J Comb Optim (2015) 30:996–1015
DOI 10.1007/s10878-015-9855-0

Optimizing word set coverage for multi-event summarization

Jihong Yan[1,2] · Wenliang Cheng[1] · Chengyu Wang[1] · Jun Liu[3] · Ming Gao[1] · Aoying Zhou[4]

Published online: 14 March 2015

Abstract We have witnessed the proliferation of the Internet over the past few decades. A large amount of textual information is generated on the Web. It is impossible to locate and digest all the latest updates available on the Web for individuals. Text summarization would provide an efficient way to generate short, concise abstracts from the massive documents. These massive documents involve many events which are hard to be identified by the summarization procedure directly. We propose a novel methodology that identifies events from these text corpora and creates summarization for each event. We employ a probabilistic, topic model to learn the potential topics from the massive documents and further discover events in terms of the topic distributions of documents. To target the summarization, we define the word set coverage problem (WSCP) to capture the most representative sentences to summarize an event. For getting solution of the WSCP, we propose an approximate algorithm to solve the optimization problem. We conduct a set of experiments to evaluate our proposed approach on two real datasets: Sina news and Johnson & Johnson medical news. On both datasets, our proposed method outperforms competitive baselines by considering the harmonic mean of coverage and conciseness.

Keywords Event summarization · Word set · Set coverage · Optimization

✉ Ming Gao
mgao@sei.ecnu.edu.cn

1 Institute for Data Science and Engineering, East China Normal University, Shanghai 200062, China

2 Economic Management Institute, Shanghai Second Polytechnic University, Shanghai 201209, China

3 Shanghai General Hospital, Shanghai Jiaotong University, Shanghai 200080, China

4 Shanghai Key Lab for Trustworthy Computing, East China Normal University, Shanghai 200062, China

1 Introduction

Due to the proliferation of the Web, a large amount of textual information that appears in all aspects of our daily lives is generated on the Internet. Unfortunately, many people spend a lot of time on both reading much useless information and finding their interested information from the Web.

For example, the Web provides a platform to share information for health care applications. A large number of Web users contribute massive Web pages related to health care daily. It is impossible for individuals to read and digest all information. Text summarization is an efficient way to create short, concise abstracts from the massive documents. Therefore, it could be helpful to share and utilize the health care information.

In the past few decades, text summarization has been intensively studied in the NLP (Natural Language Processing) community. It takes single document as input then generates an abstract to summarize the document. Another type of summarization extract of a single summary from multiple documents. Detailed surveys have been provided by Gupta and Lehal (2010) and Das and Martins (2007). In practice, we are faced with the following more complicated problems.

1. There may be a large number of events expressed in the collection of Web sites. For health care, there are several important events reflected in articles related to different infectious diseases.
2. An event may be expressed in many times in different articles. These articles share similar topics, but they may be presented the different aspects or related to different phases of the entire event.

In this paper, we consider the problem of summarizing multiple events from a text corpus. Unlike existing work, we consider multiple events that need to be summarized, each of which is related to one or more latent topics in the corpus. The task is to find a set of representative sentences in the corpus that describes each event best.

To tackle the problem, we present an algorithmic approach in the paper. Our approach first models the latent topic distributions of the documents using latent Dirichlet allocation (LDA) by Blei et al. (2003), and then clusters the topics to discover these events in the corpus. Specifically, we represent each event as a set of keywords (i.e., word set). To perform summarization, we formally define the word set coverage problem (WSCP) and propose an approach to approximately optimize the problem.

To demonstrate the efficacy of the proposed method, we conduct experiments on corpora of medical news articles. We compare our proposed approach with some reasonable baselines based on some performance measures and present in-depth experimental analysis and case study.

In summary, the main contributions of this paper are threefold:

1. We design an algorithm that can discover events to be summarized without human intervention.
2. To perform summarization, we present the WSCP and solve it by optimization.

3. We conduct extensive experiments to showcase the efficacy and effectiveness of our method.

The rest of the paper is organized as follows: we review the related work in Sect. 2. Section 3 describes the representation of multiple events and generation algorithm. In Sect. 4, we present the WSCP to summarize events. Section 5 discusses the experimental results. Finally, we conclude and discuss the future work in Sect. 6.

2 Related work

To the best of our knowledge, the formulation of multi-event summarization is novel. There is however some related work that we discuss below. Depend on the fields they cover, the related work can be further classified into the following categories:

2.1 Text summarization

Text summarization is the task of producing short texts from single or multiple documents that preserve important information, which is defined by Radev et al. (2002).

With the advent of machine learning, a number of statistical techniques have been employed in summarization. Kupiec et al. (1995) proposed a Naive Bayes classifier which categorizes each sentence as worthy of extraction or not. In the work by Lin (1999), instead of Naive Bayes, they defined rich features and trained Decision Tree classifiers for sentence selection. Furthermore, Chieu and Ng (2002) employed log-linear models and showed empirically the method outperformed previous models. Also, Svore et al. (2007) used neural networks and third party features to perform extractive summarization. Fattah and Ren (2008) studied mathematical regression to estimate text feature weights for summarization. In contrast with classification techniques, Conroy and O'leary (2001) used a sequential model, namely Hidden Markov Model, to learn the hidden state of sentences for summarization.

Besides machine learning, other methods have been studied, too. García-Hernández and Ledeneva (2009) used a direct adaptation of term frequency-inverse document frequency (TF–IDF) method to summarization. Kruengkrai and Jaruskulchai (2003) tackled the problem from a graph theoretic approach. Sentences in the documents are represented as nodes in an undirected graph for further selection. Kyoomarsi et al. (2008) performed automatic text summarization by using a fuzzy logic system, which considered fuzzy IF-THEN rules based on text characteristics. Takamura and Okumura (2009) represented text summarization as a maximum coverage problem (MCP). It solved the problem by a greedy algorithm, a randomized algorithm and a branch-and-bound method.

