Journal of Biomedical Informatics 88 (2018) 53–61

Contents lists available at ScienceDirect

Journal of Biomedical Informatics

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

CIBS: A biomedical text summarizer using topic-based sentence clustering

Milad Moradi

Department of Electrical and Computer Engineering, Isfahan University of Technology, Isfahan 84156-83111, Iran

ARTICLE INFO

Keywords:
Natural Language Processing
Medical text mining
Itemset mining
Domain knowledge
Multi-document summarization
Coverage

ABSTRACT

Automatic text summarizers can reduce the time required to read lengthy text documents by extracting the most important parts. Multi-document summarizers should produce a summary that covers the main topics of multiple related input texts to diminish the extent of redundant information. In this paper, we propose a novel summarization method named Clustering and Itemset mining based Biomedical Summarizer (CIBS). The summarizer extracts biomedical concepts from the input documents and employs an itemset mining algorithm to discover main topics. Then, it applies a clustering algorithm to put the sentences into clusters such that those in the same cluster share similar topics. Selecting sentences from all the clusters, the summarizer can produce a summary that covers a wide range of topics of the input text. Using the Recall-Oriented Understudy for Gisting Evaluation (ROUGE) toolkit, we evaluate the performance of the CIBS method against four summarizers including a state-of-the-art method. The results show that the CIBS method can improve the performance of single- and multi-document biomedical text summarization. It is shown that the topic-based sentence clustering approach can be effectively used to increase the informative content of summaries, as well as to decrease the redundant information.

1. Introduction

A summary of text documents is a shorter version of the original text that conveys the most important information [1]. Using a summarization system, users can assess main ideas within many documents without any requirement to read the whole content. In the biomedical domain, there are various sources of textual information and the number of text documents available to clinicians and researchers is growing rapidly [2]. To cope with the difficulties of attaining desired information from the large volume of biomedical text resources, many efforts have been made towards developing domain-specific summarization systems [3]. Following the trend of knowledge-rich summarization, the majority of these methods utilize sources of domain knowledge to deal with the singularities of text documents in the biomedical domain [2–5].

A high degree of information coverage is one of the main features an useful summary should have [1]. This property refers to covering a wide range of topics appearing in the original text. A summarization system should take into account the information coverage especially when it summarizes a collection of related documents [6]. In this case, there are many similar sentences in different documents of the collection. If the summarizer covers the main topics, the information diversity will increase and as a result the information redundancy will decrease. To do so, the summarizer needs to identify important topics within the whole document collection. Then, it should put similar sentences into the same group. Extracting the most relevant sentences of each group, the produced summary can cover all the important ideas within the collection. Following the above idea, we propose a novel method named **C**lustering and **I**temset mining based **B**iomedical **S**ummarizer (CIBS).

The CIBS method begins with a preprocessing step, mapping the input text to concepts contained in the Unified Medical Language System (UMLS) [7]. This mapping phase helps the summarizer to have an approximation of the semantics of sentences. This semantic can refer to denotations of the words. The summarizer passes the sentences and concepts to an itemset mining algorithm, where the main topics are extracted. Next, a clustering algorithm groups the sentences into multiple clusters. Sentences within the same cluster cover similar topics. The summarizer selects the most related sentences from all the clusters to produce a summary containing a wide range of main ideas of the original text. We evaluate the performance of our CIBS method for both single- and multi-document biomedical text summarization in comparison with four summarizers including a state-of-the-art method. The results show that the CIBS summarizer can perform better than the comparison methods in terms of the Recall-Oriented Understudy for Gisting Evaluation (ROUGE) [8] metrics.

The remainder of this paper is organized as follows. In Section 2, we give an overview of related work. Section 3 gives a detailed description

E-mail address: milad.moradi@ec.iut.ac.ir.

https://doi.org/10.1016/j.jbi.2018.11.006
Received 10 February 2018; Received in revised form 26 September 2018; Accepted 12 November 2018
Available online 13 November 2018

of the CIBS method and describes the evaluation methodology. We present and discuss experimental results in Section 4. Finally, we point out some concluding remarks and future lines of work in Section 5.

