

*This is the unrefereed Author's Original Version (or pre-print Version) of the article. The present version is not the Accepted Manuscript.*

## Query-oriented text summarization based on hypergraph transversals

Hadrien Van Lierde and Tommy W. S. Chow

*Department of Electronic Engineering, City University of Hong Kong  
83 Tat Chee Av., Kowloon Tong, Hong Kong, China  
hadrien.vanlierde@hotmail.com, eetchow@cityu.edu.hk*

### Abstract

Existing graph- and hypergraph-based algorithms for document summarization represent the sentences of a corpus as the nodes of a graph or a hypergraph in which the edges represent relationships of lexical similarities between sentences. Each sentence of the corpus is then scored individually, using popular node ranking algorithms, and a summary is produced by extracting highly scored sentences. This approach fails to select a subset of *jointly* relevant sentences and it may produce redundant summaries that are missing important topics of the corpus. To alleviate this issue, a new hypergraph-based summarizer is proposed in this paper, in which each node is a sentence and each hyperedge is a theme, namely a group of sentences sharing a topic. Themes are weighted in terms of their prominence in the corpus and their relevance to a user-defined query. It is further shown that the problem of identifying a subset of sentences covering the relevant themes of the corpus is equivalent to that of finding a hypergraph transversal in our theme-based hypergraph. Two extensions of the notion of hypergraph transversal are proposed for the purpose of summarization, and polynomial time algorithms building on the theory of submodular functions are proposed for solving the associated discrete optimization problems. The worst-case time complexity of the proposed algorithms is squared in the number of terms, which makes it cheaper than the existing hypergraph-based methods. A thorough comparative analysis with related models on DUC benchmark datasets demonstrates the effectiveness of our approach, which outperforms existing graph- or hypergraph-based methods by at least

6% of ROUGE-SU4 score.

**keywords:** Query-Oriented Text Summarization, Hypergraph Theory, Hypergraph Transversal, Sentence Clustering, Submodular Set Functions

# 1 Introduction

The development of automatic tools for the summarization of large corpora of documents has attracted a widespread interest in recent years. With fields of application ranging from medical sciences to finance and legal science, these summarization systems considerably reduce the time required for knowledge acquisition and decision making, by identifying and formatting the relevant information from a collection of documents. Since most applications involve large corpora rather than single documents, summarization systems developed recently are meant to produce summaries of multiple documents. Similarly, the interest has shifted from generic towards query-oriented summarization, in which a query expresses the user’s needs. Moreover, existing summarizers are generally extractive, namely they produce summaries by extracting relevant sentences from the original corpus.

Among the existing extractive approaches for text summarization, graph-based methods are considered very effective due to their ability to capture the global patterns of connection between the sentences of the corpus. These systems generally define a graph in which the nodes are the sentences and the edges denote relationships of lexical similarities between the sentences. The sentences are then scored using graph ranking algorithms such as the PageRank [1] or HITS [2] algorithms, which can also be adapted for the purpose of query-oriented summarization [3]. A key step of graph-based summarizers is the way the graph is constructed, since it has a strong impact on the sentence scores. As pointed out in [4], a critical issue of traditional graph-based summarizers is their inability to capture group relationships among sentences since each edge of a graph only connects a pair of nodes.

Following the idea that each topic of a corpus connects a group of multiple sentences covering that topic, hypergraph models were proposed in [4] and [5], in which the hyperedges represent similarity relationships among groups of sentences. These group relationships are formed by detecting clusters of lexically similar sentences we refer to as *themes* or *theme-based hyperedges*. Each theme is believed to cover a specific topic of the corpus. However, since the models of [4] and [5] define the themes as groups of lexically similar sentences, the underlying topics are not explicitly discovered. Moreover, their themes do not overlap which contradicts the fact that each sentence carries multiple information and may thus belong to multiple themes, as can be seen from the following example of sentence.

”Once John finished studying for his school test the next day, he caught up with his friend at the sport centre and they played soccer together.”

Two topics are covered by the sentence above: the topics of *studies* and *leisure*. Hence, the sentence should belong to multiple themes simultaneously, which is not allowed in existing hypergraph models of [4] and [5].

The hypergraph model proposed in this paper alleviates these issues by first extracting topics, i.e. groups of semantically related terms, using a new topic model referred to as *SEMCOT*. Then, a theme is associated to each topic, such that each theme is defined as the group of sentences covering the associated topic. Finally, a hypergraph is formed with sentences as nodes, themes as hyperedges and hyperedge weights reflecting the prominence of each theme and its relevance to the query. In such a way, our model alleviates the weaknesses of existing hypergraph models since each theme-based hyperedge is associated to a specific topic and each sentence may belong to multiple themes.

Furthermore, a common drawback of existing graph- and hypergraph-based summarizers is that they select sentences based on the computation of an individual relevance score for each sentence. This approach fails to capture the information jointly carried by the sentences which results in redundant summaries missing important topics of the corpus. To alleviate this issue, we propose a new approach of sentence selection using our theme-based hypergraph. A minimal hypergraph transversal is the smallest subset of nodes covering all hyperedges of a hypergraph [6]. The concept of hypergraph transversal is used in computational biology [7] and data mining [6] for identifying a subset of relevant agents in a hypergraph. In the context of our theme-based hypergraph, a hypergraph transversal can be viewed as the smallest subset of sentences covering all themes of the corpus. We extend the notion of transversal to take the theme weights into account and we propose two extensions called *minimal soft hypergraph transversal* and *maximal budgeted hypergraph transversal*. The former corresponds to finding a subset of sentences of minimal aggregated length and achieving a *target coverage* of the topics of the corpus (in a sense that will be clarified). The latter seeks a subset of sentences maximizing the total weight of covered hyperedges while not exceeding a *target summary length*. As the associated discrete optimization problems are NP-hard, we propose two approximation algorithms building on the theory of submodular functions. Our transversal-based approach for sentence selection alleviates the drawback of methods of individual sentence scoring, since it selects a set of sentences that are jointly covering a maximal number of relevant themes and produces informative and non-redundant summaries. As demonstrated in the paper, the time complexity of the method is equivalent to that of early graph-based summarization systems such as LexRank [1], which makes it more efficient than existing hypergraph-based summarizers [4, 5]. The scalability of summarization algorithms is essential, especially in applications involving large corpora such as the summarization of news reports [8] or the summarization of legal texts [9].

The method of [10] proposes to select sentences by using a maximum coverage approach, which shares some similarities with our model. However, they attempt to select a subset of sentences maximizing the number of relevant terms covered by the sentences. Hence, they fail to capture the topical relationships among sentences, which are, in contrast, included in our theme-based hypergraph.

A thorough comparative analysis with state-of-the-art summarization systems is included in the paper. Our model is shown to outperform other models on a benchmark dataset produced by the *Document Understanding Conference*. The main contributions of this paper are (1) a new topic model extracting groups of semantically related terms based on patterns of term co-occurrences, (2) a natural hypergraph model representing nodes as sentences and each hyperedge as a theme, namely a group of sentences sharing a topic, and (3) a new sentence selection approach based on hypergraph transversals for the extraction of a subset of jointly relevant sentences.

The structure of the paper is as follows. In section 2, we present work related to our method. In section 3, we present an overview of our system which is described in further details in section 4. Then, in section 5, we present experimental results. Finally, section 6 presents a discussion and concluding remarks.

## 2 Background and related work

While early models focused on the task of single document summarization, recent systems generally produce summaries of corpora of documents [11]. Similarly, the focus has shifted from generic summarization to the more realistic task of query-oriented summarization, in which a summary is produced with the essential information contained in a corpus that is also relevant to a user-defined query [12].

Summarization systems are further divided into two classes, namely abstractive and extractive models. Extractive summarizers identify relevant sentences in the original corpus and produce summaries by aggregating these sentences [11]. In contrast, an abstractive summarizer identifies conceptual information in the corpus and reformulates a summary from scratch [12]. Since abstractive approaches require advanced natural language processing, the majority of existing summarization systems consist of extractive models.

Extractive summarizers differ in the method used to identify relevant sentences, which leads to a classification of models as either feature-based or graph-based approaches. Feature-based methods represent the sentences with a set of predefined features such as the sentence position, the sentence length or the presence of cue phrases [13]. Then, they train a model to compute relevance scores for the sentences based on their features. Since feature-based approaches generally require datasets with labelled sentences which are hard to produce [12], unsupervised graph-based methods have attracted growing interest in recent years.