The existing work has present a deep study on text summarization. However, in our work, our goal is not to simply generate abstracts for articles, but to discover and summarize the "hidden"events inside the corpus.

2.2 Event summarization

Event summarization is the process of discovering event patterns to represent the original event sets. It provides a brief and accurate summary for event datasets and gives insightful views about the entire system.

Some research efforts have been working on providing summarization methods on social media datasets. Sakaki et al. (2010) showed that by mining tweets on Twitter, events with large social impacts, such as earthquakes, could be detected. Besides Twitter, Becker et al. (2010) identified social events on other social media, including Flickr and Youtube. However, the detection of events cannot reveal the structure and development process. Chakrabarti and Punera (2011) studied long-running, structure-rich events in Twitter by learning the hidden state of the events via Hidden Markov Models. Recently, Tsolmon and Lee (2014) combined timeline analysis and user behavior analysis to extract social events by adapted LDA

While these methods provide a careful analysis and summarize social events on social medium, our work is different in context. Texts in social media are short, concise, generated by Internet users and often with tags. In contrast, articles in the corpus are long and in formal language. Our approach takes these characteristics into account and treat events as collections of keywords.

2.3 Set cover problem

The set cover problem (SCP) is a typical NP-hard combinatorial problem, studied in combinational optimization. It has been employed in solving a large range of problems, including scheduling, planning, information retrieval, etc (Caragiannis et al. 2013; Yaghini et al. 2013; Deng and Lin 2011).

Currently, many algorithms have been proposed to solve the SCP. Generally, these approaches can be further divided into two categories: exact methods and approximation methods. Most exact methods are based on either branch-and-bound approaches or branch-and-cut approaches. (See Balas 1996; Avella et al. 2009). However, solving NP-hard problems via exact methods are usually time consuming and computationally expensive. Thus, the approximate methods are vital to solve the SCP (Fisher 1988). In the literature, Caprara et al. (1999) introduced a Lagrangian-based heuristic method for SCP. Ablanedo-Rosas (2010) proposed a set of constraint normalization rules to the SCP. A detailed survey on both exact and approximate algorithms can be found in Umetani and Yagiura (2007).

3 Multi-event representation and generation

In this section, we discuss how events are represented in the corpus and its generation algorithm in detail. Key notations and their meanings can be found in Table 1.

Table 1 Key notations

Notation	Description
T	The number of topics specified as a parameter
K	The number of events specified as a parameter
$D = \{d\}$	The set of all documents in the corpus
$S_i = \{s\}$	The set of sentences in the ith document
$W_{i,j} = \{w\}$	The set of unique words in the ith sentence of the jth document
$E_i = \{e_1, e_2, \ldots, e_{N_i}\}$	The set of keywords in the ith event
$\theta_i = \{\theta_i^{(1)}, \theta_i^{(2)}, \ldots, \theta_i^{(T)}\}$	The document-topic distributions for the ith documents
$D_i = \{d\}$	The set of documents related to the ith event
$S_{D_i} = \{s\}$	The set of sentences in all D_i

3.1 Word set representation

We first model our text corpus in formal. A corpus D consists of a collection of documents. Each document has a collection of sentences S_i. The ith sentence in the jth document has a collection of words $W_{i,j}$.

Unlike existing work where there is only one event in the corpus that needs to be summarized, we assume there can be K events expressed in the corpus. Each event (such as earthquake, birth, etc.) describes several latent topics in the corpus that cannot be directly observed.

To model these events concisely, like a sentence, an event E_i can be also expressed by a collection of N_i keywords. (N_i can be varied in different events.) These keywords form a word set for event E_i. For example, an earthquake can be described by "shake", "earthquake", "quake", "hurt", etc.

3.2 Word set generation

3.2.1 Topic inferring

The topic inference for each document is a crucial task in our approach since the inference forms a bridge to connect corpus and events. In this paper, we employ the probabilistic topic modeling to inference the topics for each document. Actually, there are some techniques that can also build the connection, such as the Vector Space model (VSM) for text. There are two reasons to employ the topic modeling to do that. The first reason is that the other techniques suffer from the dimensionality curse problem (see Friedman 1997) when we extract events using clustering methods, such as K-Means (by Hartigan and Wong 1979) and DBSCAN (by Ester et al. 1996). The second one is due to that it is hard to combine the heterogeneous information from a document, such as text, URLs, images, etc.

In the paper, we employ LDA by Blei et al. (2003) to generate document-topic distributions for each document. LDA is a generative, probabilistic machine learning technique to model the latent topic distribution of documents. Let θ_i be the document-topic distribution vector of the ith document. We use θ_i as the reduced vector representation of the original document.

3.2.2 Dcument clustering

To perform clustering on documents, we employ a distance measurement between two probability distributions, namely Jensen–Shannon divergence (JSD) by Lin (1991), which is defined as:

$$JSD(\theta_i, \theta_j) = \frac{1}{2} KLD\left(\theta_i || \hat{\theta}\right) + \frac{1}{2} KLD\left(\theta_j || \hat{\theta}\right) \tag{1}$$

where $\hat{\theta}$ is the average distribution of θ_i and θ_j and KLD is the Kullback-Leibler Divergence for discrete distributions, which are defined as:

$$\hat{\theta}^{(k)} = \frac{\theta_i^{(k)} + \theta_j^{(k)}}{2} \tag{2}$$

$$KLD(\theta_i || \theta_j) = \sum_{k=1}^{T} \theta_i^{(k)} log \frac{\theta_i^{(k)}}{\theta_j^{(k)}} \tag{3}$$

After calculating the JSD between different documents, we obtain the pairwise document distance matrix $M = [m_{i,j}]_{N \times N}$ where $m_{i,j} = JSD(\theta_i, \theta_j)$. The event discovery process can be performed by using K-means clustering directly on the matrix, instead of the original words. Each cluster in the result contains a list of documents that share similar latent topics. The documents of the ith cluster, denoted as D_i, are used to generate word set for the ith event.

