Information Sciences 496 (2019) 212–224

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

Learning with fuzzy hypergraphs: A topical approach to query-oriented text summarization

Hadrien Van Lierde, Tommy W.S. Chow*

Department of Electronic Engineering, City University of Hong Kong, 83 Tat Chee Av., Kowloon Tong, Hong Kong, China

ARTICLE INFO

Article history:
Received 6 November 2017
Revised 10 May 2019
Accepted 10 May 2019
Available online 11 May 2019

Keywords:
Automatic text summarization
Fuzzy graphs
Probabilistic topic models
Hierarchical Dirichlet process
Personalized PageRank
Submodular set functions

ABSTRACT

Existing graph-based methods for extractive document summarization represent sentences of a corpus as the nodes of a graph in which edges depict relationships of lexical similarity between sentences. This approach fails to capture semantic similarities between sentences when they express a similar information but have few words in common and are thus lexically dissimilar. To overcome this issue, we propose to extract semantic similarities based on topical representations of sentences. Inspired by the Hierarchical Dirichlet Process, we propose a topic model to infer topic distributions of sentences. As each topic defines a semantic connection among sentences with a certain degree of membership for each sentence, we propose a fuzzy hypergraph model in which nodes are sentences and fuzzy hyperedges are topics. To produce an informative summary, we extract a set of sentences from the corpus by simultaneously maximizing their relevance to a user-defined query, their centrality in the fuzzy hypergraph and their coverage of topics present in the corpus. We formulate an algorithm building on the theory of submodular functions to solve the associated optimization problem. A thorough comparative analysis with other graph-based summarizers demonstrates the superiority of our method in terms of content coverage of the summaries.

© 2019 Elsevier Inc. All rights reserved.

1. Introduction

The rapid expansion of the Internet led to a substantial increase in the amount of publicly available textual resources in recent years. The availability of information in the form of online documents such as news articles or legal texts facilitates decision processes in fields ranging from finance to legal matters. Automatic text summarization speeds up the process of information extraction by automatically producing summaries of large corpora. While early methods were restricted to the summarization of single documents, recent approaches focused on the more realistic problem of multi-document summarization [20]. Existing summarizers can be further classified into generic summarizers, which produce general purpose document summaries, and query-oriented summarizers, which produce summaries with the information relevant to a query formulated by the user. Since in various applications, including legal text summarization or the summarization of finance-related documents, users are seeking answers to specific questions into corpora of documents, we are proposing a query-oriented multi-document summarizer.

While an abstractive summarizer generates an abstract of a corpus based on natural language generation, extractive summarizers produce summaries by extracting and aggregating relevant sentences of the original corpora. The large majority of

* Corresponding Author.
E-mail addresses: hvanlierd2-c@my.cityu.edu.hk (H. Van Lierde), eetchow@cityu.edu.hk (T.W.S. Chow).

https://doi.org/10.1016/j.ins.2019.05.020
0020-0255/© 2019 Elsevier Inc. All rights reserved.

algorithms build on the extractive approach since it does not require extensive Natural Language Processing. Among these algorithms, graph-based summarizers have proved to outperform feature-based methods in various experiments [20] due to their ability to capture the global structure of connections between sentences of a corpus in the calculation of sentence scores. In their simplest form, graph-based summarizers first define a graph in which vertices are sentences and edges represent pairwise lexical similarities between sentences, namely similarities based on the number of words that sentences have in common. Then sentence scores are obtained by applying popular graph-based ranking algorithms such as the PageRank [21] or the HITS algorithm [29]. Since a simple graph cannot model complex collective relationships among multiple sentences, hypergraph models were also proposed [30,32], which capture groups of lexically similar sentences and then apply hypergraph extensions of ranking algorithms.

Two limitations of existing graph- and hypergraph-based algorithms alter their summarization capabilities [4,8,15,25,29,30,32–34]: the *semantic* limitation and the lack of *topical diversity*. First, the calculation of similarities between sentences is generally based on the co-occurrence of terms in sentences (lexical similarity) rather than their *semantic* relatedness [8,29]. However, sentences with no or few words in common may still refer to the same topic in the context of a specific corpus, as shown by the following example of two sentences: (1) ”after landing, the airplane slowly moved on the track until it stopped at its parking place,” and (2) ”the aircraft reached a designated area and the passengers got off”. Although they provide slightly different pieces of information, both sentences are semantically related as they share semantically related terms. However, they do not have any word in common, except stopwords. Attempts to incorporate higher order relationships among sentences include the detection of clusters of lexically similar sentences, namely groups of sentences with a large number of words in common [4,29,30,34]. However, they do not attempt to detect sets of semantically related terms or topics. As a result, they fail to capture pairwise semantic similarities between sentences when they use very different wordings.

Second, most systems include a greedy sentence selection method for redundancy removal in which sentences are considered redundant only if they have words in common [32]. Other approaches include methods simultaneously maximizing relevance and minimizing redundancy [15,33] and methods based on the detection of dominating sets [25]. These different approaches build on lexical similarities between sentences as a measure of their redundancy. However, as shown in the example above, lexically dissimilar sentences might still be semantically related. Hence, with existing algorithms of redundancy removal, the resulting summary might consist of sentences that refer to the same topic and fail to cover all major topics of the given corpus. A new approach is thus needed to enforce the *topical diversity* of the summaries instead of removing lexical redundancies.

To address the *semantic* limitation of existing systems, we use a probabilistic topic model called the Hierarchical Dirichlet Process [26]. We adapt the model for the inference of multiple topic tags for each sentence of a corpus. Since each topic connects a group of semantically related sentences, the sentences are represented as the nodes of a fuzzy hypergraph in which the topics represent fuzzy hyperedges. A recent idea proposed in [32] also incorporates topics inferred by a topic model in a hypergraph-like structure. It clusters sentences based on their topical representations and the resulting disjoint communities are modelled as crisp and disjoint hyperedges of a hypergraph instead of fuzzy hyperedges. Hence, it fails to capture the multiplicity of topics covered by sentences.

To address the issue of *topical diversity*, we propose a new sentence selection approach which extracts the sentences by maximizing the *Relevance* and the *Topical Coverage*. The *Relevance* of individual sentences expresses both their similarity with the query and their centrality in the fuzzy hypergraph. The *Topical Coverage* of a set of sentences expresses the multiplicity and the diversity of topics covered by these sentences. Our definition of Topical Coverage is based on an extension to our fuzzy hypergraph of the dominating set problem [9]. Hence, instead of removing lexical redundancies, we intend to improve the topical diversity of our summary, which is more consistent with the goal of covering all major topics of a given corpus. An algorithm based on the theory of submodular functions is proposed to solve the related optimization problem. This core algorithm of sentence selection is called *Maximum Relevance and Coverage* (MRC) algorithm.

The main contributions of this paper are the following: (1) a new fuzzy hypergraph model capturing semantic relationships among sentences, (2) a sentence selection approach based on the maximization of Relevance of individual sentences and Topical Coverage, and (3) a polynomial time algorithm building on the theory of submodular functions for solving the optimization problem.

The structure of the paper is as follows. In Section 2, we present summarization algorithms related to ours. In Section 3, we present each step of our framework. Finally, in Section 4, we provide our experimental results.

2. Related work

Extractive summarizers aggregate important sentences in a corpus while abstractive summarizers generate new summaries after identifying important information [20]. As abstractive summarization requires extensive Natural Language Processing, most summarizers to date are based on extractive approaches.

Methods of extractive summarization generally fall into two categories, namely feature-based and graph-based approaches. Feature-based methods train a model to predict the score of each sentence based on feature representations of sentences (term frequency, sentence position [20], etc.). Graph-based methods define graphs in which the nodes are the sentences and the edges represent similarities between sentences. Sentence scores are then given by node centrality measures on the graph [8,21]. The advantages of graph-based summarization over feature-based summarization are that it does not require labelled corpora, and that it is based on the global structure of links between sentences of the corpus rather than local features.

The earliest graph-based summarizer, called LexRank [8], defines edges as term co-occurrence relationships between sentences. Then, PageRank algorithm is applied to compute relevance scores of sentences. Similarly, TextRank method [16] also defines a graph using lexical similarities between sentences as edges, and it applies graph-based ranking algorithms for sentence scoring. For query-oriented summarization, topic sensitive LexRank [21] introduces a query bias in the probabilities of transition. Finally, a bipartite graph model is proposed in [29], involving both sentences and terms as vertices and it applies HITS algorithm to score sentences. In [31] this idea is combined with a PageRank-like method to score sentences, terms and documents simultaneously.