Graph-based summarizers represent the sentences of a corpus as the nodes of a graph with the edges modelling relationships of similarity between the sentences [1]. Then, graph-based algorithms are applied to identify relevant sentences. The models generally differ in the type of relationship captured by the graph or in the sentence selection approach. Most graph-based models define the edges connecting sentences based on the co-occurrence of terms in pairs of sentences [1, 3, 4]. Then, important sentences are identified either based on node ranking algorithms, or using a global optimization approach. Methods based on node ranking compute individual relevance scores for the sentences and build summaries with highly scored sentences. The earliest such summarizer, LexRank [1], applies the PageRank algorithm to compute sentence scores. Introducing a query bias in the node ranking algorithm, this method can be adapted for query-oriented summarization as in [3]. A different graph model was proposed in [14], where sentences and key phrases form the two classes of nodes of a bipartite graph. The sentences and the key phrases are then scored simultaneously by applying a mutual reinforcement algorithm. An extended bipartite graph ranking algorithm is also proposed in [2] in which the sentences represent one class of nodes and clusters of similar sentences represent the other class. The hubs and authorities algorithm is then applied to compute sentence scores. Adding terms as a third class of nodes, [15] propose to score terms, sentences and sentence clusters simultaneously, based on a mutual reinforcement algorithm which propagates the scores

across the three node classes. A common drawback of the approaches based on node ranking is that they compute individual relevance scores for the sentences and they fail to model the information jointly carried by the sentences, which may result in redundant summaries. Hence, global optimization approaches were proposed to select a set of jointly relevant and non-redundant sentences as in [16] and [17]. For instance, [18] propose a greedy algorithm to find a dominating set of nodes in the sentence graph. A summary is then formed with the corresponding set of sentences. Similarly, [16] extract a set of sentences with a maximal similarity with the entire corpus and a minimal pairwise lexical similarity, which is modelled as a multi-objective optimization problem. In contrast, [10] propose a coverage approach in which a set of sentences maximizing the number of distinct relevant terms is selected. Finally, [17] propose a two step approach in which individual sentence relevance scores are computed first. Then a set of sentences with a maximal total relevance and a minimal joint redundancy is selected. All three methods attempt to solve NP-hard problems. Hence, they propose approximation algorithms based on the theory of submodular functions.

Going beyond pairwise lexical similarities between sentences and relations based on the co-occurrence of terms, hypergraph models were proposed, in which nodes are sentences and hyperedges model group relationships between sentences [4]. The hyperedges of the hypergraph capture topical relationships among groups of sentences. Existing hypergraph-based systems [4, 5] combine pairwise lexical similarities and clusters of lexically similar sentences to form the hyperedges of the hypergraph. Hypergraph ranking algorithms are then applied to identify important and query-relevant sentences. However, they do not provide any interpretation for the clusters of sentences discovered by their method. Moreover, these clusters do not overlap, which is incoherent with the fact that each sentence carries multiple information and hence belongs to multiple semantic groups of sentences. In contrast, each hyperedge in our proposed hypergraph connects sentences covering the same topic, and these hyperedges do overlap.

A minimal hypergraph transversal is a subset of the nodes of hypergraph of minimum cardinality and such that each hyperedge of the hypergraph is incident to at least one node in the subset [6]. Theoretically equivalent to the minimum hitting set problem, the problem of finding a minimum hypergraph transversal can be viewed as finding a subset of representative nodes covering the essential information carried by each hyperedge. Hence, hypergraph transversals find applications in various areas such as computational biology, boolean algebra and data mining [19]. Extensions of hypergraph transversals to include hyperedge and node weights were also proposed in [20]. Since the associated optimization problems are generally NP-hard, various approximation algorithms were proposed, including greedy algorithms [21] and LP relaxations [22]. The problem of finding a hypergraph transversal is conceptually similar to that of finding a summarizing subset of a set of objects modelled as a hypergraph. However, to the best of our knowledge, there was no attempt to use hypergraph transversals for text summarization in the past. Since it seeks a set of jointly relevant sentences, our method shares some similarities with existing graph-based models that apply global optimization strategies for sentence selection [10, 16, 17]. However, our hypergraph better captures topical relationships among sentences than the simple graphs based on lexical similarities between sentences.

### 3 Problem statement and system overview

Given a corpus of  $N_d$  documents and a user-defined query  $q$ , we intend to produce a summary of the documents with the information that is considered both central in the corpus and relevant to the query. Since we limit ourselves to the production of extracts, our task is to extract a set  $S$  of relevant sentences from the corpus and to aggregate them to build a summary. Let  $N_s$  be the total number of sentences in the corpus. We further split the task into two subtasks:

- **target summary length:** the summary must cover the largest amount of relevant information while not exceeding a target length  $L$ , namely  $\sum_{i \in S} L_i \leq L$ , where  $\{L_i, 1 \leq i \leq N_s\}$  represent the lengths of the sentences,
- **target coverage:** the summary must have a minimum length while achieving a *target coverage* of the information expressed by a parameter  $\gamma \in [0, 1]$  expressing the fraction of the information present in the corpus that must be covered by the summary (in a sense that will be clarified).

The sentences in the set  $S$  are then aggregated to form the final summary. Figure 1 summarizes the steps of our proposed method. After some preprocessing steps, the themes are detected based on a topic detection algorithm which tags each sentence with multiple topics. A theme-based hypergraph is then built with the weight of each theme reflecting both its importance in the corpus and its similarity with the query. Finally, depending on the task at hand, one of two types of hypergraph transversal is generated. If the summary must not exceed a *target summary length*, then a *maximal budgeted hypergraph transversal* is generated. If the summary must achieve a *target coverage*, then a *minimal soft hypergraph transversal* is generated. Finally the sentences corresponding to the generated transversal are selected for the summary.

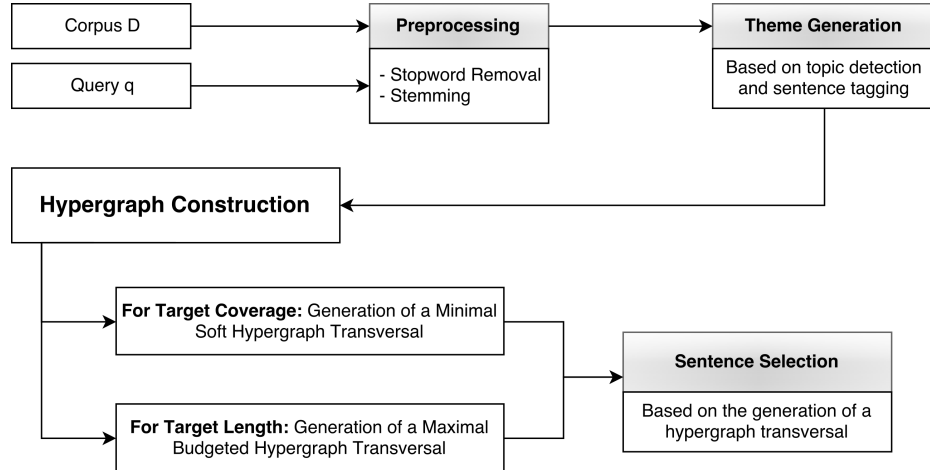


Figure 1: Algorithm Chart.

## 4 Summarization based on hypergraph transversals

In this section, we present the key steps of our algorithm: after some standard pre-processing steps, topics of semantically related terms are detected from which themes grouping topically similar sentences are extracted. A hypergraph is then formed based on the sentence themes and sentences are selected based on the detection of a hypergraph transversal.

### 4.1 Preprocessing and similarity computation

As the majority of extractive summarization approaches, our model is based on the representation of sentences as vectors. To reduce the size of the vocabulary, we remove stopwords that do not contribute to the meaning of sentences such as "the" or "a", using a publicly available list of 667 stopwords<sup>1</sup>. The words are also stemmed using Porter Stemmer [23]. Let  $N_t$  be the resulting number of distinct terms after these two preprocessing steps are performed. We define the *inverse sentence frequency*  $\text{isf}(t)$  [24] as

$$\text{isf}(t) = \log \left( \frac{N_s}{N_s^t} \right) \quad (1)$$

where  $N_s^t$  is the number of sentences containing term  $t$ . This weighting scheme yields higher weights for rare terms which are assumed to contribute more to the semantics of sentences [24]. Sentence  $i$  is then represented by a vector  $s_i = [\text{tfisf}(i, 1), \dots, \text{tfisf}(i, N_t)]$  where

$$\text{tfisf}(i, t) = \text{tf}(i, t) \text{isf}(t) \quad (2)$$

and  $\text{tf}(i, t)$  is the frequency of term  $t$  in sentence  $i$ . Finally, to denote the similarity between two text fragments  $a$  and  $b$  (which can be sentences, groups of sentences or the query), we use the cosine similarity between the  $\text{tfisf}$  representations of  $a$  and  $b$ , as suggested in [3]:

$$\text{sim}(a, b) = \frac{\sum_t \text{tfisf}(a, t) \text{tfisf}(b, t)}{\sqrt{\sum_t \text{tfisf}(a, t)^2} \sqrt{\sum_t \text{tfisf}(b, t)^2}} \quad (3)$$

where  $\text{tfisf}(a, t)$  is also defined as the frequency of term  $t$  in fragment  $a$  multiplied by  $\text{isf}(t)$ . This similarity measure will be used in section 4.3 to compute the similarity with the query  $q$ .