3.2.3 Word set generation

To generate word set for each cluster (i.e., event), we employ the TF-IDF method to calculate the weights. The TF-IDF method can determine the keywords in a document. The weight for each word is calculated by multiplying TF and IDF. The detailed algorithm can be found in Salton and McGill (1984). In the ith cluster, there are $|D_i|$ documents. For the jth document $d_{i,j}$, let $W_{d_{i,j}}$ denote the top-m% words with highest TF-IDF weights. (The default value of m is 8 in the experiment.) The word set w.r.t. the ith event E_i is defined as the union set of $W_{d_{i,j}}$ for all j:

$$E_i = \bigcup_{j \in \{1 \ldots |D_i|\}} W_{d_{i,j}} \tag{4}$$

The word set generation algorithm is shown in Algorithm 1. In Line 2, we train the LDA model. We then compute the JSD between different documents in Line 9.

The clustering process can be found in Line 13. We finally generate word set for each event in Line 21.

Algorithm 1 Word set generation algorithm

Input:
The text corpus, D;
The number of events, K;
The number of topics, T;
The percentage of words selected as keywords in a document, $m\%$;
Output:
Word sets for all events, E;
1: Initialize $E = \{\}$;
2: Train LDA model on text corpus D with T topics;
3: **for** each document d_i in D **do**
4: Generate document-topic distribution θ_i;
5: **end for**
6: **for** each document d_i in D **do**
7: **for** each document d_j in D **do**
8: **if** $JSD(\theta_j, \theta_i)$ is unknown **then**
9: Compute $JSD(\theta_i, \theta_j)$;
10: **end if**
11: **end for**
12: **end for**
13: Train K-Means model on $[JSD(\theta_i, \theta_j)]$ with K clusters;
14: **for** each $i \in [1, K]$ **do**
15: **for** each document d_j in D_i **do**
16: **for** each word w_k in document d_j **do**
17: Compute TF-IDF $weight(w_k) = tf(w_k, d_j) \times idf(d_j)$;
18: **end for**
19: Sort all words $\{w_k\}$ in descending order by weight;
20: Select top-$m\%$ words $W_{d_{i,j}}$ as keywords for document d_j in D_i;
21: Add $W_{d_{i,j}}$ to E_i: $E_i = E_i \cup W_{d_{i,j}}$;
22: **end for**
23: Add word set E_i to E: $E = E \cup E_i$;
24: **end for**
25: **return** E;

4 Multi-event summarization

In this section, we formally model the multi-event summarization problem as a WSCP which can be solved by approximate optimization.

In our setting, we consider the problem of multi-event summarization. In the word set generation algorithm, documents in the corpus are clustered into K groups, which are related to K events. Recall that K-Means clustering has the completeness and exclusiveness properties. That is, each document can be assigned to one and only one cluster. (See Hartigan and Wong 1979). Thus, given the clustering result, the summarizations of different events are independent with each other. The problem of multi-event summarization can be reduced to K separate single-event summarization problems.

4.1 Coverage and conciseness

Given a collection of keywords E_i and a collection of sentences S_{D_i} in D_i, the goal of summarization is to find a subset of sentences $S'_{D_i} \subset S_{D_i}$ that is short and concise summary of the event.

To characterize the summarization power of S'_{D_i}, similar to the work by Alguliev et al. (2011), we define the measurements: coverage and conciseness. Coverage is a measure to show the power of the summary to cover as much information as possible in the original corpus. However, coverage is not enough to measure the effectiveness of summary. Simply increasing the coverage will result in longer summary and worsen the conciseness. Conciseness is a measure to indicate that a summary should not be too long and contain too much redundant information. We define the two measures in detail.

4.1.1 Coverage

Firstly, let E'_i denote the set of keywords appeared in at least one sentence in S'_{D_i}. Intuitively, if more keywords in E_i are covered in S'_{D_i}, the more information is preserved in the summary of the original event. Given a sentence $s = \{w\}$ (the set of words in a sentence) in S'_{D_i}, let $E'_i(s)$ be the set of keywords appeared in s:

$$E'_i(s) = s \cap E_i \tag{5}$$

Then, for the collection of sentences S'_{D_i}, we can compute E'_i by calculating the union set of $E'_i(s)$ for all $s \in S'_{D_i}$:

$$E'_i = \bigcup_{s \in S'_{D_i}} E'_i(s) \tag{6}$$

Finally, the coverage of S'_{D_i} is the fraction of keywords covered by E'_i, that is:

$$Cov(S'_{D_i}) = \frac{|E'_i|}{|E_i|} \tag{7}$$

where $|E_i|$ can be treated as a normalization factor, which is useful for comparing the coverage between different events, where the size of keywords of different events may vary.

4.1.2 Conciseness

We define the conciseness of the sentence s to be the fraction of words that matches one keyword in E_i, that is:

$$Con(s) = \frac{|s \cap E_i|}{|s|} \tag{8}$$

Then, the conciseness of $S^{'}_{D_i}$ is the minimum conciseness of any sentence in the set, defined as:

$$Con(S^{'}_{D_i}) = \min_{s \in S^{'}_{D_i}} Con(s) \tag{9}$$

Note that the definition of conciseness of $S^{'}_{D_i}$ is to distinguish the long and short sentences that contain the same number of keywords. It is reasonable to select a short sentence for summarization, rather than a long sentence, if they have the same number of keywords. It is closely related to the solution of WSCP. We will explain it in the next section. Conciseness can impose a strong constraint on the summary. For example, if we require that the conciseness of a summary is above a pre-defined threshold, we can control the length of the summary.