While early methods build sentence graphs based on the co-occurrence of terms in sentences only, later approaches infer higher level relationships. In that perspective, Wan and Yang [29] build a bipartite graph in which the vertices are sentences and sentence clusters which are both scored using the HITS algorithm. A similar idea presented in [34] incorporates terms as a third class of vertices. In contrast, [4] builds on the idea that scores of sentences within each cluster should be quite different from each other. Finally, Wang et al. [30] propose a hypergraph in which hyperedges represent clusters and nodes are sentences which are scored using semi-supervised learning. However, their method is limited to disjoint sentence clusters which poorly capture the multiplicity of topical relationships among sentences.

In contrast, several summarizers propose to build on topic models rather than clusters. In the context of text summarization, each sentence is tagged with multiple topics. Popular topic modelling algorithms include Probabilistic Latent Semantic Analysis (PLSA), Latent Dirichlet Allocation (LDA) and the Hierarchical Dirichlet Process (HDP). Hennig and Labor [10] compute the similarity of sentences with a query based on PLSA. Going beyond PLSA, Arora and Ravindran [1] extracts topic distributions of sentences based on LDA and, for each topic, it selects the sentence with highest associated probability. Similarly, T-Graph [22] defines a bipartite graph in which the two vertex classes are sentences and topics discovered by LDA. A hypergraph model similar to ours was presented in [32]. Non-overlapping clusters of topically homogeneous sentences define binary and disjoint hyperedges that do not capture the multiplicity of topics covered by a sentence, which can only be captured by overlapping and fuzzy hyperedges as the ones present in our model.

Building on fuzzy set theory, fuzzy graphs associate each node with a degree of membership in each edge [17]. Relaxing the assumption of pairwise relationships, fuzzy hypergraphs are defined by a set of nodes and a set of fuzzy subsets of these nodes. Applications of fuzzy hypergraphs include portfolio management and managerial decision making [17]. To our knowledge, fuzzy hypergraphs have not yet been used for text mining purposes, including text summarization. Fuzzy hypergraphs are used to incorporate topical information in our summarizer.

After sentence scoring, a critical step is to select highly scored sentences that are not redundant. A popular method is the greedy method of redundancy removal which selects dissimilar sentences with highest scores [32]. As this method may favour long sentences, multi-objective approaches were proposed in order to maximize the sum of relevance scores of selected sentences and simultaneously minimize their redundancy [15,33]. However, their definition of redundancy is limited to lexical similarities. Other methods include the one in [25], which selects sentences by solving the dominating set problem over the sentence graph. However, their algorithm also tends to favour long sentences over short ones and it fails to model semantic relationships captured by topics. In general, existing methods of redundancy removal are merely based on lexical similarities between sentences which does not prevent semantic redundancies in the final summary. Finally, the method in [28] proposes to extract sentences by generating a minimum hitting set consisting of the smallest set of sentences covering all topics present in the corpus.

Finally, some recent methods are based on advances in deep learning and learnt sentence representations. For instance, Kageback and Mogren [12] produce phrase embeddings by combining word embeddings derived from a continuous vector space model. Similarly, Van Lierde and Chow [27] propose a summarizer that incorporates word embeddings in a probabilistic topic model to measure sentence similarities, and sentences are then scored using a PageRank-based node ranking algorithm. Supervised methods were also proposed such as [5] which introduces a recursive neural network using word and sentence-level features to produce relevance scores, Nallapati et al. [18] which proposes a classifier based on a recurrent neural network for selecting relevant sentences, or Narayan et al. [19] which proposes a supervised sentence extractor based on a recurrent neural network which is trained using a reinforcement learning framework.

3. Maximizing relevance and topical coverage based on a sentence fuzzy hypergraph

The query-oriented summarization task is defined below (Definition 1). An extractive summary is referred to as a set of selected sentences S. The prescribed summary length is the so-called *capacity* of the summary. We also refer to the set of terms of a corpus as the set of distinct words appearing at least once in the corpus. Fig. 1 provides an overview of our system. In what follows, we describe each step of our MRC algorithm in details, including the preprocessing, the topic modelling, the fuzzy hypergraph construction and the sentence selection step.

Definition 1 (Query-oriented summarization problem)**.** Given a corpus of documents consisting of a set V of sentences, the set $\{l(s)\colon s \in V\}$ of sentence lengths, a summary capacity $L > 0$ and a query represented by a sentence q, produce a summary S in which $S \subseteq V$ is a set of selected sentences that are relevant to q and contain the essential information of V, such that the capacity constraint $\sum_{s\in S} l(s) \leq L$ is satisfied.

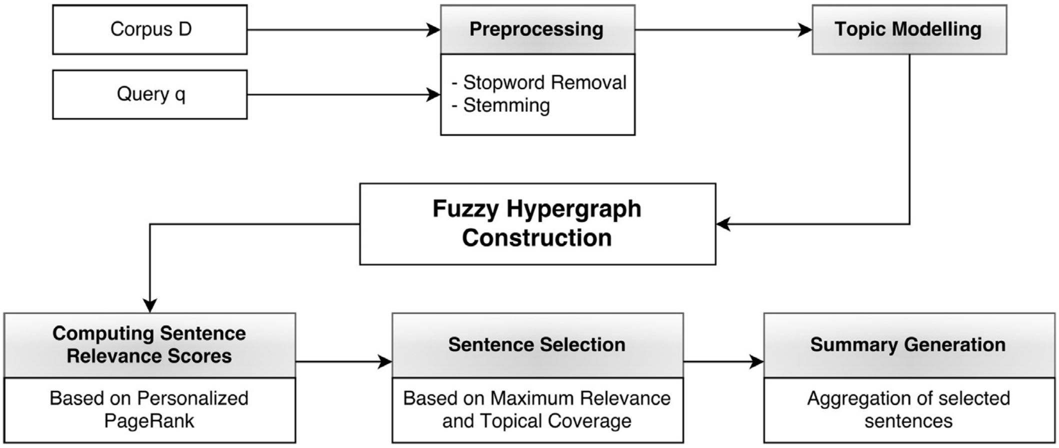

Corpus D
Query q
Preprocessing
- Stopword Removal
- Stemming
Topic Modelling
Fuzzy Hypergraph Construction
Computing Sentence Relevance Scores
Based on Personalized PageRank
Sentence Selection
Based on Maximum Relevance and Topical Coverage
Summary Generation
Aggregation of selected sentences

Fig. 1. System chart.

3.1. Preprocessing

We apply standard preprocessing methods in text mining including stopword removal based on a publicly available list of 153 English stopwords [23] and word stemming using Porter Stemmer[1] We let N_t represent the number of distinct terms in the corpus after these preprocessing operations are completed.

3.2. Topic inference

As mentioned in Sections 1 and 2, traditional graph-based summarization algorithms only take into account the co-occurrence of terms between sentences. In order to capture the semantic similarity between sentences, we propose to first extract the topics present in the corpus. Previous attempts to incorporate topical information in Automatic Summarizers include the use of Probabilistic Latent Semantic Analysis [10] which inevitably leads to overfitting [26], or Latent Dirichlet Allocation [1,22] which requires setting the number of topics manually. In contrast, we rely on the Hierarchical Dirichlet Process (HDP) [26], which is a probabilistic topic model capable of inferring the number of topics automatically.

Given a set of documents with N_t distinct terms, HDP infers a finite number K of topics in the form of probability distributions over terms $\phi_1, \ldots, \phi_K \in [0, 1]^{N_t}$ and a topic tag $z_{lj} \in \{1, \ldots, K\}$ for each word l in document j. Previous HDP-based extractive summarizers relied on single level HDP, namely inferring the topic distributions at document level, then defining the topics of each sentence as the topic tags of its words. However, due to the so-called *exchangeability* assumption [26], this approach neglects the topical information jointly carried by words within a sentence. Hence, we rather make use of a 2-level HDP ensuring that topics are shared across documents, across sentences and within sentences. In this generative model, given a base distribution H and dispersion parameters α, β and γ whose values are determined experimentally as shown in Section 4.1, a global measure G_0 is drawn from a Dirichlet Process $DP(\gamma, H)$. Then, for each document j, a document specific measure G_j is drawn from $DP(\beta, G_0)$ and, for each sentence i, a sentence specific measure G_{ij} is drawn from $DP(\alpha, G_j)$. Finally, for each word l in sentence i of document j, we generate a distribution over terms $\theta_{lij} \sim G_{ij}$, and a term is drawn from a categorical distribution $\text{cat}(\theta_{lij})$. The inference process based on the Gibbs sampler [26] yields (1) a number K of topics represented by a distribution over terms $\phi_e \in [0, 1]^{N_t}$ for each topic e, where N_t is the number of distinct terms in the corpus and ϕ_{et} is the probability of observing term t under topic e, and (2) a topic tag $z_{lij} \in \{1, \ldots, K\}$ for each word l in sentence i of document j. A Dirichlet prior $\text{dir}(\zeta \frac{\mathbf{1}_{N_t}}{N_t})$ is chosen as a base distribution H (the value of ζ is validated in Section 4.1).

