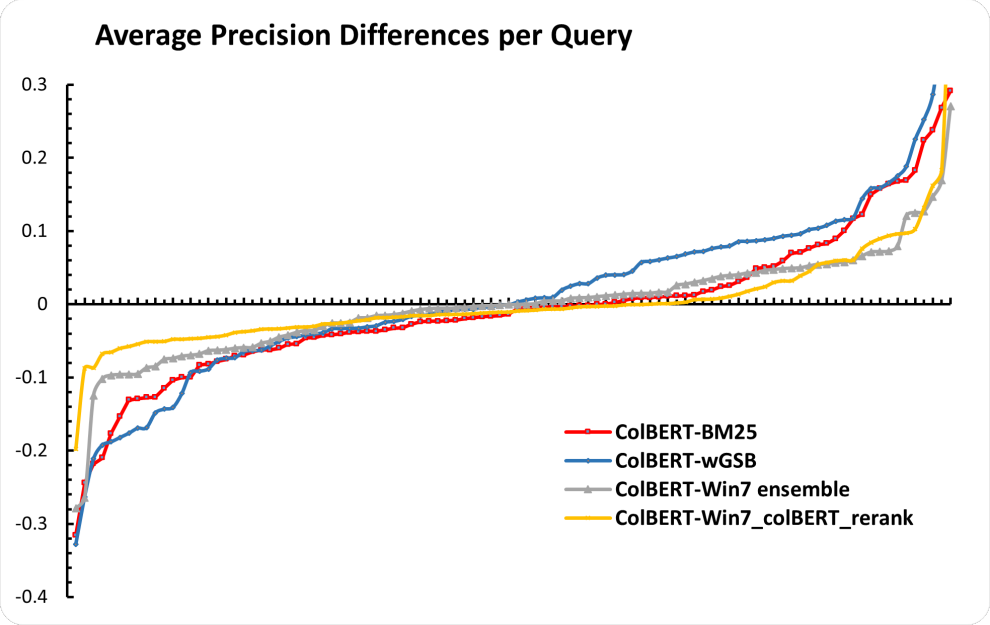


Fig. 15 Comparison of BM25 and Windowed Graph Based Extension with ColBERT on CFC per Query.

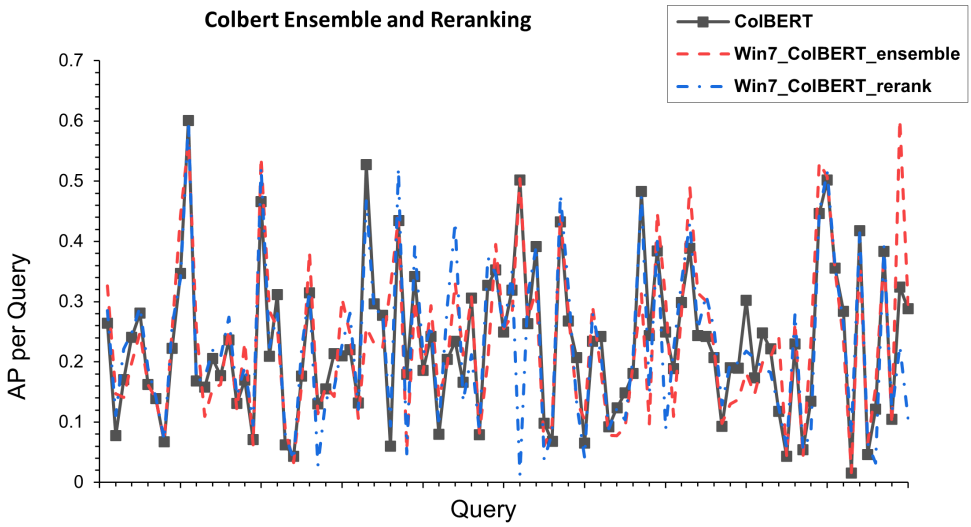


Fig. 16 Comparison of ColBERT on Ensemble and Rerank Tasks on CFC per Query.

5 Conclusions and Future Work

In this paper, we proposed improved methods for a graphical extension of the set-based model, elaborating on the importance of textual segments, in such algorithms.