4.2 Word set coverage problem

In this section, we formally define the WSCP.

Recall that for a subset of sentences $S^{'}_{D_i}$, we have proposed two measurements of effectiveness, namely coverage and conciseness. However, the two measurements cannot be optimized at the same time. If coverage is maximized, the summary will be long, which will consequently worsen conciseness. On the other hand, if conciseness is maximized, the algorithm favors short summaries and lower the coverage. As a result, there is no overall optimal solution.

In this paper, we transform it as a coverage maximization problem by constraining the conciseness. Formally we define the WSCP as follows:

Given a set of keywords E_i, a collection of sentences S_{D_i} and a predefined parameter α, find a subset of M sentences $S^{'}_{D_i}$ from S_{D_i} such that the coverage $Cov(S^{'}_{D_i})$ is maximized, while the conciseness $Con(S^{'}_{D_i})$ is at least α. That is:

$$\begin{gathered} \text{maximize } Cov(S^{'}_{D_i}) \\ \text{subject to } Con(S^{'}_{D_i}) \geq \alpha \\ |S'_{D_i}| = M \end{gathered}$$

Obviously, the WSCP is NP-hard. Here is a proof: consider a special case of WSCP where $\alpha = 0$. Our goal is simply to select M sentences that maximize the coverage. Recall that S_{D_i} and $S^{'}_{D_i}$ can both be represented as collections of keywords, E_i and $E^{'}_i$, respectively. Then, the problem can be viewed as selecting a subset of elements that cover the entire set best. Thus, the complexity of WSCP with $\alpha = 0$ is equivalent to that of the maximum coverage problem (MCP). Since MCP is NP-hard, our WSCP is NP-hard, too.

Note that we can select small value of parameter α such that the optimization problem exactly has at least one solution. Because the conciseness of $S^{'}_{D_i}$ is defined as the minimum conciseness of all sentences in the set. The conciseness constraint can be satisfied by first scanning $S^{'}_{D_i}$ to filter out sentence whose conciseness is less than α. We will show approximate algorithm in the next section.

4.3 Optimization algorithm

In this section, we present a greedy, approximate algorithm for the WSCP shown in Algorithm 2.

Algorithm 2 WSCP Optimization Algorithm

Input:
The set of keywords, E_i;
The collection of sentences, S_{D_i};
The number of sentences in summary, M;
The constraint parameter, α;
The cost parameter, β;
Output:
The collection of sentences as summary, S'_{D_i};

1: Initialize $S'_{D_i} = \emptyset$;
2: Select sentences that satisfy constraint $\Phi = \{s \in S_{D_i} | Con(s) \geq \alpha\}$;
3: **while** $|S'_{D_i}| < M$ **do**
4: **for all** sentence $s \in \Phi$ **do**
5: Compute $Gain(s) = Cov(s \cup S'_{D_i}) - Cov(S'_{D_i})$;
6: Compute $Cost(s) = (1 - Con(s))\beta + (1 - \beta)$;
7: **end for**
8: **if** $\Phi = \emptyset$ **then**
9: break;
10: **end if**
11: Select the most suitable sentence $s^\star = \arg\max_{s \in \Phi} Gain(s)/Cost(s)$;
12: Remove $s^\star$ from Φ: $\Phi = \Phi \backslash s^\star$;
13: Add $s^\star$ to summary: $S'_{D_i} = S'_{D_i} \cup s^\star$;
14: **end while**
15: **return** S'_{D_i};

The algorithm performs in iterations. At first, it chooses sentences that satisfy the conciseness constraint Φ in Line 2. At each iteration, it adds one sentence to the output summary S'_{D_i}. For each sentence s in Φ, we compute two quantities: gain and cost. The gain is the increase in coverage when a new sentence s is added to S'_{D_i} as in Line 5. That is:

$$Gain(s) = Cov\left(s \cup S'_{D_i}\right) - Cov\left(S'_{D_i}\right) \tag{10}$$

The cost is proportional to the lost in conciseness. Recall that $Con(s) \in [0, 1]$, so the lost in conciseness can be defined as $1 - Con(s)$. To control the effect of conciseness when selecting a sentence, we here introduce a cost parameter $\beta \in [0, 1]$. Then the cost of s is:

$$Cost(s) = (1 - Con(s))\beta + (1 - \beta) \tag{11}$$

In the equation, we can tune the parameter β to determine the effect of conciseness. When $\beta = 0$, the lost in conciseness has no power in the sentence selection process. Values of β between 0 and 1 regulate the effect of conciseness. When $\beta > 0$, the

Table 2 Dataset statistics

Company name	#Doc	#Sentence	Avg. #sentence per doc
Johnson & Johnson	417	9577	22.97
GlaxoSmithKline	204	5260	25.78
Abbott Laboratories	258	6203	24.04
Pitzer	132	2821	21.37
Wyeth	143	3846	26.89
Tong Ren Tang	217	4766	21.96
Yun Nan Bai Yao	254	6131	24.14

larger β is, the more coverage the algorithm needs to gain for a sentence with a low conciseness value. See Line 6 in the algorithm.

After the quantity calculation process, we select a sentence $s^\star$ with the highest gain-to-cost ratio, which is $Gain(s)/Cost(s)$, as Line 11 shows. Also, to ensure $|S'_{D_i}| = M$ is satisfied, the algorithm stops when there are already there are M sentences in the summary in Line 3.