3.3. Fuzzy hypergraph definition

A hypergraph $H = (V, E)$ over a set V of vertices is a generalization of graph in which each hyperedge in E is a subset of V [30]. In existing hypergraph-based summarizers [30,32], vertices are sentences and clusters of sentences correspond to hyperedges which do not overlap. There is also no attempt to model the degree of membership of each sentence in each hyperedge. This model is unsatisfactory since each sentence may cover multiple topics, and each topic is covered by a sentence with a different degree depending on the number of words of the sentence tagged with this topic. To overcome these limitations, we propose a model based on a fuzzy hypergraph, namely a generalization of hypergraph in which hyperedges

[1] M. F. Porter, Snowball: A language for stemming algorithms, Available at: http://www.snowball.tartarus.org/texts/introduction.html, 2001.

are defined as fuzzy subsets of the set of nodes. Below is a formal definition of fuzzy hypergraph, which is an adaptation of the one in [17].

Definition 2 (Fuzzy Hypergraph)**.** A fuzzy hypergraph is defined as a quadruplet $G = (V, E, \psi, w)$ on a set V of vertices and a set E of hyperedges such that $\psi \in [0,\ 1]^{|E|\times|V|}$ is a matrix that defines a distribution over vertices for each of the $|E|$ hyperedges, verifying $\sum_{i\in V}\psi_{ei} = 1$ for $e \in E$ and $\Sigma_{e\in E}\psi_{ei} > 0$ for $i \in V$, and a vector positive hyperedge weights $w \in \mathbb{R}^{|E|}$.

Matrix ψ defines the incidence matrix of the fuzzy hypergraph. In the context of our summarization method, we define a fuzzy hypergraph $G = (V, E, \psi, w)$ in which vertices are sentences and each fuzzy hyperedge represents a topic. The degree of membership of each sentence in a hyperedge is proportional to the number of words tagged with the corresponding topic in the sentence, i.e. $\psi_{ei} = \frac{|\{l:z_{li}=e\}|}{|\{l:z_{lj}=e,1\leq j\leq N_s\}|}$. For simplicity, we dropped document index j and we denote by z_{li} the topic of lth word in ith sentence. Hence, unlike in previous hypergraph-based approaches, we make the more realistic assumption that each sentence can belong to several semantic groups (i.e. topics) with a certain degree of membership in each group.

Next, we define the weight $w(e)$ of a fuzzy hyperedge e based on four aspects. The *in-corpus frequency* $\mathrm{tfc}(t)$ of term t in the corpus is the number of times t appears in the corpus. The *sentence discriminatory power* $\mathrm{isf}(t)$ [2] of term t is given by the logarithm of the inverse sentence frequency $\mathrm{isf}(t) = \log\left(\frac{N_s}{N_s^t}\right)$, where N_s is the total number of sentences and N_s^t is the number of sentences containing term t. As in [2], the isf function yields a smaller weight for terms occurring in a large number of sentences. The *in-topic frequency* $\mathrm{tft}(t,\ e)$ of term t in topic e is the probability of encountering term t conditioned on e, i.e. $\mathrm{tft}(t, e) = \phi_{et}$. The *topic discriminatory power* $\mathrm{tdp}(t)$ of a term t is based on the idea that a term t appearing in relatively few topics has a significant contribution to the semantics of sentences, while terms appearing in a large number of topics might have ambiguous meanings. We quantify $\mathrm{tdp}(t)$ based on the entropy of the distribution over topics $H(t) = -\sum_e p(e|t)\log(p(e|t))$, where $p(e|t)$ measures the fraction of occurrences of term t in the corpus that are tagged with topic e. Then, the $\mathrm{tdp}(t)$ is given by a shifted inverse of the entropy of this distribution, i.e. $\mathrm{tdp}(t) = \frac{1}{1+H(t)}$ which equals 1 if t is only tagged with a single topic in the whole corpus. Finally the *relevance* $\mathrm{rel}(e)$ of topic e is given by $\mathrm{rel}(e) = \mathrm{f}(e)\log\left(\frac{N_s}{N_s^e}\right)$. where $\mathrm{f}(e)$ is the number of occurrences of topic e in the corpus and N_s^e is the number of sentences in which topic e occurs. The relevance $\mathrm{rel}(e)$ of topic e is an adaptation of the tfisf weights for weighting topics instead of terms [2].

The weights of hyperedges are obtained by combining all the above scores:

$$w(e) = \mathrm{rel}(e)\sum_t \mathrm{tfc}(t)\mathrm{isf}(t)\mathrm{tft}(t, e)\mathrm{tdp}(t). \tag{1}$$

This definition yields a high weight for frequent topics including terms that occur a large number of times in the corpus, have strong discriminatory power over sentences and are not semantically ambiguous. As opposed to previous topic-based summarization algorithms [1,32], we take advantage of the representation of topics as distributions over terms in order to compute the topic weights. Algorithm 1 summarizes the step of the fuzzy hypergraph construction. The computational complexity of the algorithm is $O(K(N_s + N_t))$ where K is the number of topics.

Algorithm 1: Fuzzy Hypergraph Construction.

```
INPUT: for $1 \leq e \leq K$, $\phi_e$, $N_s^e$ (number of sentences tagged with topic $e$) and $f(e)$; for $1 \leq t \leq N_t$, $N_s^t$ (number of sentences containing term $t$); for $1 \leq i \leq N_s$, topic tags $z_{wi}$ for each word $w$ in sentence $i$,
OUTPUT: Hypergraph $H(\{1, \ldots, N_s\}, \{1, \ldots, K\}, \psi, w)$
for each $t \in \{1, \ldots, N_t\}$:
    Compute $\mathrm{tfc}(t)$, $H(t)$, $\mathrm{isf}(t)$, $\mathrm{tdp}(t)$
    for each $e \in \{1, \ldots, K\}$: $\mathrm{tft}(t, e) \leftarrow \phi_{et}$
for each $e \in \{1, \ldots, K\}$:
    $\mathrm{rel}(e) \leftarrow \mathrm{f}(e)\log\left(\frac{N_s}{N_s^e}\right)$, $w(e) \leftarrow \mathrm{rel}(e)\sum_t \mathrm{tfc}(t)\mathrm{isf}(t)\mathrm{tft}(t, e)\mathrm{tdp}(t)$
    for each $i \in \{1, \ldots, N_s\}$: $\psi_{ei} \leftarrow \frac{|\{l:z_{li}=e\}|}{|\{l:z_{lj}=e,1\leq j\leq N_s\}|}$
```

3.4. Relevance and coverage maximization for sentence selection

Each sentence is ranked in terms of its relevance to the query and its centrality in the whole corpus. Sentences are then selected by maximizing their individual Relevance and their joint Topical Coverage.

3.4.1. Computing relevance scores of sentences

We introduce an algorithm that scores sentences according to their relevance to the query and their centrality in the fuzzy hypergraph. Most graph-based summarization algorithms score sentences using PageRank-like algorithms [8,21]. The underlying assumption is that the generation of a coherent text from isolated sentences can be modelled as a Markov chain in which the states are sentences and the probability of transition between two sentences depends on their similarity in some sense. Stationary probabilities provide the sentence ranks in the context of generic summarization. We extend this method by defining a random walk over fuzzy hypergraphs in which the transition probability between two vertices depends on the hyperedges shared by these vertices. The transition from vertex i to another vertex is performed in two steps:

1. draw a hyperedge $e \in E$ with probability $p(e|i) = \frac{p(i|e)w(e)}{\sum_f p(i|f)w(f)} = \frac{\psi_{ei}w(e)}{\sum_f \psi_{fi}w(f)}$,
2. draw a vertex j in V with probability $p(j|e) = \psi_{ej}$.