We focused on a variety of related options, such as simple windows, a single window, or multiple, that capture the sentence-paragraph relation. Moreover, by considering the start and end of a window as the punctuation point we indicate that a strict sentence approximation might be futile. Finally, we approached the terms or node importance on a graphical aspect, classifying irrelevant terms as stopwords, on collection size, having a 90% precision on the first 100 purposed stopwords.

Our model is in parallel with a BM25, a well-known and widely used model for Information Retrieval. To be more specific, it is implemented as a first-stage ranker on models that apply transformer-based reranking in later stages. Therefore, our approach, as BM25, can be used as such. We evaluated the performance of our approaches in multiple collections with a variety of characteristics. An attempt is made to elaborate on the effect of each model's parameter and proposed methods and thought processes for choosing suitable values. Finally, a framework of model ensemble is proposed to boost the stability and effectiveness of our approaches.

In further research, the focal points should aim at decreasing the complexity of the proposed algorithms by modifying them or implementing parallelism through a distributed system, that will allow for more graphical processing. Furthermore, the node embeddings proposed by Grover and Leskovec (2016) or even word representations found in the work of Mikolov et al (2013) allow information encoding on real-valued vectors. By inducing such on-edge weights, structural and semantic information will be taken into consideration by our model. Moreover, embeddings will allow machine learning models to be applied simultaneously with graph features, such as centrality measures or core/truss numbers. Finally, the research focus has turned more to deep learning methods and algorithms, thus an effort to employ Graph Neural Networks (GNNs) (Wu et al (2022)) on graphs derived from textual data can be made, in order to exploit nodes' structural and semantic relationships.

References

Agrawal R, Srikant R (1994) Fast algorithms for mining association rules in large databases. In: Proceedings of the 20th International Conference on Very Large Data Bases. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, VLDB '94, pp 487–499, URL http://dl.acm.org/citation.cfm?id=645920.672836

Anava Y, Shtok A, Kurland O, et al (2016) A probabilistic fusion framework. In: Proceedings of the 25th ACM International on Conference on Information and Knowledge Management. Association for Computing Machinery, New York, NY, USA, CIKM '16, p 1463–1472, https://doi.org/10.1145/2983323.2983739, URL https://doi.org/10.1145/2983323.2983739

Blanco R, Lioma C (2012) Graph-based term weighting for information retrieval. Inf Retr 15:54–92. https://doi.org/10.1007/s10791-011-9172-x

de Borda JC (1781) Mémoire sur les élections au scrutin. Histoire de l'Académie Royale des Sciences

Bordag S, Heyer G, Quasthoff U (2003) Small worlds of concepts and other principles of semantic search. In: Böhme T, Heyer G, Unger H (eds) Innovative Internet Community Systems. Springer Berlin Heidelberg, Berlin, Heidelberg, pp 10–19

Ferrer i Cancho R, Solé RV, Köhler R (2004) Patterns in syntactic dependency networks. Phys Rev E 69:051915. https://doi.org/10.1103/PhysRevE.69.051915

Ferrer i Cancho R, Capocci A, Caldarelli G (2007) Spectral methods cluster words of the same class in a syntactic dependency network. International Journal of Bifurcation and Chaos 17(07):2453–2463. https://doi.org/10.1142/S021812740701852X, URL https://doi.org/10.1142/S021812740701852X, https://doi.org/10.1142/S021812740701852X

Cohen J (2008) Trusses: Cohesive subgraphs for social network analysis. National security agency Technical Report 16(3.1):1–29. URL https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=05d8623c39e9b92f771fa66ca7d3eef03cf859bb

Datar M, Gionis A, Indyk P, et al (2002) Maintaining stream statistics over sliding windows. SIAM Journal on Computing 31(6):1794–1813. https://doi.org/10.1137/S0097539701398363, URL https://epubs.siam.org/doi/10.1137/S0097539701398363

De Filippis GM, Rinaldi AM, Russo C, et al (2024) Enhanced semantic understanding with graph-based information retrieval. In: Proceedings of the IRon-Graphs: International Workshop on Graph-Based Approaches in Information Retrieval, URL https://www.researchgate.net/publication/384801189_Enhanced_Semantic_Understanding_with_Graph-Based_Information_Retrieval