### 4.2 Sentence theme detection based on topic tagging

As mentioned in section 1, our hypergraph model is based on the detection of themes. A theme is defined as a group of sentences covering the same topic. Hence, our theme detection algorithm is based on a 3-step approach: the extraction of topics, the process of tagging each sentence with multiple topics and the detection of themes based on topic tags.

---

<sup>1</sup>Stopword Lists by Ranks NL Webmaster Tools, <https://www.ranks.nl/stopwords>, accessed on 15 November 2017

A topic is viewed as a set of semantically similar terms, namely terms that refer to the same subject or the same piece of information. In the context of a specific corpus of related documents, a topic can be defined as a set of terms that are likely to occur close to each other in a document [25]. In order to extract topics, we make use of a clustering approach based on the definition of a semantic dissimilarity between terms. For terms  $u$  and  $v$ , we first define the joint isf weight  $\text{isf}(u, v)$  as

$$\text{isf}(u, v) = \log \left( \frac{N_s}{N_s^{uv}} \right) \quad (4)$$

where  $N_s^{uv}$  is the number of sentences in which both terms  $u$  and  $v$  occur together. Then, the semantic dissimilarity  $d_{\text{sem}}(u, v)$  between the two terms is defined as

$$d_{\text{sem}}(u, v) = \frac{\text{isf}(u, v) - \min(\text{isf}(u), \text{isf}(v))}{\max(\text{isf}(u), \text{isf}(v))} \quad (5)$$

which can be viewed as a special case of the so-called google distance which was already successfully applied to learn semantic similarities between terms on webpages [26]. Using concepts from information theory,  $\text{isf}(u)$  represents the number of bits required to express the occurrence of term  $u$  in a sentence using an optimally efficient code. Then,  $\text{isf}(u, v) - \text{isf}(u)$  can be viewed as the number of bits of information in  $v$  relative to  $u$ . Assuming  $\text{isf}(v) \geq \text{isf}(u)$ ,  $d_{\text{sem}}(u, v)$  can be viewed as the improvement obtained when compressing  $v$  using a previously compressed code for  $u$  and compressing  $v$  from scratch [27]. More details can be found in [26]. In practice, two terms  $u$  and  $v$  with a low value of  $d_{\text{sem}}(u, v)$  are expected to consistently occur together in the same context, and they are thus considered to be semantically related in the context of the corpus.

Based on the semantic dissimilarity measure between terms, we define a topic as a group of terms with a high semantic density, namely a group of terms such that each term of the group is semantically related to a sufficiently high number of terms in the group. The DBSCAN algorithm is a method of density-based clustering that achieves this result by iteratively growing cohesive groups of agents, with the condition that each member of a group should contain a sufficient number of other members in an  $\epsilon$ -neighborhood around it [28]. Using the semantic dissimilarity as a distance measure, DBSCAN extracts groups of semantically related terms which are considered as topics. The advantages offered by DBSCAN over other clustering algorithms are threefold. First, DBSCAN is capable of detecting the number of clusters automatically. Second, although the semantic dissimilarity is symmetric and nonnegative, it does not satisfy the triangle inequality. This prevents the use of various clustering algorithms such as the agglomerative clustering with complete linkage [29]. However, DBSCAN does not explicitly require the triangle inequality to be satisfied. Finally, it is able to detect noisy samples in low density region, that do not belong to any other cluster.

Given a set of pairwise dissimilarity measures, a density threshold  $\epsilon$  and a minimum neighborhood size  $m$ , DBSCAN returns a number  $K$  of clusters and a set of labels  $\{c(i) \in \{-1, 1, \dots, K\} : 1 \leq i \leq N_t\}$  such that  $c(i) = -1$  if term  $i$  is considered a noisy term. While it is easy to determine a natural value for  $m$ , choosing a value for  $\epsilon$  is not straightforward. Hence, we adapt DBSCAN algorithm to build our topic model referred to as *Semantic Clustering Of Terms (SEMCOT)* algorithm. It iteratively applies DBSCAN and decreases the parameter  $\epsilon$  until the size of each cluster does not exceed a predefined



value. Algorithm 4.1 summarizes the process. Apart from  $m$ , the algorithm also takes parameters  $\epsilon_0$  (the initial value of  $\epsilon$ ),  $M$  (the maximum number of points allowed in a cluster) and  $\beta \leq 1$  (a factor close to 1 by which  $\epsilon$  is multiplied until all clusters have sizes lower than  $M$ ). Experiments on real-world data suggest empirical values of  $m = 3$ ,  $\epsilon_0 = 0.9$ ,  $M = 0.1N_t$  and  $\beta = 0.95$ . Additionally, we observe that, among the terms considered as noisy by DBSCAN, some could be highly infrequent terms with a high isf value but yet having a strong impact on the meaning of sentences. Hence, we include them as topics consisting of single terms if their isf value exceeds a threshold  $\mu$  whose value is determined by cross-validation, as explained in section 5.

**Algorithm 4.1: SEMCOT**

INPUT: Semantic Dissimilarities  $\{d_{\text{sem}}(u, v) : 1 \leq u, v \leq N_t\}$ ,  
PARAMETERS:  $\epsilon_0, M, m, \beta \leq 1, \mu$   
OUTPUT: Number  $K$  of topics, topic tags  $\{c(i) : 1 \leq i \leq N_t\}$   
 $\epsilon \leftarrow \epsilon_0$ , minTerms  $\leftarrow m$ , proceed  $\leftarrow \text{True}$   
**while** proceed:  
     $[c, K] \leftarrow \text{DBSCAN}(d_{\text{sem}}, \epsilon, \text{minTerms})$   
    **if**  $\max_{1 \leq k \leq K} (|\{i : c(i) = k\}|) < M$ : proceed  $\leftarrow \text{False}$   
    **else**:  $\epsilon \leftarrow \beta\epsilon$   
**for each**  $t$  s.t.  $c(t) = -1$  (noisy terms):  
    **if**  $\text{isf}(t) \geq \mu$ :  
         $c(t) \leftarrow K + 1$ ,  $K \leftarrow K + 1$

Once the topics are obtained based on algorithm 4.1, a *theme* is associated to each topic, namely a group of sentences covering the same topic. The sentences are first tagged with multiple topics based on a scoring function. The score of the  $l$ -th topic in the  $i$ -th sentence is given by

$$\sigma_{il} = \sum_{t:c(t)=l} \text{tfisf}(i, t) \quad (6)$$

and the sentence is tagged with topic  $l$  whenever  $\sigma_{il} \geq \delta$ , in which  $\delta$  is a parameter whose value is tuned as explained in section 5 (ensuring that each sentence is tagged with at least one topic). The scores are intentionally not normalized to avoid tagging short sentences with an excessive number of topics. The  $l$ -th theme is then defined as the set of sentences

$$T_l = \{i : \sigma_{il} \geq \delta, 1 \leq i \leq N_s\}. \quad (7)$$

While there exist other summarization models based on the detection of clusters or groups of similar sentence, the novelty of our theme model is twofold. First, each theme is easily interpretable as the set of sentences associated to a specific topic. As such, our themes can be considered as groups of semantically related sentences. Second, it is clear that the themes discovered by our approach do overlap since a single sentence may be tagged with multiple topics. To the best of our knowledge, none of the previous cluster-based summarizers involved overlapping groups of sentences. Our model is thus more realistic since it better captures the multiplicity of the information covered by each sentence.

### 4.3 Sentence hypergraph construction

A hypergraph is a generalization of a graph in which the hyperedges may contain any number of nodes, as expressed in definition 1 [4]. Our hypergraph model moreover includes both hyperedge and node weights.

**Definition 1** (Hypergraph). *A node- and hyperedge-weighted hypergraph is defined as a quadruplet  $H = (V, E, \phi, w)$  in which  $V$  is a set of nodes,  $E \subseteq 2^V$  is a set of hyperedges,  $\phi \in \mathbb{R}_+^{|V|}$  is a vector of positive node weights and  $w \in \mathbb{R}_+^{|E|}$  is a vector of positive hyperedge weights.*

For convenience, we will refer to a hypergraph by its weight vectors  $\phi$  and  $w$ , its hyperedges represented by a set  $E \subseteq 2^V$  and its incidence lists  $\text{inc}(i) = \{e \in E : i \in e\}$  for each  $i \in V$ .

As mentioned in section 1, our system relies on the definition of a theme-based hypergraph which models groups of semantically related sentences as hyperedges. Hence, compared to traditional graph-based summarizers, the hypergraph is able to capture more complex group relationships between sentences instead of being restricted to pairwise relationships.