Integrating out the hyperedges, we obtain the probability of transition

$$p(j|i) = \sum_e p(j|e)\frac{p(i|e)w(e)}{\sum_f p(i|f)w(f)} \tag{2}$$

from vertex i to vertex j. In contrast to previous graph-based summarizers which were based on lexical similarities, our proposed probabilities of transition depend on the co-occurrence of topics between pairs of sentences. Moreover, since we intend to extract sentences that are both central in the corpus and relevant to a user-defined query, we introduce a query bias in the probabilities of transition as in [21]. Given probabilities of transition $p(j|q)$ from the query sentence q to any sentence j, the query-biased probability of transition from i to j is

$$p^q(j|i) = (1-\lambda)p(j|q) + \lambda p(j|i) \tag{3}$$

where $\lambda \in [0, 1]$ is the *query balance*, i.e. the weight of the query in the scoring process. We propose a topic-based query-relevance measure defined by $p(j|q) = \sum_t \sum_e \psi_{ej} p(e|t)p(t|q)$, where $(\psi_{ej})_{\substack{1\le e\le K \\ 1\le j\le N_s}}$ is the incidence matrix of the fuzzy hypergraph as defined in Section 3.3, $p(e|t)$ is the fraction of occurrences of term t that are tagged with topic e, and $p(t|q)$ is proportional to the frequency of term t in the query. The query-biased probabilities of transition result in higher scores for the sentences that are semantically similar to the query. This topic-based query relevance goes beyond the lexical similarity that is generally used in other systems [30,32]. The final scores $\{p(i)\colon 1 \le i \le N_s\}$ are obtained by the PageRank algorithm:

$$p^T(j) = (1-\mu)\frac{\mathbf{1}_{N_s}}{N_s} + \mu\sum_{\substack{i=1\\ i\neq j}}^{N_s} p^q(j|i)p^{T-1}(i),\; T = 1, 2, \ldots \tag{4}$$

where $\mathbf{1}_{N_s}$ is a vector of ones and $\mu \in [0, 1]$ is the so-called damping factor [8]. If $\mu > 0$, the Markov chain is ergodic and the algorithm is guaranteed to converge to a unique vector p with positive entries for any initial probability vector p^0 [8].

3.4.2. Sentence selection

The sentence scores described in preceding section are further used to select sentences to be included in the summary. To avoid redundancies, previous graph-based summarizers proposed a greedy algorithm whereby sentences are selected in decreasing order of scores provided that the pairwise similarity between selected sentences does not exceed a threshold [32]. However, there is no guarantee that the resulting summary properly covers all the important topics present in the corpus. To that end, we propose to select sentences by maximizing what we call their *Topical Coverage*. Our aim is to ensure that each sentence in the corpus shares a sufficient number of topics with the summary. In probabilistic terms, we measure this as the probability $P(S|i)$ for a random walker starting in a vertex i to reach the set S of selected vertices in at most one step, i.e.

$$p(S|i) = \begin{cases} \sum_{j\in S}\sum_e \psi_{je}\frac{\psi_{ie}w(e)}{\sum_f \psi_{if}w(f)} & \text{if } i \notin S \\ 1 & \text{if } i \in S. \end{cases} \tag{5}$$

The *Topical Coverage* $C(S)$ is thus defined as the sum of these probabilities over all vertices, namely

$$C(S) = \sum_{i\in V} p(S|i) = |S| + \sum_{\substack{j\in S\\ i\notin S}}\sum_e \psi_{je}\frac{\psi_{ie}w(e)}{\sum\limits_f \psi_{if}w(f)}. \tag{6}$$

Maximizing the Topical Coverage ensures that each sentence in the corpus is sufficiently similar to sentences in the summary. The corresponding decision problem can be viewed as a generalization of the dominating set problem in the case of fuzzy hypergraphs [9]. We may give another interpretation of topical coverage. When maximizing $C(S)$, the first term in Eq. (6) encourages to select short sentences which balances the fact that long sentences tend to have higher relevance scores. The second term of $C(S)$ can be written as

$$\sum_e \sum_{\substack{j\in S\\ i\notin S}} p(j|e)p(e|i) \tag{7}$$

which encourages hyperedges to have a balanced number of incident vertices respectively in S and in $V\backslash S$. This implies that each topic is indeed covered by sentences in S while reducing the risk of including semantically redundant sentences covering the exact same topics. For this reason, we refer to $C(S)$ as the *Topical Coverage* of S.

Combining *Relevance* and *Topical Coverage*, our proposed method seeks sentences that are individually relevant and that jointly cover the semantic content of the corpus. This translates into a multi-objective discrete optimization problem (Definition 3) which is proved to be NP-hard in Theorem 1.

Definition 3 (Maximum Relevance and Coverage Problem (MRC))**.** Given a set V of sentences extracted from a corpus, a summary capacity L and a set of relevance scores $\{p(s)\colon s \in V\}$, the Maximum Relevance and Coverage Problem is

$$\max_{S\subseteq V} \; (1-\nu)\sum_{s\in S} p(s) + \frac{\nu}{N_s} C(S), \text{ subject to } \sum_{s\in S} l(s) \leq L \tag{8}$$

where $\{l(s)\colon s \in V\}$ are the sentence lengths, $\nu \in [0, 1]$ and $N_s = |V|$.

Theorem 1. *The decision problem associated to MRC is NP-hard.*

Proof. We show that the $0-1$ Knapsack problem reduces to MRC problem. Given any instance P_1 of $0-1$ Knapsack problem with item set I, item values $\{v(s)\colon s \in I\}$, item weights $\{w(s)\colon s \in I\}$ and maximum weight W, we build an instance P_2 of MRC, with sentence set I, sentence lengths $\{w(s)\colon s \in I\}$, sentence relevance scores $\{v(s)\colon s \in I\}$, capacity W, and $\nu = 0$. Then, solving P_1 is equivalent to solving P_2, and thus Knapsack problem reduces to MRC. □

As MRC problem is NP-hard, we provide a polynomial time algorithm for it, with a constant approximation factor. As for various other approximation algorithms, our method is based on the modularity and the non-decreasing property of the MRC objective function (as defined in [13]), which are proved in Theorem 2.

Theorem 2. *The objective function $F\colon P(V) \to [0, 1]$ of MRC problem (Eq. (8)) is submodular and monotonically non-decreasing.*

Proof. Let V be the set of sentences in the corpus, $S \subseteq V$ be the selected sentences and $R(S) = \sum_{s\in S} p(s)$. Then, F becomes $F(S) = (1-\nu)R(S) + \frac{\nu}{N_s} C(S)$. Also let $p(j|i) = \sum_e \psi_{je} \frac{\psi_{ie} w(e)}{\sum_f \psi_{if} w(f)}$. Defining $F(\emptyset) = 0$, we have $\forall S \subset V$ and $\forall r \in V\backslash S$

$$\begin{aligned} N_s F(S\cup\{r\}) &= (1-\nu)N_s R(S\cup\{r\}) + \nu C(S\cup\{r\}) \geq (1-\nu)N_s R(S) + \nu\Big(|S| + \sum_{j\in S} p(j|r) + \sum_{\substack{j\in S\\ i\notin S\cup\{r\}}} p(j|i) + \sum_{i\notin S\cup\{r\}} p(r|i)\Big) \\ &\geq (1-\nu)N_s R(S) + \nu\Big(|S| + \sum_{\substack{j\in S\\ i\notin S}} p(j|i)\Big) = N_s F(S) \end{aligned} \tag{9}$$

which proves that F is monotonically non-decreasing. To prove F is submodular, we observe that $\forall S \subseteq T \subset V$ and $r \in V\backslash T$

$$N_s((F(S\cup\{r\}) - F(S)) - (F(T\cup\{r\}) - F(T))) = \nu\left(\sum_{i\notin S\cup\{r\}} p(r|i) - \sum_{i\notin T\cup\{r\}} p(r|i)\right) + \nu\left(\sum_{j\in T} p(j|r) - \sum_{j\in S} p(j|r)\right). \tag{10}$$

Considering the first term in Eq. (10), we have $\sum_{i\notin S\cup\{r\}} p(r|i) - \sum_{i\notin T\cup\{r\}} p(r|i) = \sum_{i\in T\backslash S} p(r|i) \geq 0$. For the second term, we have $\sum_{j\in T} p(j|r) - \sum_{j\in S} p(j|r) = \sum_{j\in T\backslash S} p(j|r) \geq 0$ which completes the proof of submodularity. □

We yield the approximation Algorithm 2 for solving MRC problem based on a general approach proposed by Lin and Bilmes [15] for the maximization of monotonically non-decreasing submodular functions under budget constraints. We prove in Theorem 3 that Algorithm 2 provides a near-optimal solution to MRC problem with a relative performance guarantee. The proof relies on the submodularity and non-decreasing property proved in Theorem 2. The time complexity of Algorithm 2 is dominated by the computation of relevance scores and the sentence selection step which have a time complexity of $O(\tau N_s^2)$ where τ is the number of iterations for the iterative computation of relevance scores. The final summary is produced by aggregating the sentences selected by Algorithm 2.