Devlin J, Chang MW, Lee K, et al (2018) Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:181004805

Firth JR (1957) A synopsis of linguistic theory 1930-55. Studies in Linguistic Analysis (special volume of the Philological Society) 1952-59:1–32

Grover A, Leskovec J (2016) node2vec: Scalable feature learning for networks. In: Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining, pp 855–864

Kalogeropoulos NR, Doukas I, Makris C, et al (2020) A graph-based extension for the set-based model implementing algorithms based on important nodes. In: Maglogiannis I, Iliadis L, Pimenidis E (eds) Artificial Intelligence Applications and Innovations. AIAI 2020 IFIP WG 12.5 International Workshops. Springer International Publishing, Cham, pp 143–154

Kalogeropoulos NR, Kontogiannis G, Makris C (2025) Spectral clustering and query expansion using embeddings on the graph-based extension of the set-based information retrieval model. Expert Systems with Applications 263:125771. https://doi.org/

https://doi.org/10.1016/j.eswa.2024.125771, URL https://www.sciencedirect.com/science/article/pii/S0957417424026381

Khalid M, Verberne S (2008) Passage retrieval for question answering using sliding windows. In: Coling 2008: Proceedings of the 2nd workshop on Information Retrieval for Question Answering. Coling 2008 Organizing Committee, Manchester, UK, pp 26–33, URL https://aclanthology.org/W08-1804.pdf

Khattab O, Zaharia M (2020) 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

Lin J, Nogueira R, Yates A (2022) Pretrained transformers for text ranking: Bert and beyond. Springer Nature

Mallia A, Khattab O, Suel T, et al (2021) Learning passage impacts for inverted indexes. In: Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, pp 1723–1727

Markovits G, Shtok A, Kurland O, et al (2012) Predicting query performance for fusion-based retrieval. In: Proceedings of the 21st ACM international conference on Information and knowledge management, pp 813–822

Medeiros Soares M, Corso G, Lucena L (2005) The network of syllables in portuguese. Physica A: Statistical Mechanics and its Applications 355(2):678–684. https://doi.org/https://doi.org/10.1016/j.physa.2005.03.017

Mihalcea R, Tarau P (2004) TextRank: Bringing order into text. In: Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics, Barcelona, Spain, pp 404–411, URL https://aclanthology.org/W04-3252

Mikolov T, Chen K, Corrado G, et al (2013) Efficient estimation of word representations in vector space. arXiv preprint arXiv:13013781

Mongiovì M, Gangemi A (2024) Graal: Graph-based retrieval for collecting related passages across multiple documents. Information 15(6). https://doi.org/10.3390/info15060318, URL https://www.mdpi.com/2078-2489/15/6/318

Nogueira R, Yang W, Cho K, et al (2019) Multi-stage document ranking with bert. arXiv preprint arXiv:191014424

Page L, Brin S, Motwani R, et al (1999) The pagerank citation ranking: Bringing order to the web. Technical Report 1999-66, Stanford InfoLab, URL http://ilpubs.stanford.edu:8090/422/, previous number = SIDL-WP-1999-0120

Peng B, Zhu Y, Liu Y, et al (2024) Graph retrieval-augmented generation: A survey. arXiv preprint arXiv:240808921 URL https://arxiv.org/abs/2408.08921

Possas B, Ziviani N, Meira WJr., et al (2002) Set-based model: A new approach for information retrieval. In: Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, IGIR '02, pp 230–237, https://doi.org/10.1145/564376.564417, URL http://doi.acm.org/10.1145/564376.564417

Pôssas B, Ziviani N, Meira W, et al (2005) Set-based vector model: An efficient approach for correlation-based ranking. ACM Trans Inf Syst 23:397–429

Rabinovich E, Rom O, Kurland O (2014) Utilizing relevance feedback in fusion-based retrieval. In: Proceedings of the 37th International ACM SIGIR Conference on Research & Development in Information Retrieval. Association for Computing Machinery, New York, NY, USA, SIGIR '14, p 313–322, https://doi.org/10.1145/2600428.2609573, URL https://doi.org/10.1145/2600428.2609573