In our sentence-based hypergraph, the sentences are the nodes and each theme defines a hyperedge connecting the associated sentences. The weight  $\phi_i$  of node  $i$  is the length of the  $i$ -th sentence, namely:

$$\begin{aligned} V &= \{1, \dots, N_s\} \text{ and } \phi_i = L_i, \quad 1 \leq i \leq N_s \\ E &= \{e_1, \dots, e_K\} \subseteq 2^V \\ e_l &= T_l \text{ i.e. } e_l \in \text{inc}(i) \leftrightarrow i \in T_l \end{aligned} \tag{8}$$

Finally, the weights of the hyperedges are computed based on the centrality of the associated theme and its similarity with the query:

$$w_l = (1 - \lambda)\text{sim}(T_l, D) + \lambda\text{sim}(T_l, q) \tag{9}$$

where  $\lambda \in [0, 1]$  is a parameter and  $D$  represents the entire corpus.  $\text{sim}(T_l, D)$  denotes the similarity of the set of sentences in theme  $T_l$  with the entire corpus (using the tfidf-based similarity of equation 3) which measures the centrality of the theme in the corpus.  $\text{sim}(T_l, q)$  refers to the similarity of the theme with the user-defined query  $q$ .

### 4.4 Detection of hypergraph transversals for text summarization

The sentences to be included in the query-oriented summary should contain the essential information in the corpus, they should be relevant to the query and, whenever required, they should either not exceed a *target length* or jointly achieve a *target coverage* (as mentioned in section 3). Existing systems of graph-based summarization generally solve the problem by ranking sentences in terms of their *individual* relevance [1, 3, 4]. Then, they extract a set of sentences with a maximal total relevance and pairwise similarities not exceeding a predefined threshold. However, we argue that the *joint* relevance of a group of sentences is not reflected by the individual relevance of each sentence. And

limiting the redundancy of selected sentences as done in [4] does not guarantee that the sentences jointly cover the relevant themes of the corpus.

Considering each topic as a distinct piece of information in the corpus, an alternative approach is to select the smallest subset of sentences covering each of the topics. The latter condition can be reformulated as ensuring that each theme has at least one of its sentences appearing in the summary. Using our sentence hypergraph representation, this corresponds to the detection of a minimal hypergraph transversal as defined below [6].

**Definition 2.** Given an unweighted hypergraph  $H = (V, E)$ , a minimal hypergraph transversal is a subset  $S^* \subseteq V$  of nodes satisfying

$$\begin{aligned} S^* &= \underset{S \subseteq V}{\operatorname{argmin}} |S| \\ \text{s.t. } &\bigcup_{i \in S} \operatorname{inc}(i) = E \end{aligned} \quad (10)$$

where  $\operatorname{inc}(i) = \{e : i \in e\}$  denotes the set of hyperedges incident to node  $i$ .

Figure 2 shows an example of hypergraph and a minimal hypergraph transversal of it (star-shaped nodes). In this case, since the nodes and the hyperedges are unweighted, the minimal transversal is not unique.

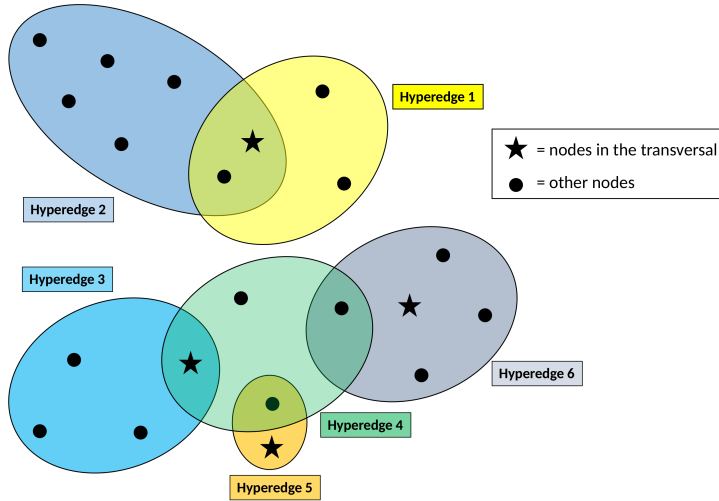


Figure 2: Example of hypergraph and minimal hypergraph transversal.

The problem of finding a minimal transversal in a hypergraph is NP-hard [30]. However, greedy algorithms or LP relaxations provide good approximate solutions in practice [22]. As intended, the definition of transversal includes the notion of *joint* coverage of the themes by the sentences. However, it neglects node and hyperedge weights and it is unable to identify query-relevant themes. Since both the sentence lengths and the relevance of themes should be taken into account in the summary generation, we introduce two extensions of transversal, namely the *minimal soft hypergraph transversal* and the

*maximal budgeted hypergraph transversal.* A minimal soft transversal of a hypergraph is obtained by minimizing the total weights of selected nodes while ensuring that the total weight of covered hyperedges exceeds a given threshold.

**Definition 3** (minimal soft hypergraph transversal). *Given a node and hyperedge weighted hypergraph  $H = (V, E, \phi, w)$  and a parameter  $\gamma \in [0, 1]$ , a minimal soft hypergraph transversal is a subset  $S^* \subseteq V$  of nodes satisfying*

$$\begin{aligned} S^* &= \underset{S \subseteq V}{\operatorname{argmin}} \sum_{i \in S} \phi_i \\ \text{s.t. } &\sum_{e \in \operatorname{inc}(S)} w_e \geq \gamma W \end{aligned} \quad (11)$$

in which  $\operatorname{inc}(S) = \bigcup_{i \in S} \operatorname{inc}(i)$  and  $W = \sum_e w_e$ .

The extraction of a minimal soft hypergraph transversal of the sentence hypergraph produces a summary of minimal length achieving a *target coverage* expressed by parameter  $\gamma \in [0, 1]$ . As mentioned in section 3, applications of text summarization may also involve a hard constraint on the total summary length  $L$ . For that purpose, we introduce the notion of *maximal budgeted hypergraph transversal* which maximizes the volume of covered hyperedges while not exceeding the target length.

**Definition 4** (maximal budgeted hypergraph transversal). *Given a node and hyperedge weighted hypergraph  $H = (V, E, \phi, w)$  and a parameter  $L > 0$ , a maximal budgeted hypergraph transversal is a subset  $S^* \subseteq V$  of nodes satisfying*

$$\begin{aligned} S^* &= \underset{S \subseteq V}{\operatorname{argmax}} \sum_{e \in \operatorname{inc}(S)} w_e \\ \text{s.t. } &\sum_{i \in S} \phi_i \leq L. \end{aligned} \quad (12)$$

We refer to the function  $\sum_{e \in \operatorname{inc}(S)} w_e$  as the *hyperedge coverage* of set  $S$ . We observe that both weighted transversals defined above include the notion of *joint* coverage of the hyperedges by the selected nodes. As a result and from the definition of hyperedge weights (equation 9), the resulting summary covers themes that are both central in the corpus and relevant to the query. This approach also implies that the resulting summary does not contain redundant sentences covering the exact same themes. As a result selected sentences are expected to cover different themes and to be semantically diverse. Both the problems of finding a minimal soft transversal or finding a maximal budgeted transversal are NP-hard as proved by theorem 1.

**Theorem 1** (NP-hardness). *The problems of finding a minimal soft hypergraph transversal or a maximal budgeted hypergraph transversal in a weighted hypergraph are NP-hard.*

*Proof.* Regarding the minimal soft hypergraph transversal problem, with parameter  $\gamma = 1$  and unit node weights, the problem is equivalent to the classical set cover problem (definition 2) which is NP-complete [30]. The maximal budgeted hypergraph transversal problem can be shown to be equivalent to the maximum coverage problem with knapsack constraint which was shown to be NP-complete in [30].  $\square$

Since both problems are NP-hard, we formulate polynomial time algorithms to find approximate solutions to them and we provide the associated approximation factors. The algorithms build on the submodularity and the non-decreasing properties of the hyperedge coverage function, which are defined below.

**Definition 5** (Submodular and non-decreasing set functions). *Given a finite set  $A$ , a function  $f : 2^A \rightarrow \mathbb{R}$  is monotonically non-decreasing if  $\forall S \subset A$  and  $\forall u \in A \setminus S$ ,*

$$f(S \cup \{u\}) \geq f(S) \quad (13)$$

*and it is submodular if  $\forall S, T$  with  $S \subseteq T \subset A$ , and  $\forall u \in A \setminus T$ ,*

$$f(T \cup \{u\}) - f(T) \leq f(S \cup \{u\}) - f(S). \quad (14)$$

Based on definition 5, we prove in theorem 2 that the hyperedge coverage function is submodular and monotonically non-decreasing, which provides the basis of our algorithms.