Theorem 3. *Let F be the objective function of MRC problem, then Algorithm 2 produces a summary S^P verifying*

$$F(S^P) \geq (1 - e^{-\frac{1}{2}})F(S^*) \tag{11}$$

where S^* is the optimal solution of MRC problem.

Proof. Theorem 1 in [15] states that, for the maximization of any submodular and monotonically non-decreasing function f under a Knapsack constraint, namely

$$\max_{S\subseteq V} f(S) \text{ subject to } \sum_{i\in S} w_i \leq L, \tag{12}$$

Algorithm 2: Maximal Relevance and Coverage (MRC) Algorithm.

```
INPUT: Set $V$ of sentences, parameter $\nu$, capacity $L$, sentence lengths $\{l_s : 1 \leq s \leq N_s\}$, Hypergraph $H(\{1,\ldots,N_s\},\{1,\ldots,K\},\psi,w)$.
OUTPUT: Set $S^P$ of indices of sentences to be included in the summary.
for each $j, i \in \{1,\ldots,N_s\}$: compute $p(j|i)$ and $p^q(j|i)$ (Eqs. (2)-(3))
Compute sentence relevance scores $\{p_i : 1 \leq i \leq N_s\}$ (Eq. (4))
Let $Z \leftarrow V$, $T \leftarrow \emptyset$, $\rho \leftarrow 0$
for each $j \in \{1,\ldots,N_s\}$: $\pi_j \leftarrow \frac{1}{l_j}\left(\frac{\nu}{N_s}(1+\sum_{i\neq j} p(j|i)) + (1-\nu)p_j\right)$
while $Z \neq \emptyset$:
    $s^* \leftarrow \underset{s\in Z}{\text{argmax}}\ \pi_s$, $Z \leftarrow Z \setminus \{s^*\}$
    if $\rho + l_{s^*} \leq L$:
        $T \leftarrow T \cup \{s^*\}$, $\rho \leftarrow \rho + l_{s^*}$
        for each $j \in \{1,\ldots,N_s\}$: $\pi_j \leftarrow \pi_j - \frac{\nu}{N_s l_j}(p(s^*|j) + p(j|s^*))$.
Let $Q \leftarrow \{\{s\}\colon l(s) \leq L, s \in V\}$
Let $S^P \leftarrow \underset{S\in\{T\}\cup Q}{\text{argmax}}(1-\nu)\sum_{s\in S} p(s) + \frac{\nu}{N_s}\left(|S| + \sum_{j\in S, i\notin S} p(j|i)\right)$
```

with optimal solution S^*, an approximate solution S_F satisfying $f(S^F) \geq f(S^*)(1 - e^{\frac{1}{2}})$ can be found with the following three steps algorithm: (1) starting with $S_0 = \emptyset$, iteratively grow a set S_t according to $S_{t+1} = S_t \cup \{\underset{s\in V\setminus S_t}{\text{argmax}}\{\frac{f(S_t\cup\{s\} - f(S_t)}{w_s}, w_s + \sum_{i\in S_t} w_i \leq L\}\}$ until a final set S_T of sentences is computed; (2) obtain the set H of elements of V satisfying the capacity constraint, i.e. $H \leftarrow \{\{i\} \in V,\ w_i \leq L\}$; (3) build a solution $S^F \leftarrow \underset{S\in\{S_T\}\cup H}{\text{argmax}} f(S^F)$.

From Theorem 2, MRC problem also corresponds to the maximization of a submodular and monotonically non-decreasing function under a Knapsack constraint. It is thus a particular case of Problem (12). Moreover, Algorithm 2 corresponds to the three steps described above. Hence, the summary S^P returned by Algorithm 2 also satisfies the inequality in Eq. (11). □

4. Experiments and evaluation

Our summarizer is tested on the benchmark datasets of *Document Understanding Conferences* DUC05, DUC06 and DUC07 for query-oriented text summarization [6,7,11]. The datasets contain 50, 50 and 45 corpora, respectively. Each corpus consists of a query, about 30 news articles of 1000 words on average, and query-oriented *reference summaries*. The length of the reference summaries is set to 250 words, so we set the summary capacity L to 250.

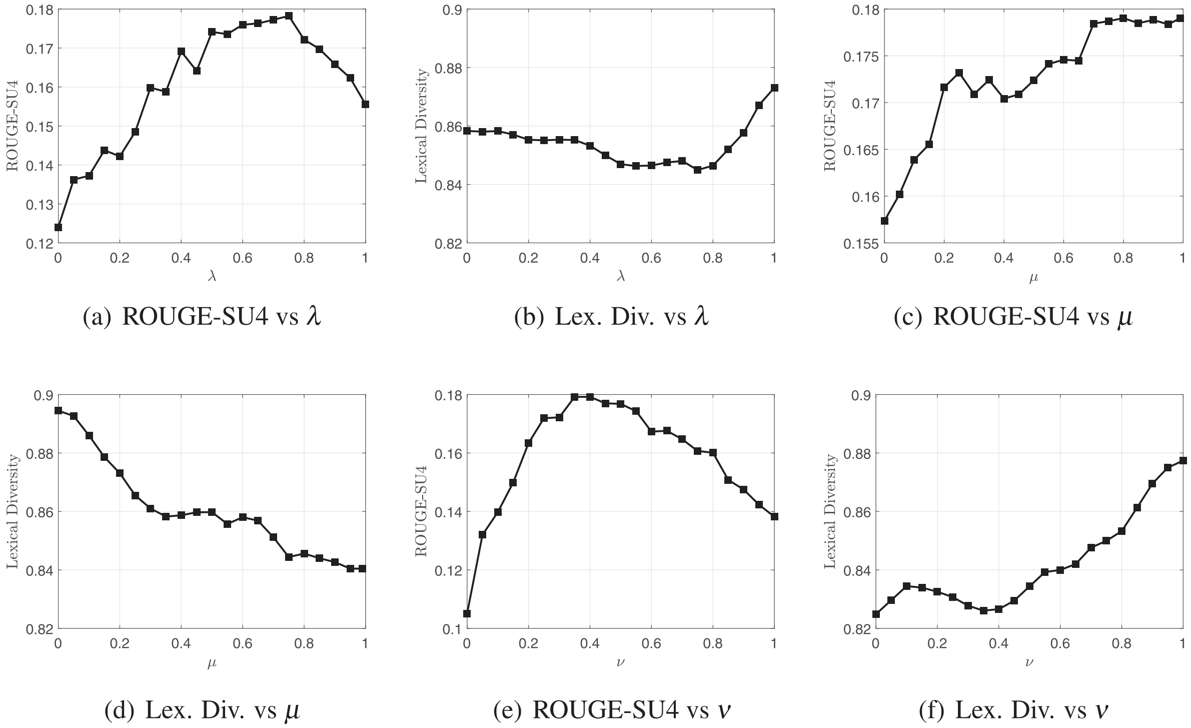
Two aspects of our automatically generated summaries are evaluated based on a comparison with reference summaries written by humans, i.e. their *content* and *diversity*. For content evaluation, we make use of ROUGE-2 [14] which measures the 2-gram overlap between a candidate and a reference summary, and ROUGE-SU4 [14] which counts both the number of common unigrams (terms) and 4-skip-bigrams, namely pairs of words that are separated by at most four words. As at DUC [6,7,11], we perform jackknife resampling, words in summaries are stemmed but stop-words are not removed. To evaluate the diversity or non-redundancy of a summary, we measure the *Normalized Entropy* of its term distribution $[p_1, \ldots, p_{N_t}]$, namely $H(p) = -\frac{1}{\log N_t}\sum_{i=1}^{N_t} p_i \log p_i$. It can be interpreted as a measure of the *Lexical Diversity* of a summary[2]

4.1. Parameter tuning

For the inference of the HDP, the Gibbs sampler of Teh et al. [26] is used. We first set parameters λ, μ and ν to 0.9, 0.99 and 0.2, respectively, and we tune the values of α, β ad γ using a leave-one-out cross-validation on a validation set consisting of 90% of corpora of DUC07 dataset. Similar to Xiong and Ji [32], we test values of γ in the range $1, \ldots, 10$, of β from 0.5 to 5 and of α from 0.25 to 2.5. The highest ROUGE-SU4 scores are achieved with $\gamma = 7.0$, $\beta = 1.5$, and $\alpha = 0.75$. Finally we choose the value of concentration parameter ζ of the symmetric Dirichlet prior to be 0.5 in accordance with what was suggested in the original version of HDP [26].