To conclude, although the algorithm is greedy and approximate, we can solve the problem in a heuristic way. Thus, our proposed algorithm can efficiently solve the WSCP.

5 Experimental analysis

The goal of our experiments is to showcase the efficacy of the proposed method in generating event summaries from text corpus.

5.1 Dataset

To evaluate the efficacy of our proposed approach, we crawl the news articles from Sina.com[1], a famous news website in China. In our setting, we pick the domain of health care and study the events of a certain medical company. To generate the required dataset, we use the following preprocessing steps:

1. We perform sentence segmentation and word segmentation on all the news documents and build inverted index on them.
2. We use the names of famous companies as keywords to search for all the documents related to the companies.
3. We regard all the documents related to one company as a testing corpus.

After preprocessing steps, we generate the descriptive statistics about the corpus in Table 2. We present the experimental results on the corpus related to world famous

[1] http://www.sina.com.cn/.

pharmaceutical companies such as Johnson &Johnson. The results for other companies are similar and not shown due to space limitation.

5.2 Baselines

To show the superiority of our method, we compare our WSCP against four baselines, namely, MaxCover, MaxLength, MinLength and Random. We now introduce these baselines in brief.

In MaxCover, we consider the coverage only when selecting the sentences, but do not consider the conciseness constraint. In this case, we solve the MCP instead of WSCP. In MaxLength, we simply select top M longest sentences, with the intuition that longer sentences may cover more keywords of the event. Conversely, for the sake the conciseness, MinLength selects top M shortest sentences in the corpus. For Random, the naive approach, we select M sentences in random.

5.3 Evaluation metrics

Our evaluation is based on three metrics. We have described coverage and conciseness in detail for single-event summarization. In our setting, we generate K summaries $S_D^{'}$ in total for all events. Thus, we define average coverage (AvgCov) and average conciseness (AvgCon) as the arithmetical means of the coverage and conciseness, respectively, shown as follows:

$$AvgCov(S_D^{'}) = \frac{\sum_{i=1}^{K} Cov(S_{D_i}^{'})}{K} \tag{12}$$

$$AvgCon(S_D^{'}) = \frac{\sum_{i=1}^{K} Con(S_{D_i}^{'})}{K} \tag{13}$$

However, AvgCov and AvgCon can be inconsistent in their trends. In order to have an overall metric that balances the AvgCov and AvgCon trade-off, inspired by the F1 measure, we first define the harmonic mean (HMean) of coverage and conciseness for an event as:

$$HMean\left(S_{D_i}^{'}\right) = \frac{2 \times Cov\left(S_{D_i}^{'}\right) \times Con\left(S_{D_i}^{'}\right)}{Cov\left(S_{D_i}^{'}\right) + Con\left(S_{D_i}^{'}\right)} \tag{14}$$

Then, we define the average HMean (AvgHMean) for all events:

$$AvgHMean\left(S_D^{'}\right) = \frac{\sum_{i=1}^{K} HMean\left(S_{D_i}^{'}\right)}{K} \tag{15}$$

5.4 Parameter tuning for event generation

In the experiment, we need to determine the values of T, the number of topics in LDA, and K, the number of events first.

5.4.1 Number of topics

T. The number of topics influences the performance of clustering. We compute the perplexity of the corpus D, which is employed in language modeling by convention, defined as:

$$perplexity(D) = exp\left\{-\frac{\sum_{i=1}^{|D|} log\ p(d_i)}{\sum_{i=1}^{|D|} N_i}\right\} \tag{16}$$

where N_i is the length of the ith document and $p(d_i$ is the probability of the ith document in language modeling. We test the perplexity in various settings. See Fig. 1a. We select $T = 30$ since it performs well and does not suffer from model overfitting.

5.4.2 Number of events

K. To determine the number of events K, we employ a step-by-step, iterative approach. We use the distortion value to evaluate the clustering. The distortion value is the sum of the distances of each point to its cluster centroid, defined as:

$$Distortion = \sum_{i=1}^{K}\sum_{j=1}^{|D_i|}(\theta_j - c_i)^2 \tag{17}$$

where c_i is the centroid of the ith cluster. The detailed evaluation method can be found in Hartigan and Wong (1979).

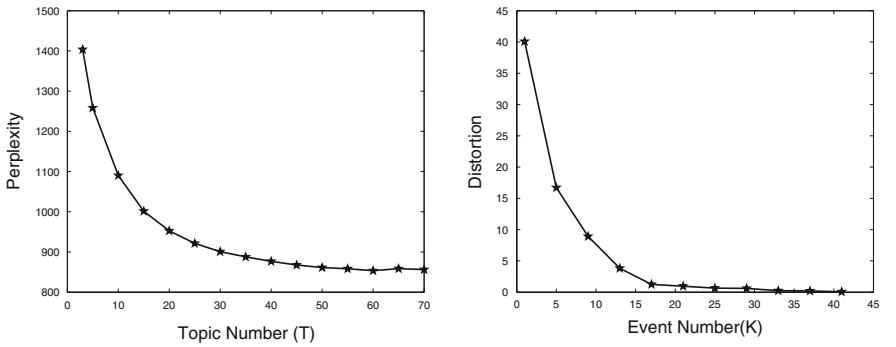
In practice, we first set $K = 1$ and perform clustering. At each time, we increase K with a step μ, i.e., $K = K + \mu$. We calculate the decrease in distortion value and select the final value of K until the decrease is less than a threshold. In the experiment, We set each step to be 4 and perform clustering iteratively. The distortion value begins to drop slowly between $K = 17$ and $K = 21$, shown in Fig. 1b. We then set $K = 18$.