**Theorem 2.** *Given a hypergraph  $H = (V, E, \phi, w)$ , the hyperedge coverage function  $f : 2^V \rightarrow \mathbb{R}$  defined by*

$$f(S) = \sum_{e \in \text{inc}(S)} w_e \quad (15)$$

*is submodular and monotonically non-decreasing.*

*Proof.* The hyperedge coverage function  $f$  is clearly monotonically non-decreasing and it is submodular since  $\forall S \subseteq T \subset V$ , and  $s \in V \setminus T$ ,

$$\begin{aligned} & (f(S \cup \{s\}) - f(S)) - (f(T \cup \{s\}) - f(T)) \\ &= \left[ \sum_{e \in \text{inc}(S \cup \{s\})} w_e - \sum_{e \in \text{inc}(S)} w_e \right] - \left[ \sum_{e \in \text{inc}(T \cup \{s\})} w_e - \sum_{e \in \text{inc}(T)} w_e \right] \\ &= \left[ \sum_{e \in \text{inc}(\{s\}) \setminus \text{inc}(S)} w_e \right] - \left[ \sum_{e \in \text{inc}(\{s\}) \setminus \text{inc}(T)} w_e \right] \\ &= \sum_{e \in (\text{inc}(T) \cap \text{inc}(\{s\})) \setminus \text{inc}(S)} w_e \geq 0 \end{aligned} \quad (16)$$

where  $\text{inc}(R) = \{e : e \cap R \neq \emptyset\}$  for  $R \subseteq V$ . The last equality follows from  $\text{inc}(S) \subseteq \text{inc}(T)$  and  $\text{inc}(\{s\}) \setminus \text{inc}(T) \subseteq \text{inc}(\{s\}) \setminus \text{inc}(S)$ .  $\square$

Various classes of NP-hard problems involving a submodular and non-decreasing function can be solved approximately by polynomial time algorithms with provable approximation factors. Algorithms 4.2 and 4.3 are our core methods for the detection of approximations of maximal budgeted hypergraph transversals and minimal soft hypergraph transversals, respectively. In each case, a transversal is found and the summary is formed by extracting and aggregating the associated sentences. Algorithm 4.2 is based on an adaptation of an algorithm presented in [31] for the maximization of submodular functions under a Knapsack constraint. It is our primary transversal-based summarization model, and we refer to it as the method of *Transversal Summarization with Target Length* (*TL-TransSum* algorithm). Algorithm 4.3 is an application of the algorithm presented

in [21] for solving the submodular set covering problem. We refer to it as *Transversal Summarization with Target Coverage* (TC-*TranSum* algorithm). Both algorithms produce transversals by iteratively appending the node inducing the largest increase in the total weight of the covered hyperedges relative to the node weight. While long sentences are expected to cover more themes and induce a larger increase in the total weight of covered hyperedges, the division by the node weights (i.e. the sentence lengths) balances this tendency and allows the inclusion of short sentences as well. In contrast, the methods of sentence selection based on a maximal relevance and a minimal redundancy such as, for instance, the maximal marginal relevance approach of [32], tend to favor the selection of long sentences only. The main difference between algorithms 4.2 and 4.3 is the stopping criterion: in algorithm 4.3, the approximate minimal soft transversal is obtained whenever the targeted hyperedge coverage is reached while algorithm 4.2 appends a given sentence to the approximate maximal budgeted transversal only if its addition does not make the summary length exceed the target length  $L$ .

**Algorithm 4.2:** *Transversal Summarization with Target Length* (TL-*TranSum*)

INPUT: Sentence Hypergraph  $H = (V, E, \phi, w)$ , target length  $L$ .

OUTPUT: Set  $S$  of sentences to be included in the summary.

**for each**  $i \in \{1, \dots, N_s\}$ :  $r_i \leftarrow \frac{1}{\phi_i} \sum_{e \in \text{inc}(i)} w_e$

$R \leftarrow \emptyset, Q \leftarrow V, f \leftarrow 0$

**while**  $Q \neq \emptyset$ :

$s^* \leftarrow \underset{i \in Q}{\text{argmax}} r_i, Q \leftarrow Q \setminus \{s^*\}$

**if**  $\phi_{s^*} + f \leq L$ :

$R \leftarrow R \cup \{s^*\}, f \leftarrow f + l^*$

**for each**  $i \in \{1, \dots, N_s\}$ :  $r_i \leftarrow r_i - \frac{\sum_{e \in \text{inc}(s^*) \cap \text{inc}(i)} w_e}{\phi_i}$

Let  $G \leftarrow \{\{i\} : i \in V, \phi_i \leq L\}$

$S \leftarrow \underset{S \in \{Q\} \cup G}{\text{argmax}} \sum_{e \in \text{inc}(S)} w_e$

return  $S$

**Algorithm 4.3:** *Transversal Summarization with Target Coverage (TC-TransSum)*

INPUT: Sentence Hypergraph  $H = (V, E, \phi, w)$ , parameter  $\gamma \in [0, 1]$ .

OUTPUT: Set  $S$  of sentences to be included in the summary.

**for each**  $i \in \{1, \dots, N_s\}$ :  $r_i \leftarrow \frac{1}{\phi_i} \sum_{e \in \text{inc}(i)} w_e$

$S \leftarrow \emptyset, Q \leftarrow V, \tilde{W} \leftarrow 0, W \leftarrow \sum_e w_e$

**while**  $Q \neq \emptyset$  and  $\tilde{W} < \gamma W$ :

$s^* \leftarrow \underset{i \in Q}{\operatorname{argmax}} r_i$

$S \leftarrow S \cup \{s^*\}, \tilde{W} \leftarrow \tilde{W} + \phi_{s^*} r_{s^*}$

**for each**  $i \in \{1, \dots, N_s\}$ :  $r_i \leftarrow r_i - \frac{\sum_{e \in \text{inc}(s^*) \cap \text{inc}(i)} w_e}{\phi_i}$

return  $S$

We next provide theoretical guarantees that support the formulation of algorithms 4.2 and 4.3 as approximation algorithms for our hypergraph transversals. Theorem 3 provides a constant approximation factor for the output of algorithm 4.2 for the detection of minimal soft hypergraph transversals. It builds on the submodularity and the non-decreasing property of the hyperedge coverage function.

**Theorem 3.** *Let  $S^L$  be the summary produced by our TL-TransSum algorithm 4.2, and  $S^*$  be a maximal budgeted transversal associated to the sentence hypergraph, then*

$$\sum_{e \in \text{inc}(S^L)} w_e \geq \frac{1}{2} \left(1 - \frac{1}{e}\right) \sum_{e \in \text{inc}(S^*)} w_e. \quad (17)$$

*Proof.* Since the hyperedge coverage function is submodular and monotonically non-decreasing, the extraction of a maximal budgeted transversal is a problem of maximization of a submodular and monotonically non-decreasing function under a Knapsack constraint, namely

$$\max_{S \subseteq V} f(S) \text{ s.t. } \sum_{i \in S} \phi_i \leq L \quad (18)$$

where  $f(S) = \sum_{e \in \text{inc}(S)} w_e$ . Hence, by theorem 2 in [31], the algorithm forming a transversal

$S^F$  by iteratively growing a set  $S_t$  of sentences according to

$$S_{t+1} = S_t \cup \left\{ \underset{s \in V \setminus S_t}{\operatorname{argmax}} \left\{ \frac{f(S \cup \{s\}) - f(S)}{\phi_s}, \phi_s + \sum_{i \in S_t} \phi_i \leq L \right\} \right\} \quad (19)$$

produces a final summary  $S^F$  satisfying

$$f(S^F) \geq f(S^*) \frac{1}{2} \left(1 - \frac{1}{e}\right). \quad (20)$$

As algorithm 4.2 implements the iterations expressed by equation 19, it achieves a constant approximation factor of  $\frac{1}{2} \left(1 - \frac{1}{e}\right)$ .  $\square$

Similarly, theorem 4 provides a data-dependent approximation factor for the output of algorithm 4.3 for the detection of maximal budgeted hypergraph transversals. It also builds on the submodularity and the non-decreasing property of the hyperedge coverage function.

**Theorem 4.** *Let  $S^P$  be the summary produced by our TC-TranSum algorithm 4.3 and let  $S^*$  be a minimal soft hypergraph transversal, then*

$$\sum_{i \in S^P} \phi_i \leq \sum_{i \in S^*} \phi_i \left( 1 + \log \left( \frac{\gamma W}{\gamma W - \sum_{e \in \text{inc}(S^{T-1})} w_e} \right) \right) \quad (21)$$

where  $S_1, \dots, S_T$  represent the consecutive sets of sentences produced by algorithm 4.3.

*Proof.* Consider the function  $g(S) = \min(\gamma W, \sum_{e \in \text{inc}(S)} w_e)$ . Then the problem of finding a minimal soft hypergraph transversal can be reformulated as

$$S^* = \underset{S \subseteq V}{\text{argmin}} \sum_{s \in S} \phi_s \text{ s.t. } g(S) \geq g(V) \quad (22)$$

As  $g$  is submodular and monotonically non-decreasing, theorem 1 in [21] shows that the summary  $S^G$  produced by iteratively growing a set  $S_t$  of sentences such that

$$S_{t+1} = S_t \cup \left\{ \underset{s \in V \setminus S_t}{\text{argmax}} \left\{ \frac{f(S \cup \{s\}) - f(S)}{\phi_s} \right\} \right\} \quad (23)$$

produces a summary  $S^G$  satisfying

$$\sum_{i \in S^G} \phi_i \leq \sum_{i \in S^*} \phi_i \left( 1 + \log \left( \frac{g(V)}{g(V) - g(S^{T-1})} \right) \right). \quad (24)$$

which can be rewritten as

$$\sum_{i \in S^G} \phi_i \leq \sum_{i \in S^*} \phi_i \left( 1 + \log \left( \frac{\gamma W}{\gamma W - \sum_{e \in \text{inc}(S^{T-1})} w_e} \right) \right). \quad (25)$$

As algorithm 4.3 implements the iterations expressed by equation 23, the summary  $S^S$  produced by our algorithm 4.3 satisfies the same inequality.  $\square$

In practice, the result of theorem 4 suggests that the quality of the output depends on the relative increase in the hyperedge coverage induced by the last sentence to be appended to the summary. In particular, if each sentence that is appended to the summary in the iterations of algorithm 4.3 covers a sufficient number of new themes that are not covered already by the summary, the approximation factor is low.