For the query balance λ, the damping factor μ and the coverage balance ν, we apply an alternating maximization strategy in which two parameters are set to a value in [0,1] and we seek the value of the third parameter that maximizes the ROUGE-SU4, which yields $\lambda = 0.75$, $\mu = 0.99$ and $\nu = 0.35$. With $\lambda = 0.75$, the score propagation is favoured compared to the query relevance, $\mu = 0.99$ is a standard damping factor for a PageRank-like algorithm [21] and $\nu = 0.35$ gives more weight to the Relevance criterion than the Topical Coverage criterion.

[2] In addition to the experiments of Section 4, a running example and an example of automatic summary are provided in the supplemental materials.



(a) ROUGE-SU4 vs λ (b) Lex. Div. vs λ (c) ROUGE-SU4 vs μ

(d) Lex. Div. vs μ (e) ROUGE-SU4 vs ν (f) Lex. Div. vs ν

Fig. 2. ROUGE-SU4 and Lexical Diversity as functions of each parameter.

For $\mu = 0.99$ and $\nu = 0.35$, Fig. 2(a) and (b) display the ROUGE-SU4 and Lexical Diversity as a function of λ. The ROUGE-SU4 reaches a peak of 0.1792 close to $\lambda = 0.75$. This shows that the fuzzy hypergraph scoring produces higher scores compared to query relevance only. Moreover, the smooth variation of ROUGE-SU4 shows that our method is not highly sensitive to λ. In Fig. 2(b), we observe that the lexical diversity does not vary significantly for $\lambda \in [0, 0.8]$ and it subsequently increases with λ. For $\lambda = 0.75$ and $\nu = 0.35$, Fig. 2(c) shows that the ROUGE-SU4 reaches a peak for μ close to 0.99. The Lexical Diversity of the summary displayed in Fig. 2(d) obviously rises when μ decreases, since a lower value of μ results in similar scores for all sentences. Finally, for $\lambda = 0.75$ and $\mu = 0.99$, Fig. 2(e) shows that the ROUGE-SU4 reaches a peak around $\nu = 0.35$. The impact of the Topical Coverage criterion is significant since $\nu = 0.35$ greatly increases the ROUGE-SU4 score over $\nu = 0$. Moreover, the smooth variation of ROUGE-SU4 around $\nu = 0.35$ confirms the low sensitivity of our method to ν. On the other hand, Fig. 2(f) shows that the Lexical Diversity of the summary grows with ν. Thus, our Topical Coverage criterion reduces the lexical redundancies compared to a selection based on relevance only.

4.2. Testing the hypergraph construction

This experiment shows the relevance of our choice of hyperedge model compared to five other ways to infer relationships between sentences. First, the *Latent Dirichlet Allocation* (LDA) [3] which, in contrast to HDP, requires the number of topics to be determined by cross-validation. Second, a term-based model in which each term defines a hyperedge connecting the sentences in which it occurs. The term frequency defines the hyperedge distribution over sentences. The weight of each hyperedge is defined as the product between the term frequency and the isf weight of the term. Finally, we test three clustering algorithms using the cosine distance between tfisf representations of sentences as a distance metric: *k-means*, *agglomerative clustering* (AC) [24], and a nonparametric version of *DBSCAN* [30]. As in [30], additional pairwise hyperedges based on the cosine similarity between tfisf representations of sentences are also included in the hypergraph. After validation, the number k of clusters for k-means is set to 70, the number of topics for LDA is set to 55, and the stopping criterion for AC is set to 0.21. Table 1 displays ROUGE-2 and ROUGE-SU4 scores and the corresponding 95% confidence intervals for all hyperedge models. We observe that our MRC algorithm outperforms the LDA-based approach by 14% of ROUGE-SU4 which confirms that the hierarchical structure of our topic model provides a more accurate model for the distribution of sentences over topics. Moreover, it also outperforms the term-based model by 5% of ROUGE-SU4. Finally our MRC algorithm outperforms the best performing clustering algorithm, DBSCAN, by 5% of ROUGE-SU4. This justifies our choice of a topic model tagging sentences with multiple topics instead of a cluster-based approach classifying each sentence in a single cluster.

Table 1
Performance of our MRC algorithm and other hyperedge models.

Hyperedge model	ROUGE-2	ROUGE-SU4
MRC	**0.12745(0.11791 – 0.13699)**	**0.1792(0.17065 – 0.18775)**
LDA	0.09336(0.081 – 0.10572)	0.15666(0.15078 – 0.16254)
TERMS	0.1131(0.10833 – 0.11786)	0.1708(0.16616 – 0.17544)
KMEANS	0.10574(0.09366 – 0.11781)	0.16831(0.16095 – 0.17567)
AC	0.09251(0.07899 – 0.10603)	0.1534(0.14236 – 0.16444)
DBSCAN	0.10636(0.09475 – 0.11797)	0.17049(0.16385 – 0.17713)

Table 2
Performance of our MRC sentence selection compared to GRR, OPH, MRMS and MCS.

Sentence selection method	ROUGE-2	ROUGE-SU4	Lexical Diversity
MRC	**0.12745(0.11791 – 0.13699)**	**0.1792(0.17065 – 0.18775)**	0.86313(0.84105 – 0.88521)
GRR	0.11858(0.10694 – 0.13021)	0.1682(0.1603 – 0.1761)	0.85114(0.81745 – 0.88482)
OPH	0.09346(0.08096 – 0.10595)	0.14857(0.14135 – 0.15579)	**0.95309(0.94411 – 0.96206)**
MRMS	0.12621(0.11438 – 0.13803)	0.16936(0.16147 – 0.17725)	0.85403(0.82349 – 0.88456)
MCS	0.10608(0.0934 – 0.11875)	0.15269(0.14337 – 0.16201)	0.93929(0.92726 – 0.95132)

4.3. Testing the relevance and coverage criterion

In this experiment, we compare our MRC-based sentence selection step to four other approaches. The *Greedy Redundancy Removal* (GRR) [32] selects sentences in descending order of scores, provided that their pairwise cosine similarity does not exceed a threshold. The *One-Per-Hyperedge* (OPH) method selects the sentence i with maximum probability ψ_{ei} for each hyperedge e considered in decreasing order of weight. The *Maximal Relevance Minimum Similarity* (MRMS) method [33] seeks a summary by simultaneously maximizing relevance and minimizing pairwise similarity. The *Maximum Corpus Similarity* (MCS) [15] produces a summary with minimum redundancy and maximum similarity with the sentence of the corpus. For MRMS and MCS, similarities are computed based on the transition probabilities over our fuzzy hypergraph $\mathrm{Sim}(i, j) = \frac{1}{2}(p(i|j) + p(j|i))$ (Eq. (2)). The values of the parameters for each method are determined by cross-validation. As shown in Table 2, in terms of ROUGE-SU4, our MRC algorithm outperforms other approaches by at least 6%. OPH (21%) yields the worst performance. This confirms that selecting one sentence only per hyperedge severely deteriorates the quality of the summary. The Lexical Diversity achieved by our MRC algorithm exceeds that of GRR and MRMS by about 1%. While our MRC method outperforms MCS by 17% of ROUGE-SU4, MCS yields a slightly higher lexical diversity since it selects lexically dissimilar sentences while our MRC algorithm focuses on Topical Coverage.