2. Related work

Summarization systems are divided into two broad categories of extractive and abstractive [1,9]. An extractive summarizer generates the final summary by extracting and putting together some parts of the original input, while an abstractive method produces a new text conveying the important content. Abstractive summarization can be much more intricate than extractive since it needs a wide range of natural language processing techniques for re-interpreting the content and producing a new text [1]. The increasing feasibility of developing extractive methods has led the research community to making the majority of effort towards addressing the challenges of extractive summarization, as the latest survey in biomedical text summarization [3] includes 27 extractive versus 7 abstractive studies. In this paper, we address extractive summarization.

According to the number of inputs, summarization methods are single- or multi-document [1,9]. There are some issues that may affect the quality of text summaries, especially in multi-document summarization, where relevant information is distributed among many inputs [6]. For example, searching for articles related to a specific disease in the MEDLINE biomedical literature database may result hundreds of documents. In this case, all the retrieved articles may contain important information and the user needs to be provided with a summary that conveys the most relevant information within the documents. Therefore, the summary should cover a variety of important topics as far as possible [6,10]. In addition to the coverage, the presence of redundant information is another issue that the summarizer should deal with [6,11]. However, it should be taken into account that there is a trade-off between coverage and redundancy. If a summary covers only a set of important topics, it may contain some redundant information. On the other hand, having a small degree of redundancy may lead to covering a set of unimportant contents. With respect to this trade-off, we use ROUGE metrics to evaluate how much informative content is covered by summaries. We also assess the occurrence of concepts and topics included in summaries to evaluate the redundancy. Adopting an efficient strategy to select important sentences containing various topics within a collection of documents, we show that our CIBS method can effectively deal with both the coverage and redundancy issues.

Many available summarization methods employ generic features including the length of sentences, the position of sentences, the presence of cue phrases, etc. to identify the most important parts of an input text [12–14]. These features have shown their usefulness for summarization of specific types of documents such as newswire articles [15]. However, they may not serve as adequate measures for summarization of biomedical text documents due to variety of document types and some issues such as various synonymous words for an entity, homonym words, and abbreviations [2,4]. Some previous efforts improved the performance of biomedical summarizers by utilizing domain-specific knowledge sources [2,4,16,17,40,41]. It has been shown that the quality of summaries can be improved when the summarizer assesses the informative content of sentences based on their approximated semantics rather than using generic measures. It is worth-mentioning that the usage of sources of domain knowledge has also led to improvements in other tasks where text summarization is utilized as a peripheral method to achieve other goals rather than only providing the user with a textual summary. Among these tasks we can point out linking databases of Gene functions to the biomedical literature [18], quality assurance of Gene functions stored in the Entrez Gene database [19], personalized retrieval and summarization of images, video and language [20], extracting drug interventions from clinical reports [21], and question answering [22].

In recent years, many summarization methods have been proposed for biomedical text documents. Mishra et al. [3] categorized these methods into four groups, i.e. statistical, machine learning, natural language processing, and hybrid techniques. Biomedical summarizers have been considered for summarization of different types of documents such as biomedical literature [2,23], electronic health records [24], and clinical free text notes [25]. Since dealing with terms in a biomedical text document can lead to various challenges [2,4], there has been a trend toward developing knowledge rich methods. These methods utilize sources of domain knowledge to construct a concept-based model from the input text. In this way, the summarizer can decide about the importance and relatedness of sentences according to their approximated semantics [2,4,16], leading to an improvement in the quality of summaries.

The UMLS [7] is a well-known source of knowledge in the biomedical domain, employed by several domain-specific summarization systems. It consists of three main components, i.e. the Specialist Lexicon [26], the Metathesaurus [27], and the Semantic Network [28]. Containing a set of English and biomedical vocabularies, the Specialist Lexicon serves as a database for lexicographic information. The Metathesaurus stores large collections of biomedical and health-related concepts along with their semantic information. The Semantic Network defines a set of semantic types categorizing the Metathesaurus concepts into subject classifications. Our CIBS summarizer utilizes UMLS in a preprocessing step to extract biomedical concepts and prepare the input text for identifying main topics.

Itemset mining is a data mining technique for discovering frequent and meaningful patterns in different types of datasets [29–31]. The itemset-based biomedical summarizer is a method employing itemset mining for extracting important topics in single-document summarization [2]. It uses the support values of frequent itemsets to score the sentences of the input text. The problem of redundant information was not addressed by the itemset-based summarizer, but it was shown that frequent itemsets can be effectively used to quantify the informativeness of sentences and produce summaries with higher levels of informative content. The first contribution of the current research is to show that the usage of itemsets should be reinforced to address the redundancy and information coverage issues in multi-document summarization. Our CIBS method incorporates strategies to assess the shared content between sentences in terms of important topics, to identify sets of semantically related sentences, and to deal with the redundancy issue. The second contribution is to show that compared to the generic methods like TexLexAn [14], SUMMA [12], and MEAD [13], the topic-based sentence clustering approach used by the CIBS method can significantly improve the performance of both single- and multi-document biomedical summarization.