## 4.5 Complexity analysis

We analyse the worst case time complexity of each step of our method. The time complexity of DBSCAN algorithm [28] is  $O(N_t \log(N_t))$ . Hence, the theme detection algorithm 4.1 takes  $O(N_c N_t \log(N_t))$  steps, where  $N_c$  is the number of iterations of algorithm 4.1 which is generally low compared to the number of terms. The time complexity for the hypergraph construction is  $O(K(N_s + N_t))$  where  $K$  is the number of topics, or  $O(N_t^2)$  if  $N_t \geq N_s$ . The time complexity of the sentence selection algorithms 4.2 and 4.3 are bounded by  $O(N_s K C^{\max} L^{\max})$  where  $C^{\max}$  is the number of sentences in the largest theme and  $L^{\max}$  is the length of the longest sentences. Assuming  $N_t$  is larger than  $N_s$ , the overall time complexity of the method is of  $O(N_t^2)$  steps in the worst case. Hence the method is essentially equivalent to early graph-based models for text summarization in terms of computational burden, such as the LexRank-based systems [1, 3] or greedy approaches based on global optimization [16–18]. However, it is computationally more efficient than traditional hypergraph-based summarizers such as the one in [5] which involves a Markov Chain Monte Carlo inference for its topic model or the one in [4] which is based on an iterative computation of scores involving costly matrix multiplications at each step.

## 5 Experiments and evaluation

We present experimental results obtained with a Python implementation of algorithms 4.2 and 4.3 on a standard computer with a  $2.5GHz$  processor and a 8GB memory.

### 5.1 Dataset and metrics for evaluation

We test our algorithms on DUC2005 [33], DUC2006 [34] and DUC2007 [35] datasets which were produced by the Document Understanding Conference (DUC) and are widely used as benchmark datasets for the evaluation of query-oriented summarizers. The datasets consist respectively of 50, 50 and 45 corpora, each consisting of 25 documents of approximately 1000 words, on average. A query is associated to each corpus. For evaluation purposes, each corpus is associated with a set of query-relevant summaries written by humans called *reference summaries*. In each of our experiments, a candidate summary is produced for each corpus by one of our algorithms and it is compared with the reference summaries using the metrics described below. Moreover, in experiments involving algorithm 4.2, the target summary length is set to 250 words as required in DUC evaluations.

In order to evaluate the similarity of a candidate summary with a set of reference summaries, we make use of the ROUGE toolkit of [36], and more specifically of ROUGE-2 and ROUGE-SU4 metrics, which were adopted by DUC for summary evaluation. ROUGE-2 measures the number of bigrams found both in the candidate summary and the set of reference summaries. ROUGE-SU4 extends this approach by counting the number of unigrams and the number of 4-skip-bigrams appearing in the candidate and the reference summaries, where a 4-skip-bigram is a pair of words that are separated by no more than 4 words in a text. We refer to ROUGE toolkit [36] for more details about the evaluation metrics. ROUGE-2 and ROUGE-SU4 metrics are computed following the same setting as

in DUC evaluations, namely with word stemming and jackknife resampling but without stopword removal.

## 5.2 Parameter tuning

Besides the parameters of SEMCOT algorithm for which empirical values were given in section 4.2, there are three parameters of our system that need to be tuned: parameters  $\mu$  (threshold on isf value to include a noisy term as a single topic in SEMCOT),  $\delta$  (threshold on the topic score for tagging a sentence with a given topic) and  $\lambda$  (balance between the query relevance and the centrality in hyperedge weights). The values of all three parameters are determined by an alternating maximization strategy of ROUGE-SU4 score in which the values of two parameters are fixed and the value of the third parameter is tuned to maximize the ROUGE-SU4 score produced by algorithm 4.2 with a target summary length of 250 words, in an iterative fashion. The ROUGE-SU4 scores are evaluated by cross-validation using a leave-one-out process on a validation dataset consisting of 70% of DUC2007 dataset, which yields  $\mu = 1.98$ ,  $\delta = 0.85$  and  $\lambda = 0.4$ .

Additionally, we display the evolution of ROUGE-SU4 and ROUGE-2 scores as a function of  $\delta$  and  $\lambda$ . For parameter  $\delta$ , we observe in graphs 3(a) and 3(b) that the quality of the summary is low for  $\delta$  close to 0 since it encourages our theme detection algorithm to tag the sentences with irrelevant topics with low associated tfidf values. In contrast, when  $\delta$  exceeds 0.9, some relevant topics are overlooked and the quality of the summaries drops severely. Regarding parameter  $\lambda$ , we observe in graphs 4(a) and 4(b) that  $\lambda = 0.4$  yields the highest score since it combines both the relevance of themes to the query and their centrality within the corpus for the computation of hyperedge weights. In contrast, with  $\lambda = 1$ , the algorithm focuses on the lexical similarity of themes with the query but it neglects the prominence of each theme.

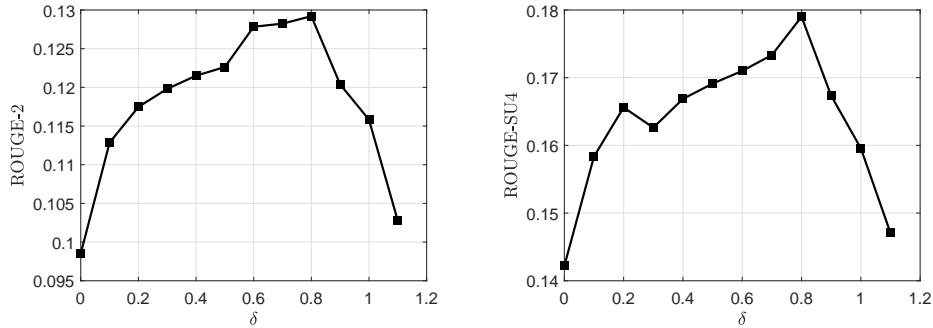


Figure 3: ROUGE-2 and ROUGE-SU4 as a function of  $\delta$  for  $\lambda = 0.4$  and  $\mu = 1.98$ .

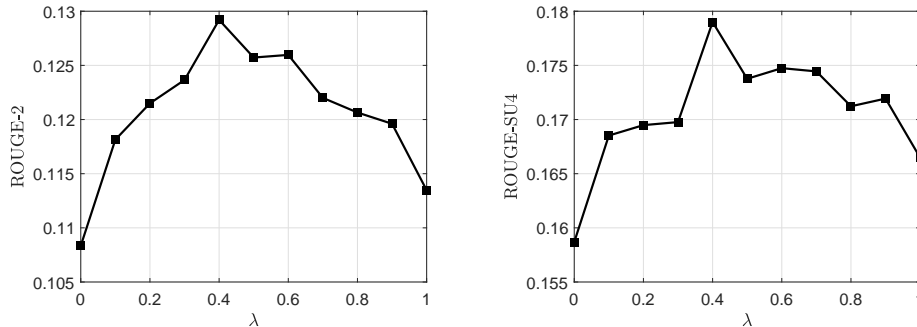


Figure 4: ROUGE-2 and ROUGE-SU4 as a function of  $\lambda$  for  $\delta = 0.85$  and  $\mu = 1.98$ .

### 5.3 Testing the TC-TranSum algorithm

In order to test our soft transversal-based summarizer, we display the evolution of the summary length and the ROUGE-SU4 score as a function of parameter  $\gamma$  of algorithm 4.3. In figure 5(b), we observe that the summary length grows linearly with the value of parameter  $\gamma$  which confirms that our system does not favor longer sentences for low values of  $\gamma$ . The ROUGE-SU4 curve of figure 5(a) has a concave shape, with a low score when  $\gamma$  is close to 0 (due to a poor recall) or when  $\gamma$  is close to 1 (due to a poor precision). The overall concave shape of the ROUGE-SU4 curve also demonstrates the efficiency of our TC-TranSum algorithm: based on our hyperedge weighting scheme and our hyperedge coverage function, highly relevant sentences inducing a significant increase in the ROUGE-SU4 score are identified and included first in the summary.