5.5 Converge and conciseness

The objective of these experiments is to show the effects of parameters α and β in our WSCP optimization algorithm. We use the keywords of the events generated from the previous experiments. The results can be found in Fig. 2. We fix $M = 15$ in this set of experiments and explore how M varies can influence the result in the next section.

5.5.1 Varying α

To show the power of the conciseness constraint, we set α from 0.1 to 0.4.



(a) Selection of the number of topics T **(b)** Selection of the number of events K

Fig. 1 Varying T and K for event generation

In the experiment, as α increases from 0.1 to 0.4, we can see that the average coverage drops because the constraint effects. When $\alpha = 0$ and $\alpha = 0.1$, the coverage does not change significantly. This is due to the fact that when α is low, the algorithm does not impose a strong constraint on conciseness. There is an obvious drop when $\alpha = 0.3$ and $\alpha = 0.4$. Because when the conciseness constraint becomes strong, the converge is penalized.

On the other hand, just as the definition of WSCP itself suggests, as α increases, the conciseness grows. We have an interesting observation that when $\alpha = 0.4$, the average conciseness is 0.36, lower than 0.4. It is because the constraint is so strong that the WSCP has no solution for some events, which makes the conciseness 0. As a result, we should keep $\alpha \leq 0.4$.

In terms of AvgHMean, it shows that $\alpha = 0.2$ tends to have a better balance between having high coverage and high conciseness. We can also find the baseline MaxCover performs poorly due to low conciseness.

5.5.2 Varying β

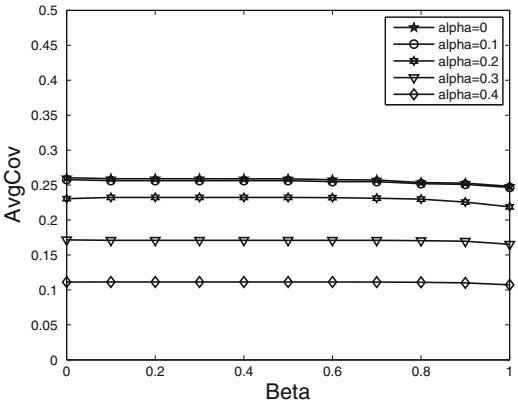
We analysis the effect of changes in β. We set β from 0 to 1 in the experiments.

For coverage, we can see when β increases and $0 \leq \beta \leq 0.7$, it does not influence the coverage much. However, when $\beta \geq 0.8$, the coverage begins to decrease. Because when the conciseness is not large, changes in β do not have a big impact on cost. In WSCP optimization algorithm, when we select sentences using the highest gain-cost-ratio, changes in β do not make a difference in the process. We observe a similar trend in conciseness when β increases. $\beta = 0.8$ is also an important turning point where conciseness rises fast when $\beta \geq 0.8$.

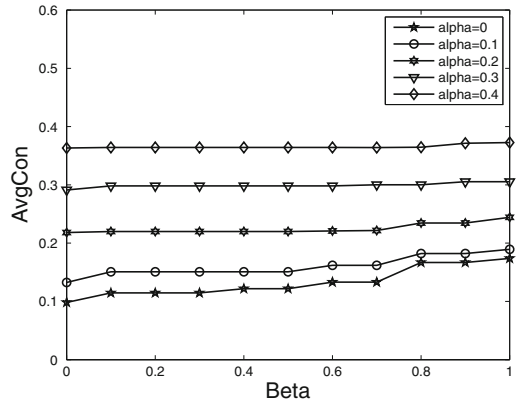
5.5.3 Parameter selection

To determine the suitable values of α and β, we continue to analyze the trend in AvgHMean. It reaches its peak 0.22 when $\alpha = 0.2$ and $\beta = 0.8$. Thus, we find

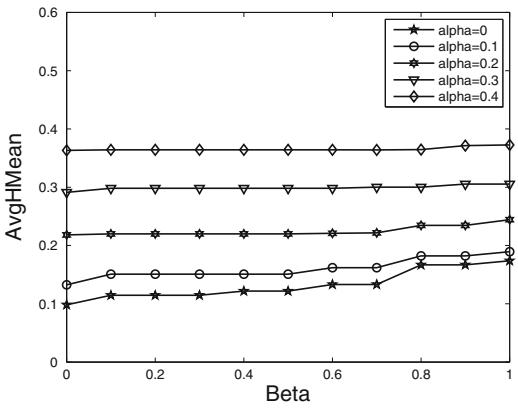
Fig. 2 Varying α and β for WSCP



(a) Average Coverage



(b) Average Conciseness



(c) Average HMean

a balance between coverage and conciseness. In the following experiments, we set $\alpha = 0.2$ and $\beta = 0.8$.

5.6 Performance comparison

In the following section, we compare our approach against all baselines. We use the same parameter setting to the previous experiments for WSCP. The results can be found in Fig. 3.

5.6.1 Coverage

As expected, MaxCover has the highest coverage because coverage is the only optimization objective. Our WSCP ranks second in all methods, which means our method does not sacrifice coverage much. In addition, WSCP outperforms remainder baselines. MinLength and Random have low coverage because they usually have few sentences that can cover a lot of keywords. When the number of sentences M increases, the coverage values of all methods rise, too.

5.6.2 Conciseness

In the experiment, our method outperforms all baselines greatly. MinLength has relatively high conciseness because it only selects short sentences, which do not have too much redundant information. MaxCover and MaxLength perform poorly because they both have high coverage while taking no consideration of conciseness. As the number of sentences M increases, the conciseness drops slowly in all the above methods. Random, as its name suggests, is not stable, so it has no clear trend.