4.4. Comparison with other summarization algorithms

We compare our MRC algorithm to nine state-of-the-art summarization systems. Unless stated otherwise, lexical similarity denotes the cosine similarity between tfisf representations of sentences as defined in [2]. The early *Topic-sensitive LexRank* (TS-LexRank) [21] and *TextRank* [16] both score sentences based on PageRank-like algorithms. The summarizer based on *Hubs and Authorities* (H&A) [29] applies the HITS algorithm to simultaneously score sentences and clusters of sentences. Similarly, The *T-Graph* method [22] simultaneously scores sentences and topics discovered by LDA, using the hubs and authorities algorithm. *HyperSum* [30] defines a sentence cluster-based hypergraph and uses a semi-supervised learning method for sentence scoring. *HERF* [32] is also based on a hypergraph-based node ranking algorithm but it includes topics as hyperedges. A summarizer based on a *Continuous Vector Space Model* (CVSM) uses phrase embeddings for sentence similarity computation. *SummaRuNNer* [18] proposes a classifier based on a recurrent neural network for selecting relevant sentences. In contrast, *R2N2* [5] proposes a recursive neural network which yields relevance scored for both sentences and words. Some of the above methods produce generic summaries. In that case, the extracted sentences are further scores in terms of their query relevance.

From Tables 3 and 4, our MRC algorithm outperforms TS-LexRank, TextRank, H&A and T-Graph respectively by 2%, 7%, 5% and 3% of ROUGE-SU4 on DUC05, and by at least 8% of ROUGE-SU4 on DUC06 and DUC07, which justifies our use of a hypergraph that incorporates group relationships among sentences rather than a simple graph. HyperSum performs slightly better than MRC on DUC05 in terms of ROUGE-2. However, our method outperforms HyperSum and HERF by at least 5% of ROUGE-SU4 on DUC06 and DUC07. These two hypergraph approaches are limited to the detection of disjoint sentence clusters and do not take advantage of the fuzzy semantic relationships between sentences. They also fail to provide a proper method of sentence selection after sentence ranking. Our method also slightly outperforms CVSM, by 1%, 2% and 4% of ROUGE-SU4, respectively on DUC05, DUC06 and DUC07 datasets. Their method is however based on the costly computation of phrase embeddings and it is not meant for query-oriented summarization. Finally our method outperforms SummaRuNNer and R2N2 by approximately 5% of ROUGE-SU4 on DUC05, 9% on DUC06, and 5% on DUC07. Indeed, these methods were designed for generic summarization and they lack of a proper handling of queries.

Table 3
Comparison of our MRC algorithm with 9 methods on DUC05 and DUC06 datasets.

	DUC05		DUC06	
Method	ROUGE-2	ROUGE-SU4	ROUGE-2	ROUGE-SU4
MRC	**0.07864** (**0.07061** – **0.08667**)	0.12824 (0.11942 – 0.13706)	**0.10947** (**0.10017** – **0.11877**)	**0.16141** (**0.15334** – **0.16948**)
TS-LEXRANK	0.07231 (0.06510 – 0.07952)	0.12554 (0.11777 – 0.13331)	0.08892(0.08072 – 0.09712)	0.14741 (0.13833 – 0.15649)
TEXTRANK	0.06302 (0.05443 – 0.07162)	0.11970 (0.11087 – 0.12852)	0.07813 (0.07070 – 0.08557)	0.14056 (0.13214 – 0.14897)
H&A	0.06902 (0.06140 – 0.07664)	0.12217 (0.11268 – 0.13166)	0.08172 (0.07220 – 0.09124)	0.13731 (0.13001 – 0.14461)
T-GRAPH	0.06906 (0.06079 – 0.07733)	0.12456 (0.11585 – 0.13327)	0.09226 (0.08556 – 0.09895)	0.14914 (0.14165 – 0.15664)
HYPERSUM	0.07291 (0.06424 – 0.08086)	**0.13087** (**0.12226** – **0.13872**)	0.09569 (0.08722 – 0.10404)	0.15182 (0.14424 – 0.15899)
HERF	0.06212 (0.05250 – 0.07174)	0.12244 (0.11397 – 0.13091)	0.07226 (0.06355 – 0.08097)	0.15346 (0.14520 – 0.16172)
CVSM	0.07426 (0.06868 – 0.07984)	0.12751 (0.12074 – 0.13429)	0.10136 (0.09527 – 0.10746)	0.15790 (0.14996 – 0.16583)
SummaRuNNer	0.07117 (0.06627 – 0.07608)	0.12169 (0.11440 – 0.12898)	0.09652 (0.09321 – 0.09984)	0.14880 (0.14252 – 0.15508)
R2N2	0.06607 (0.05994 – 0.07219)	0.12273 (0.11560 – 0.12985)	0.09286 (0.08607 – 0.09965)	0.14656 (0.13759 – 0.15552)

Table 4
Comparison of our MRC algorithm with 9 methods on DUC07 dataset.

	DUC07	
Algorithm	ROUGE-2	ROUGE-SU4
MRC	**0.12745** (**0.11791** – **0.13699**)	**0.17920** (**0.17065** – **0.18775**)
TS-LEXRANK	0.11048 (0.10061 – 0.12035)	0.16524 (0.15542 – 0.17506)
TEXTRANK	0.10443 (0.09580 – 0.11305)	0.15865 (0.14941 – 0.16790)
H&A	0.10493 (0.09633 – 0.11353)	0.15756 (0.14938 – 0.16574)
T-GRAPH	0.10953 (0.10185 – 0.11721)	0.15870 (0.15030 – 0.16711)
HYPERSUM	0.11197 (0.10331 – 0.12063)	0.16612 (0.15659 – 0.17565)
HERF	0.11234 (0.10485 – 0.11983)	0.16330 (0.15343 – 0.17317)
CVSM	0.12217 (0.11378 – 0.13056)	0.17276 (0.16284 – 0.18269)
SummaRuNNer	0.11570 (0.11158 – 0.11982)	0.17096 (0.16488 – 0.17704)
R2N2	0.11265 (0.10395 – 0.12135)	0.16601 (0.15542 – 0.17660)

To further illustrate the above quantitative results, we performed a qualitative comparison of the methods with the help of two human judges. A sample of ten summaries was presented to the readers who were asked to perform a pairwise comparison of systems in terms of four criteria: *informativeness* (the amount of information contained in the summary), *non-redundancy, query relevance*, and *intelligibility* (the coherence of the summary and the absence of unresolved references). For each pair M_1, M_2 of methods and for each criterion, the reader was asked whether M_1 performed significantly better, equally or worse than M_2. The results show that our MRC system along with CVSM, SummaRuNNer and R2N2 perform equivalently in terms of informativeness, and they outperform all other methods. As for other methods, HyperSum, HERF and T-Graph were found to perform better than TS-LexRank, TextRank, H&A in terms of informativeness. In terms of non-redundancy, MRC, TS-LexRank, HyperSum, CVSM, SummaRuNNer and R2N2 perform equivalently since all these methods attempt to reduce redundancies in some way, while HERF produced slightly less redundant summaries. Finally, H&A and TextRank were found to produce more redundant summaries than other summarizers in general. In terms of query relevance, MRC, HERF and HyperSum outperformed all other summarizers. H&A, SummaRuNNer, R2N2 and CVSM produced the worst summaries in terms of query relevance since these methods were not initially meant for query-oriented summarization. Finally, all systems were found to perform equally in terms of intelligibility of the summaries. Indeed, in the absence of co-reference resolution, all summaries contained a few unresolved references. We conclude that the main strengths of our MRC system are the production of informative and query-relevant summaries. Among its weaknesses, we observed that the intelligibility of the summaries could be improved, possibly by resolving co-references in the original corpus as a preprocessing step. A minor drawback of our system is the production of slightly redundant summaries.

4.5. Comparison with DUC systems

For each DUC dataset, Table 5 reports the ROUGE-2 and ROUGE-SU4 scores for the top four competing systems, for the worst human summarizer (*Hum*), for the baseline chosen by NIST, and for the average performance of all systems. The top systems are *S*15, *S*17, *S*10 and *S*8 for DUC05, *S*24, *S*15, *S*12 and *S*8 for DUC06, *S*15, *S*29, *S*4 and *S*24 for DUC07. The confidence intervals are not given as they are not officially released by DUC. Apart from DUC05, we observe that our proposed method slightly outperforms other summarizers in terms of ROUGE-2 and ROUGE-SU4 but it performs worse than the human summaries which was expected since we merely produce extractive summaries which cannot match human summaries. Overall, we observe that our system achieves better performances on DUC06 and DUC07 than it does on DUC05 dataset.

Table 5
Comparison with DUC05, DUC06 and DUC07 systems.