In the subsequent experiments, we focus on TL-TranSum algorithm 4.2 which includes a target summary length and can thus be compared with other summarization systems which generally include a length constraint.

### 5.4 Testing the hypergraph structure

To justify our theme-based hypergraph definition, we test other hypergraph models. We only change the hyperedge model which determines the kind of relationship between sentences that is captured by the hypergraph. The sentence selection is performed by applying algorithm 4.2 to the resulting hypergraph. We test three alternative hyperedge models. First a model based on *agglomerative* clustering instead of SEMCOT: the same definition of semantic dissimilarity (equation 5) is used, then topics are detected as clusters of terms obtained by agglomerative clustering with single linkage with the semantic dissimilarity as a distance measure. The themes are detected and the hypergraph is constructed in the same way as in our model. Second, *Overlap* model defines hyperedges as overlapping clusters of sentences obtained by applying an algorithm of overlapping cluster detection [37] and using the cosine distance between tfidf representations of sentences as a distance metric. Finally, we test a hypergraph model already proposed in HyperSum system by [4] which combines pairwise hyperedges joining any two sentences having terms

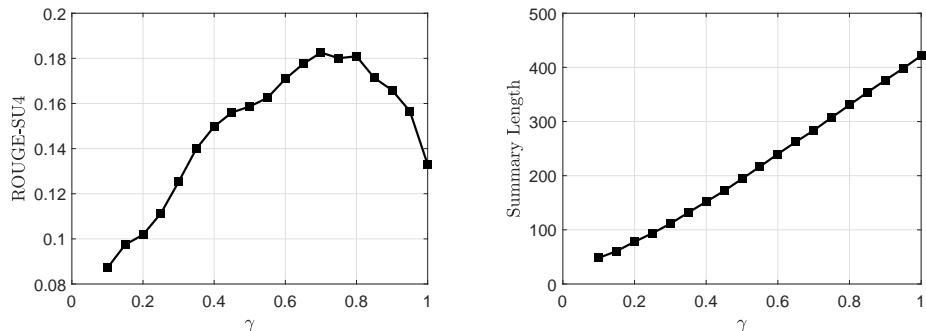


Figure 5: Evolution of the ROUGE-SU4 score (left) and the summary length (right) as a function of the coverage parameter  $\gamma$  of TC-TranSum algorithm 4.3.

in common and hyperedges formed by non-overlapping clusters of sentences obtained by DBSCAN algorithm. Table 1 displays the ROUGE-2 and ROUGE-SU4 scores and the corresponding 95% confidence intervals for each model. We observe that our model outperforms both *HyperSum* and *Overlap* models by at least 4% and 15% of ROUGE-SU4 score, respectively, which confirms that a two-step process extracting consistent topics first and then defining theme-based hyperedges from topic tags outperforms approaches based on sentence clustering, even when these clusters do overlap. Our model also outperforms the *Agglomerative* model by 10% of ROUGE-SU4 score, due to its ability to identify noisy terms and to detect the number of topics automatically.

System	ROUGE-2	ROUGE-SU4
<b>TL-TranSum</b>	<b>0.12997(0.12548 – 0.13446)</b>	<b>0.17995(0.17612 – 0.18377)</b>
Agglomerative	0.12334(0.11673 – 0.12994)	0.16292(0.15302 – 0.17282)
Overlap	0.11831(0.11334 – 0.12328)	0.15640(0.14762 – 0.16518)
HyperSum	0.12317(0.11743 – 0.12892)	0.17231(0.16561 – 0.17900)

Table 1: ROUGE-2 and ROUGE-SU4 scores for our TL-TranSum system compared to three other hypergraph models.

## 5.5 Comparison with related systems

We compare the performance of our TL-TranSum algorithm 4.2 with that of five related summarization systems. *Topic-sensitive LexRank* of [3] (TS-LexRank) and *HITS* algorithms of [2] are early graph-based summarizers. TS-LexRank builds a sentence graph based on term co-occurrences in sentences, and it applies a query-biased PageRank algorithm for sentence scoring. HITS method additionally extracts clusters of sentences and it applies the hubs and authorities algorithm for sentence scoring, with the sentences as authorities and the clusters as hubs. As suggested in [4], in order to extract query relevant

sentences, only the top 5% of sentences that are most relevant to the query are considered. *HyperSum* extends early graph-based summarizers by defining a cluster-based hypergraph with the sentences as nodes and hyperedges as sentence clusters, as described in section 5.4. The sentences are then scored using an iterative label propagation algorithm over the hypergraph, starting with the lexical similarity of each sentence with the query as initial labels. In all three methods, the sentences with highest scores and pairwise lexical similarity not exceeding a threshold are included in the summary. Finally, we test two methods that also build on the theory of submodular functions. First, the *MaxCover* approach [10] seeks a summary by maximizing the number of distinct relevant terms appearing in the summary while not exceeding the target summary length (using equation 9 to compute the term relevance scores). While the objective function of the method is similar to that of the problem of finding a maximal budgeted hypergraph transversal (equation 12) of [17], they overlook the semantic similarities between terms which are captured by our SEMCOT algorithm and our hypergraph model. Similarly, the *Maximal Relevance Minimal Redundancy* (MRMR) first computes relevance scores of sentences as in equation 9, then it seeks a summary with a maximal total relevance score and a minimal redundancy while not exceeding the target summary length. The problem is solved by an iterative algorithm building on the submodularity and non-decreasing property of the objective function.

Table 2 displays the ROUGE-2 and ROUGE-SU4 scores with the corresponding 95% confidence intervals for all six systems, including our TL-TranSum method. We observe that our system outperforms other graph and hypergraph-based summarizers involving the computation of individual sentence scores: LexRank by 6%, HITS by 13% and HyperSum by 6% of ROUGE-SU4 score; which confirms both the relevance of our theme-based hypergraph model and the capacity of our transversal-based summarizer to identify jointly relevant sentences as opposed to methods based on the computation of individual sentence scores. Moreover, our TL-TranSum method also outperforms other approaches such as MaxCover (5%) and MRMR (7%). These methods are also based on a submodular and non-decreasing function expressing the information coverage of the summary, but they are limited to lexical similarities between sentences and fail to detect topics and themes to measure the information coverage of the summary.

System	ROUGE-2	ROUGE-SU4
<b>TL-TranSum</b>	<b>0.12997(0.12548 – 0.13446)</b>	<b>0.17995(0.17612 – 0.18377)</b>
TS-LexRank	0.11037(0.10263 – 0.11811)	0.16939(0.16233 – 0.17645)
HITS	0.10972(0.10155 – 0.11789)	0.15927(0.15251 – 0.16603)
HyperSum	0.11994(0.11298 – 0.12690)	0.16993(0.16189 – 0.17797)
MaxCover	0.11985(0.11028 – 0.12943)	0.17072(0.16155 – 0.17988)
MRMR	0.11840(0.10999 – 0.12681)	0.16857(0.16046 – 0.17668)

Table 2: Comparison with related graph- and hypergraph-based summarization systems.

## 5.6 Comparison with DUC systems

As a final experiment, we compare our TL-TranSum approach to other summarizers presented at DUC contests. Table 3 displays the ROUGE-2 and ROUGE-SU4 scores for the worst summary produced by a human, for the top four systems submitted for the contests, for the baseline proposed by NIST (a summary consisting of the leading sentences of randomly selected documents) and the average score of all methods submitted, respectively for DUC2005, DUC2006 and DUC2007 contests. Regarding DUC2007, our method outperforms the best system by 2% and the average ROUGE-SU4 score by 21%. It also performs significantly better than the baseline of NIST. However, it is outperformed by the human summarizer since our systems produces extracts, while humans naturally reformulate the original sentences to compress their content and produce more informative summaries. Tests on DUC2006 dataset lead to similar conclusions, with our TL-TranSum algorithm outperforming the best other system and the average ROUGE-SU4 score by 2% and 22%, respectively. On DUC2005 dataset however, our TL-TranSum method is outperformed by the beset system which is due to the use of advanced NLP techniques (such as sentence trimming [38]) which tend to increase the ROUGE-SU4 score. Nevertheless, the ROUGE-SU4 score produced by our TL-TranSum algorithm is still 15% higher than the average score for DUC2005 contest.