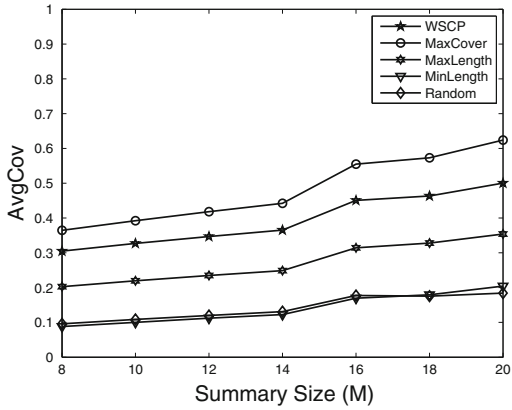
5.6.3 HMean

In AvgHMean, we can see the superiority of our method clearly. Our WSCP method outperforms the rest significantly. For example, when the number of sentences in summary $M = 18$, AvgHMean is equal to 0.253, about 13 % higher than all the baselines.

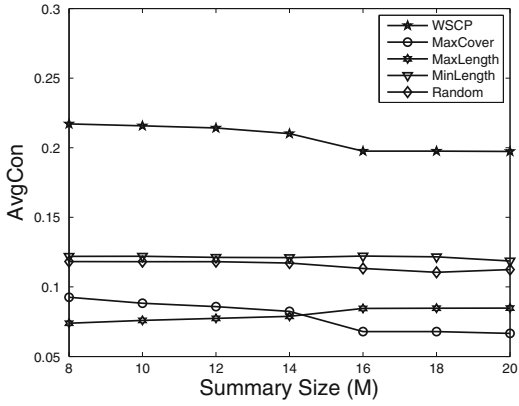
5.7 Case study

To better present our results, in this section, we provide a case study on summarization of Johnson & Johnson from real-life medical news articles. We present the date of the news and the corresponding summaries generated by our method. The detailed results can be found in Table 3. The results show the efficacy of our method to extract events and summaries for these events.

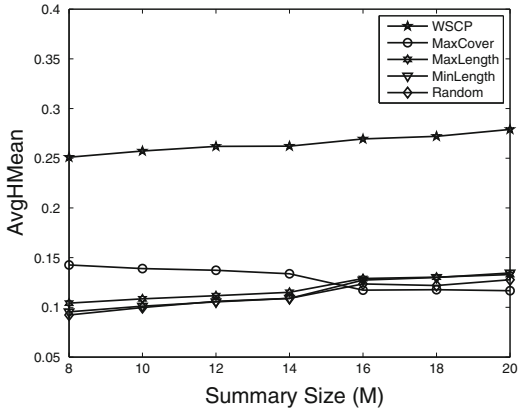
Comparing our WSCP approach with baselines, Table 4 illustrates the performance of our proposed approach and baselines. In the table, we can find that WSCP significantly outperforms the baselines w.r.t. conciseness and HMean. Consider coverage, MaxCover obtains the best performance since the summarization given by MaxCover



(a) Average Coverage



(b) Average Conciseness



(c) Average HMean

Fig. 3 Varying M for comparison between WSCP and baselines

Table 3 Case study of Johnson & Johnson event summarization

Date of news	Summary
10th July, 2010	McNeil, Johnson & Johnson's subordinate enterprise, announced to recall drugs again due to quality issues
22th July, 2010	Most of these drugs were produced in Johnson & Johnson's factory in Fort Washington, Pennsylvania. This is the third time when Johnson & Johnson recalls drugs this year
3rd August, 2010	The capital market also benefited from the reorganization of Johnson & Johnson. After the reorganization, the number of the taxis Johnson & Johnson held was doubled. The price of the stock increased by daily limit
26th November, 2010	Johnson & Johnson announced to recall more than 9 million bottles of Tylenol cold medicine. The reason was that the company did not add a note on the sticker of the bottle, indicating that the drug contains a moderate amount of alcohol
13th April, 2011	At least from 1998 until early 2006, the company illegally bribed doctors in public hospitals in Greece, to let them use surgical medical instruments produced by the company
29th April, 2011	Johnson & Johnson sued Guilin Zhonghui Biotechnology Co., Ltd. for violation against their right to a registered trademark. It illegally produced the matching strips for glucose meters of Johnson & Johnson's OneTouch series
16th July, 2011	Johnson & Johnson recalls 57,000 bottles of epilepsy drug TOPAMAX.
14th December, 2011	After the consultation with State Drug Administration of China, Johnson & Johnson's subordinate enterprise, Xian-Janssen Pharmaceutical Ltd. decided to conduct a level 3 preventive recall on Caelyx
28th June, 2012	Since the beginning of 2009, Johnson & Johnson's subordinate enterprise has recalled products 33 times in China

Table 4 Event summarization of Johnson & Johnson performance comparison

Approach	Coverage (%)	Conciseness (%)	HMean (%)
WSCP	51.73	21.51	30.27
MaxCover	65.13	7.57	13.32
MaxLength	34.98	7.23	11.88
MinLength	17.47	10.23	15.59
Random	17.63	9.76	12.25

only considers the top longest sentences. Naturally, it obtains the poor performance w.r.t. conciseness. Therefore, our approach gets the best trade-off between coverage and conciseness.

6 Conclusion and future work

In this paper, we introduce the problem of multi-event summarization. The method we present is novel both in the word set generation process for multiple events, as well as in the optimization of WSCP. The optimization problem is shown to be NP-hard, so we design a greedy algorithm to solve it. Experiments show that our method can discover and summarize multiple events in a text corpus.

The future work of our research lies in two aspects. First, we need to further improve the efficacy of our method to generate summaries from a large corpus. Second, we need to cover more topics in the domain of medicine and health care to support advanced applications from a optimization perspective.