	DUC05		DUC06		DUC07	
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
MRC	**0.07864**	0.12824	**0.10947**	**0.16141**	**0.12745**	**0.1792**
1st	0.07251	**0.13163**	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

5. Conclusion

In this paper, we proposed a novel query-oriented summarization approach in which semantic relationships between sentences are captured by a probabilistic topic model. The resulting topics are represented as hyperedges of a fuzzy hypergraph in which nodes are sentences. Sentences are then scored based on their relevance to the query and their centrality in the hypergraph. Then, a set of sentences is selected by maximizing the individual Relevance scores and the joint Topical Coverage, which is formulated as a fuzzy extension of the dominating set problem. Experiments show that both our topic-based fuzzy hypergraph model and our sentence selection algorithm contribute to an improvement in the content coverage of the summaries, when compared to other graph-based summarizers and summarizers presented at DUC contests. As a future research direction, we will investigate how to better address the issue of intelligibility of the summaries produced by our method as pointed out in Section 4.4 (e.g. with co-reference resolution). We will also investigate how to adapt the model for related tasks including update summarization and community question answering. We will also attempt to incorporate sentence fusion and compression in our fuzzy hypergraph-based method to determine whether topical relationships can help in these tasks.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Supplementary material

Supplementary material associated with this article can be found, in the online version, at doi:10.1016/j.ins.2019.05.020.

References

[1] R. Arora, B. Ravindran, Latent Dirichlet allocation based multi-document summarization, in: Proc. of AND 2008, ACM, Singapore, 2008, pp. 91–97.
[2] C. Blake, A comparison of document, sentence, and term event spaces, in: Proc. of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics, ACL, Sydney, Australia, 2006, pp. 601–608.
[3] D.M. Blei, A.Y. Ng, M.I. Jordan, Latent Dirichlet allocation, J. Mach. Learn. Res. 3 (2003) 993–1022.
[4] X. Cai, W. Li, Ranking through clustering: an integrated approach to multi-document summarization, IEEE Trans. Audio Speech. Lang. Process. 21 (7) (2013) 1424–1433.
[5] Z. Cao, F. Wei, L. Dong, S. Li, M. Zhou, Ranking with recursive neural networks and its application to multi-document summarization, in: Proc. of AAAI 2015, AAAI Press, Austin, TX, 2015, pp. 2153–2159.
[6] H.T. Dang, Overview or DUC 2005, in: Proc. of the Document Understanding Conference, NIST, Vancouver, Canada, 2005, pp. 1–12.
[7] H.T. Dang, Overview of the DUC 2007 summarization task, in: Proc. of the Document Understanding Conference, NIST, Rochester, NY, 2007.
[8] G. Erkan, D. Radev, Lexrank: graph-based centrality as salience in text summarization, J. Artif. Intell. Res. 22 (1) (2004) 457–479.
[9] M. Garey, D.S. Johnson, Computers and Intractability, vol. 29, W. H. Freeman & Co, New York, NY, 2002.
[10] L. Hennig, D.A.I. Labor, Topic-based multi-document summarization with probabilistic latent semantic analysis, in: Proc. of RANLP 2009, ACL, Borovets, Bulgaria, 2009, pp. 144–149.
[11] T.D. Hoa, Overview of DUC 2006, in: Proc. of the Document Understanding Conference, NIST, New York, NY, 2006.
[12] M. Kageback, O. Mogren, N. Tahmasebi, D. Dubhashi, Extractive summarization using continuous vector space models, in: Proc. of the 2nd Workshop on Continuous Vector Space Models and their Compositionality (CVSC) 2014, ACL, Gothenburg, Sweden, 2014, pp. 31–39.
[13] A. Krause, D. Golovin, Submodular function maximization, Tractability: Practical Approaches to Hard Problems, Cambridge University Press, Cambridge, UK, 2014.
[14] C.Y. Lin, E.H. Hovy, Automatic evaluation of summaries using n-gram co-occurrence statistics, in: Proc. of HLT-NAACL 2003, ACL, Edmonton, Canada, 2003, pp. 71–78.
[15] H. Lin, J. Bilmes, Multi-document summarization via budgeted maximization of submodular functions, in: Proc. of HLT-NAACL 2010, ACL, Los Angeles, CA, 2010, pp. 912–920.
[16] R. Mihalcea, P. Tarau, Textrank: Bringing order into text, in: Proc. of EMNLP 2004, ACL, Barcelona, Spain, 2004.
[17] J.N. Mordeson, P.S. Nair, Fuzzy hypergraphs, fuzzy graphs and fuzzy hypergraphs, Physica, 2000, pp. 135–231. Heidelberg, Germany
[18] R. Nallapati, F. Zhai, B. Zhou, SummaruNNer: a recurrent neural network based sequence model for extractive summarization of documents, in: Proc. of AAAI 2017, 2017, pp. 3075–3081.
[19] S. Narayan, S.B. Cohen, M. Lapata, Ranking sentences for extractive summarization with reinforcement learning, Proc. of HLT-NAACL 2018, ACL, New Orleans, LA, 2018, pp. 1747–1759.

[20] A. Nenkova, K. McKeown, Automatic summarization, Found. Trends Inf. Retriev. 5 (2–3) (2011) 103–233.

[21] J. Otterbacher, G. Erkan, D. Radev, Using random walks for question-focused sentence retrieval, in: Proc. of HLT/EMNLP 2005, ACL, Vancouver, Canada, 2005, pp. 915–922.

[22] D. Parveen, H.M. Ramsl, M. Strube, Topical coherence for graph-based extractive summarization, in: Proc. of EMNLP 2015, ACL, Lisbon, Portugal, 2015, pp. 1949–1954.

[23] F. Pedregosa, et al., Scikit-learn: machine learning in python, J. Mach. Learn. Res. 12 (2011) 2825–2830.

[24] L. Rokach, O. Maimon, Clustering Methods, Data Mining and Knowledge Discovery Handbook, Springer, New York, NY, 2005, pp. 321–352.

[25] C. Shen, T. Li, Multi-document summarization via the minimum dominating set, in: Proc. of COLING 2010, ACL, Beijing, China, 2010, pp. 984–992.

[26] Y.W. Teh, M.I. Jordan, M.J. Beal, D.M. Blei, Sharing clusters among related groups: hierarchical Dirichlet processes, in: Advances in Neural Information Processing Systems, MIT Press, Vancouver, Canada, 2005, pp. 1385–1392.

[27] H.V. Lierde, T.W.S. Chow, Incorporating word embeddings in the hierarchical Dirichlet process for query-oriented text summarization, in: Proc. of INDIN 2017, IEEE, Emden, Germany, 2017, pp. 1037–1042.

[28] H.V. Lierde, T.W.S. Chow, Query-oriented text summarization based on hypergraph transversals, Inf. Process. Manag. 56 (4) (2019) 1317–1338.

[29] X. Wan, J. Yang, Multi-document summarization using cluster-based link analysis, in: Proc. of SIGIR 2008, ACM, Singapore, Singapore, 2008, pp. 299–306.

[30] W. Wang, S. Li, J. Li, W. Li, F. Wei, Exploring hypergraph-based semi-supervised ranking for query-oriented summarization, Inf. Sci. 237 (2013) 271–286.

[31] F. Wei, W. Li, Q. Lu, Y. He, A document-sensitive graph model for multi-document summarization, Knowl. Inf. Syst. 22 (2) (2010) 245–259.

[32] S. Xiong, D. Ji, Query-focused multi-document summarization using hypergraph-based ranking, Inf. Process. Manag. 52 (4) (2016) 670–681.

[33] W. Yin, Y. Pei, Optimizing sentence modeling and selection for document summarization, in: Proc. of IJCAI 2015, AAAI Press, Buenos Aires, Argentina, 2015, pp. 1383–1389.

[34] Z. Zhang, S.S. Ge, H. He, Mutual-reinforcement document summarization using embedded graph based sentence clustering for storytelling, Inf. Process. Manag. 48 (4) (2012) 767–778.

Hadrien Van Lierde received the B.Sc. and M. Eng. degree in applied mathematics from Universite catholique de Louvain, Louvain-la-Neuve, Belgium, in 2015. He is currently working at the City University of Hong Kong, Hong Kong, toward the Ph.D. degree in electronic engineering. His research interests include machine learning, text mining, automatic text summarization and complex networks.

Tommy W. S. Chow received the B.Sc. (Hons.) and Ph.D. degrees from the Department of Electrical and Electronic Engineering, University of Sunderland, Sunderland, U.K., He is currently a professor with the Department of Electronic Engineering, City University of Hong Kong, Hong Kong. He has authored or co-authored over 200 technical articles related to his research, five book chapters, and one book. His current research interests include neural networks, machine learning, pattern recognition.