	DUC2005		DUC2006		DUC2007	
Method	ROUGE-2	ROUGE-SU4	ROUGE-2	ROUGE-SU4	ROUGE-2	ROUGE-SU4
Hum	0.0897	0.151	0.13260	0.18385	0.17528	0.21892
<b>TL-TranSum</b>	<b>0.077392</b>	0.12869	<b>0.10779</b>	<b>0.15854</b>	<b>0.12997</b>	<b>0.17995</b>
1st	0.07251	<b>0.13163</b>	0.09558	0.15529	0.12448	0.17711
2nd	0.07174	0.12972	0.09097	0.14733	0.12028	0.17074
3rd	0.06984	0.12525	0.08987	0.14755	0.11887	0.16999
4th	0.06963	0.12795	0.08954	0.14607	0.11793	0.17593
Syst. Av.	0.05842	0.11205	0.07463	0.13021	0.09597	0.14884
Basel.	0.04026	0.08716	0.04947	0.09788	0.06039	0.10507

Table 3: Comparison with DUC2005, DUC2006 and DUC2007 systems

## 6 Conclusion

In this paper, a new hypergraph-based summarization model was proposed, in which the nodes are the sentences of the corpus and the hyperedges are themes grouping sentences covering the same topics. Going beyond existing methods based on simple graphs and pairwise lexical similarities, our hypergraph model captures groups of semantically related sentences. Moreover, two new method of sentence selection based on the detection of hypergraph transversals were proposed: one to generate summaries of minimal length and achieving a target coverage, and the other to generate a summary achieving a maximal coverage of relevant themes while not exceeding a target length. The approach generates informative summaries by extracting a subset of sentences jointly covering the relevant themes of the corpus. Experiments on a real-world dataset demonstrate the effectiveness of the approach. The hypergraph model itself is shown to produce more accurate summaries than other models based on term or sentence clustering. The overall

system also outperforms related graph- or hypergraph-based approaches by at least 10% of ROUGE-SU4 score.

As a future research direction, we may analyse the performance of other algorithms for the detection of hypergraph transversals, such as methods based on LP relaxations. We may also further extend our topic model to take the polysemy of terms into account: since each term may carry multiple meanings, a given term could refer to different topics depending on its context. Finally, we intend to adapt our model for solving related problems, such as community question answering.

## References

## References

- [1] G. Erkan and D. R. Radev, “Lexrank: Graph-based lexical centrality as salience in text summarization,” *Journal of Artificial Intelligence Research*, vol. 22, pp. 457–479, 2004.
- [2] X. Wan and J. Yang, “Multi-document summarization using cluster-based link analysis,” in *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, pp. 299–306, ACM, 2008.
- [3] J. Otterbacher, G. Erkan, and D. R. Radev, “Using random walks for question-focused sentence retrieval,” in *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, pp. 915–922, ACL, 2005.
- [4] W. Wang, S. Li, J. Li, W. Li, and F. Wei, “Exploring hypergraph-based semi-supervised ranking for query-oriented summarization,” *Information Sciences*, vol. 237, pp. 271–286, 2013.
- [5] S. Xiong and D. Ji, “Query-focused multi-document summarization using hypergraph-based ranking,” *Information Processing & Management*, vol. 52, no. 4, pp. 670–681, 2016.
- [6] D. Gunopulos, H. Mannila, R. Khardon, and H. Toivonen, “Data mining, hypergraph transversals, and machine learning,” in *Proceedings of the sixteenth ACM SIGACT-SIGMOD-SIGART symposium on Principles of database systems*, pp. 209–216, ACM, 1997.
- [7] S. Klamt, U. U. Haus, and F. Theis, “Hypergraphs and cellular networks,” *PLoS computational biology*, vol. 5, no. 5, p. e1000385, 2009.
- [8] K. Hong, J. M. Conroy, B. Favre, A. Kulesza, H. Lin, and A. Nenkova, “A repository of state of the art and competitive baseline summaries for generic news summarization,” in *LREC*, pp. 1608–1616, 2014.
- [9] A. Kanapala, S. Pal, and R. Pamula, “Text summarization from legal documents: a survey,” *Artificial Intelligence Review*, pp. 1–32, 2017.

- [10] H. Takamura and M. Okumura, “Text summarization model based on maximum coverage problem and its variant,” in *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics*, pp. 781–789, Association for Computational Linguistics, 2009.
- [11] A. Nenkova, K. McKeown, *et al.*, “Automatic summarization,” *Foundations and Trends® in Information Retrieval*, vol. 5, no. 2–3, pp. 103–233, 2011.
- [12] A. Nenkova and K. McKeown, “A survey of text summarization techniques,” in *Mining text data* (C. C. Aggarwal and C. Zhai, eds.), ch. 3, pp. 43–76, Springer Science & Business Media, 2012.
- [13] M. A. Fattah, “A hybrid machine learning model for multi-document summarization,” *Applied intelligence*, vol. 40, no. 4, pp. 592–600, 2014.
- [14] H. Zha, “Generic summarization and keyphrase extraction using mutual reinforcement principle and sentence clustering,” in *Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*, pp. 113–120, ACM, 2002.
- [15] Z. Zhang, S. S. Ge, and H. He, “Mutual-reinforcement document summarization using embedded graph based sentence clustering for storytelling,” *Information Processing & Management*, vol. 48, no. 4, pp. 767–778, 2012.
- [16] H. Lin and J. Bilmes, “Multi-document summarization via budgeted maximization of submodular functions,” in *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pp. 912–920, Association for Computational Linguistics, 2010.
- [17] W. Yin and Y. Pei, “Optimizing sentence modeling and selection for document summarization,” in *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence*, pp. 1383–1389, 2015.
- [18] C. Shen and T. Li, “Multi-document summarization via the minimum dominating set,” in *Proceedings of the 23rd International Conference on Computational Linguistics*, pp. 984–992, Association for Computational Linguistics, 2010.
- [19] A. Gainer-Dewar and P. Vera-Licona, “The minimal hitting set generation problem: algorithms and computation,” *SIAM Journal on Discrete Mathematics*, vol. 31, no. 1, pp. 63–100, 2017.
- [20] E. Boros, V. Gurvich, L. Khachiyan, and K. Makino, “Dual-bounded generating problems: weighted transversals of a hypergraph,” *Discrete Applied Mathematics*, vol. 142, no. 1, pp. 1–15, 2004.
- [21] L. A. Wolsey, “An analysis of the greedy algorithm for the submodular set covering problem,” *Combinatorica*, vol. 2, no. 4, pp. 385–393, 1982.
- [22] A. Gainer-Dewar and P. Vera-Licona, “The minimal hitting set generation problem: algorithms and computation,” *SIAM Journal on Discrete Mathematics*, vol. 31, no. 1, pp. 63–100, 2017.
- [23] M. F. Porter, “Snowball: A language for stemming algorithms.” <http://www.snowball.tartarus.org/texts/introduction.html>, 2001. Accessed 15 November 2017.



- [24] G. Salton and C. Buckley, “Term-weighting approaches in automatic text retrieval,” *Information processing & management*, vol. 24, no. 5, pp. 513–523, 1988.
- [25] S. Arora, R. Ge, and A. Moitra, “Learning topic models—going beyond svd,” in *Foundations of Computer Science (FOCS), 2012 IEEE 53rd Annual Symposium on*, pp. 1–10, IEEE, 2012.
- [26] R. L. Cilibrasi and P. M. Vitányi, “The google similarity distance,” *IEEE Transactions on knowledge and data engineering*, vol. 19, no. 3, 2007.
- [27] R. Cilibrasi and P. M. Vitányi, “Clustering by compression,” *IEEE Transactions on Information theory*, vol. 51, no. 4, pp. 1523–1545, 2005.
- [28] M. Ester, H.-P. Kriegel, J. Sander, X. Xu, *et al.*, “A density-based algorithm for discovering clusters in large spatial databases with noise,” in *KDD’96 Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, vol. 96, pp. 226–231, 1996.
- [29] L. Rokach and O. Maimon, “Clustering methods,” in *Data mining and knowledge discovery handbook*, pp. 321–352, Springer, 2005.
- [30] R. M. Karp, “Reducibility among combinatorial problems,” in *Complexity of computer computations* (R. Miller, ed.), pp. 85–103, Springer, 1972.
- [31] J. Leskovec, A. Krause, C. Guestrin, C. Faloutsos, J. VanBriesen, and N. Glance, “Cost-effective outbreak detection in networks,” in *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 420–429, ACM, 2007.
- [32] J. Carbonell and J. Goldstein, “The use of mmr, diversity-based reranking for re-ordering documents and producing summaries,” in *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, pp. 335–336, ACM, 1998.
- [33] H. T. Dang, “Overview of duc 2005,” in *Proceedings of the document understanding conference*, 2005.
- [34] T. D. Hoa, “Overview of duc 2006,” in *Proceedings of the document understanding conference*, 2006.
- [35] H. T. Dang, “Overview of the duc 2007 summarization task,” in *Proceedings of the document understanding conference*, 2007.
- [36] C.-Y. Lin and E. Hovy, “Automatic evaluation of summaries using n-gram co-occurrence statistics,” in *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1*, pp. 71–78, Association for Computational Linguistics, 2003.
- [37] A. Lancichinetti, S. Fortunato, and J. Kertész, “Detecting the overlapping and hierarchical community structure in complex networks,” *New Journal of Physics*, vol. 11, no. 3, p. 033015, 2009.
- [38] D. Zajic, B. Dorr, R. Schwartz, C. Monz, and J. Lin, “A sentence-trimming approach to multi-document summarization,” in *Proceedings of HLT/EMNLP 2005 Workshop on Text Summarization (HLT/EMNLP 05)*, pp. 151–158, 2005.