Acknowledgments This work is partially supported by the National Basic Research Program (973) of China (No. 2012CB316203) and NSFC under Grant Nos. 61402177, 61170838 and 61272036. The author would also like to thank Key Disciplines of Software Engineering of Shanghai Second Polytechnic University under Grant No. XXKZD1301 and Project of Shanghai Shen-kang Hospital Development Centre (No. 2014SKMR-04).

References

Ablanedo-Rosas Rego (2010) Surrogate constraint normalization for the set covering problem. Eur J Oper Res 205:540–551

Alguliev RM, Aliguliyev RM, Hajirahimova MS, Mehdiyev CA (2011) Mcmr: maximum coverage and minimum redundant text summarization model. Expert Syst Appl 38:14514–14522

Avella P, Boccia M, Vasilyev I (2009) Computational experience with general cutting planes for the set covering problem. Oper Res Lett 37:16–20

Balas Carrera (1996) A dynamic subgradient-based branch-and-bound procedure for set covering. Oper Res 44:875–890

Becker H, Naaman M, Gravano L (2010) Learning similarity metrics for event identification in social media. In: Proceedings of the third ACM international conference on Web search and data mining, ACM, pp 291–300

Blei DM, Ng AY, Jordan MI (2003) Latent Dirichlet allocation. J Mach Learn Res 3:993–1022

Caprara A, Fischetti M, Toth P (1999) Aheuristic method for the set covering problem. Oper Res 47:730–743

Caragiannis I, Kaklamanis C, Kyropoulou M (2013) Tight approximation bounds for combinatorial frugal coverage algorithms. J Comb Optim 26:292–309

Chakrabarti D, Punera K (2011) Event summarization using tweets. In: ICWSM

Chieu HL, Ng HT (2002) A maximum entropy approach to information extraction from semi-structured and free text. In: Proceedings of the eighteenth national conference on artificial intelligence and fourteenth conference on innovative applications of artificial intelligence, Edmonton, Alberta, Canada. pp 786–791, 28 July–1 August 2002

Conroy JM, O'leary DP (2001) Text summarization via hidden markov models. In: Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval, ACM, pp 406–407

Das D, Martins AF (2007) A survey on automatic text summarization. Lit Surv Lang Stat Course CMU 4:192–195

Deng G, Lin W (2011) Ant colony optimization-based algorithm for airline crew scheduling problem. Expert Syst Appl 38:5787–579

Ester M, Kriegel HP, Sander J, Xu X (1996) A density-based algorithm for discovering clusters in large spatial databases with noise. KDD 96:226–231

Fattah MA, Ren F (2008) Automatic text summarization. World Acad Sci Eng Technol 37:2008

Fisher Kan R (1988) The design, analysis and implementation of heuristics. Manag Sci 34:263–265

Friedman JH (1997) On bias, variance, 0/1loss, and the curse-of-dimensionality. Data Min Knowl Discov 1:55–77

García-Hernández RA, Ledeneva Y (2009) Word sequence models for single text summarization. In: Advances in computer-human interactions, 2009. Second International Conferences on ACHI'09, IEEE, pp 44–48

Gupta V, Lehal GS (2010) A survey of text summarization extractive techniques. J Emerg Technol Web Intell 2:258–268

Hartigan JA, Wong MA (1979) Algorithm as 136: a k-means clustering algorithm. Appl Stat 28:100–108

Kruengkrai C, Jaruskulchai C (2003) Generic text summarization using local and global properties of sentences In: Web intelligence, 2003. WI 2003. Proceedings. International Conference on IEEE/WIC, IEEE, pp 201–206

Kupiec J, Pedersen J, Chen F (1995) A trainable document summarizer. In: Proceedings of the 18th annual international ACM SIGIR conference on Research and development in information retrieval, ACM, pp 68–73

Kyoomarsi F, Khosravi H, Eslami E, Dehkordy PK, Tajoddin A (2008) Optimizing text summarization based on fuzzy logic. In: ACIS-ICIS, pp 347–352

Lin CY (1999) Training a selection function for extraction. In: Proceedings of the eighth international conference on information and knowledge management, ACM, pp 55–62

Lin J (1991) Divergence measures based on the shannon entropy. IEEE Trans Inf Theory 37:145–151

Radev DR, Hovy E, McKeown K (2002) Introduction to the special issue on summarization. Comput Linguist 28:399–408

Sakaki T, Okazaki M, Matsuo Y (2010) Earthquake shakes twitter users: real-time event detection by social sensors. In: Proceedings of the 19th international conference on World wide web, ACM, pp 851–860

Salton G, McGill M (1984) Introduction to modern information retrieval. McGraw-Hill Book Company, New York

Svore KM, Vanderwende L, Burges CJC (2007) Enhancing single-document summarization by combining ranknet and third-party sources In EMNLP-CoNLL 2007, In: Proceedings of the 2007 joint conference on empirical methods in natural language processing and computational natural language learning, Prague, Czech Republic, pp 448–457, 28–30 June 2007

Takamura H, Okumura M (2009) 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, Association for Computational Linguistics, pp 781–789

Tsolmon B, Lee K (2014) An event extraction model based on timeline and user analysis in latent dirichlet allocation. In: The 37th international ACM SIGIR conference on research and development in information retrieval, SIGIR '14, Gold Coast, QLD, Australia, pp 1187–1190, 06–11 July 2014

Umetani, Yagiura (2007) Relaxation heuristics for the set covering problem. J Oper Res Soc Jpn 50:350–375

Yaghini M, Karimi M, Rahbar M (2013) A set covering approach for multi-depot train driver scheduling. J Comb Optim pp 1–19