Rivas AR, Iglesias EL, Borrajo L (2014) Study of query expansion techniques and their application in the biomedical information retrieval. ScientificWorldJournal 2014:132158. https://doi.org/10.1155/2014/132158

Robertson SE, Sparck Jones K (1977) Relevance weighting of search terms. Journal of the American Society for Information Science

Robertson SE, Walker S (1994) Some simple effective approximations to the 2-poisson model for probabilistic weighted retrieval. In: Proceedings of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, ACM/Springer, pp 232–241

Robertson SE, Zaragoza H, Taylor M (2004) Simple bm25 extension to multiple weighted fields. In: Proceedings of the 2004 ACM CIKM International Conference on Information and Knowledge Management, ACM, pp 42–49

Rousseau F, Vazirgiannis M (2013) Graph-of-word and tw-idf: New approach to ad hoc ir. In: Proceedings of the 22Nd ACM International Conference on Information and Knowledge Management. ACM, CIKM '13, pp 59–68, https://doi.org/10.1145/2505515.2505671

Rousseau F, Vazirgiannis M (2015) Main core retention on graph-of-words for single-document keyword extraction. In: Advances in Information Retrieval. Springer International Publishing, pp 382–393

Seidman SB (1983a) Network structure and minimum degree. Social Networks 5(3):269 – 287. https://doi.org/https://doi.org/10.1016/0378-8733(83)90028-X

Seidman SB (1983b) Network structure and minimum degree. Social Networks 5(3):269 – 287. https://doi.org/https://doi.org/10.1016/0378-8733(83)90028-X

Shaw MW., Wood BJ., Wood BJ., et al (1991) The cystic fibrosis database: Content and research opportunities.

Sonawane S, Kulkarni P (2014) Graph based representation and analysis of text document: A survey of techniques. International Journal of Computer Applications 96:1–8. https://doi.org/10.5120/16899-6972

Sparck Jones K, Walker S, Robertson SE (2000) A probabilistic model of information retrieval: Development and comparative experiments. part 2. Information Processing and Management pp 809–840

Tixier A, Malliaros F, Vazirgiannis M (2016) A graph degeneracy-based approach to keyword extraction. In: Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics, Austin, Texas, pp 1860–1870, https://doi.org/10.18653/v1/D16-1191, URL https://aclanthology.org/D16-1191

Trotman A (2004) An artificial intelligence approach to information retrieval. In: SIGIR, Citeseer, pp 603–608

Trotman A (2005) Learning to rank. Information Retrieval 8:359–381

Tyomkin L, Kurland O (2023) Revisiting condorcet fusion. In: Proceedings of the 2023 ACM SIGIR International Conference on Theory of Information Retrieval. Association for Computing Machinery, New York, NY, USA, ICTIR '23, p 199–204, https://doi.org/10.1145/3578337.3605140, URL https://doi.org/10.1145/3578337.3605140

University of Glasgow (2011) Test collections. http://ir.dcs.gla.ac.uk/resources/test_collections/, online; accessed 21/12/2023

Vaswani A, Shazeer N, Parmar N, et al (2017) Attention is all you need. Advances in neural information processing systems 30

Wang P, Shi R, et al (2024) Grand: A fast and accurate graph retrieval framework via knowledge distillation. In: Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval. ACM, pp 1639–1648

Wu L, Cui P, Pei J, et al (2022) Graph Neural Networks, Springer Nature Singapore, Singapore, pp 27–37. https://doi.org/10.1007/978-981-16-6054-2_3, URL https://doi.org/10.1007/978-981-16-6054-2_3

Young HP (1988) Condorcet's theory of voting. The American Political Science Review 82(4):1231–1244. URL http://www.jstor.org/stable/1961757

Zhang C, Bonifati A, Özsu MT (2024) Incremental sliding window connectivity over streaming graphs. arXiv preprint arXiv:240606754 URL https://arxiv.org/abs/2406.06754

Zhou F, Mahler S, Toivonen H (2012) Simplification of Networks by Edge Pruning, Springer Berlin Heidelberg, pp 179–198. https://doi.org/10.1007/978-3-642-31830-6_13