3. Methods

3.1. Summarization method

Our CIBS method consists of four main steps: (1) preprocessing, (2) topic extraction, (3) sentence clustering, and (4) summary generation. Fig. 1 illustrates the architecture of the CIBS method. We give a detailed description of each step in the following.

3.1.1. Preprocessing

The summarization process begins with a preprocessing step in which the input documents are mapped to concepts. For the concept extraction task, the CIBS method utilizes the MetaMap program [32] developed by the US National Library of Medicine. This program utilizes natural language processing techniques to identify noun phrases within the input text and map them to concepts contained in the UMLS metathesaurus. If there are multiple mapping candidates with the same score for a noun phrase, MetaMap returns all the candidates, and the CIBS method considers all of them. This strategy has obtained satisfactory results in concept-based biomedical text summarization [33].

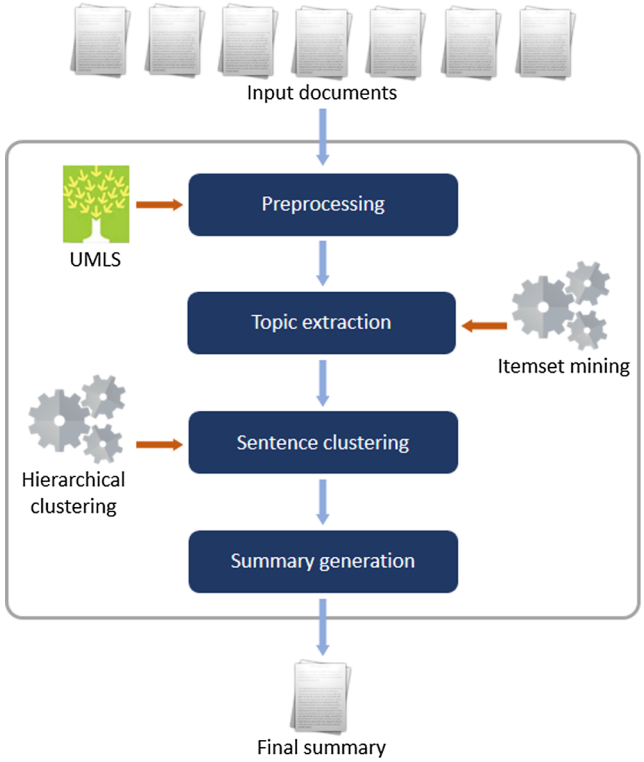

Fig. 1. The architecture of the CIBS method.

Each concept belongs to a semantic type which determines the semantic classification of the concept. For example, Fig. 2 shows a sentence from a sample text and the concepts and their semantic types extracted by MetaMap. There are nine semantic types experimentally identified as generic and unimportant in this type of concept-based text modeling [4]. After concept extraction, the summarizer removes concepts categorized in these semantic types. Table 1 presents the generic semantic types and their definitions.

Table 1
Nine generic semantic types and their definitions. Concepts belonging to these semantic types are discarded in the preprocessing step.

Semantic type	Definition
qnco	Quantitative Concept
qlco	Qualitative Concept
tmco	Temporal Concept
ftcn	Functional Concept
idcn	Idea or Concept
inpr	Intellectual Product
menp	Mental Process
spco	Spatial Concept
lang	Language

For the preprocessing step, we use the 2016v2 release of MetaMap along with the 2016AB version of UMLS and 'USAbase' as the data version. We use this list of parameters to run the MetaMap program: -V USAbase -L 15 -Z 2016AB -E -Ay –XMLf –E. The output of the program is a file whose content is encoded within XML tags that facilitate automatic reading and processing. There are many tags in the file, each one containing different information related to the input text. With respect to the task at hand, it can be determined which tags should be processed. Our summarizer identifies the tags that contain sentences, mappings, and concepts. Sentences are encoded within individual < Utterance > tags. For each sentence, phrases are encoded within < Phrase > tags. For each phrase, different mappings are contained within hierarchical < Mapping >

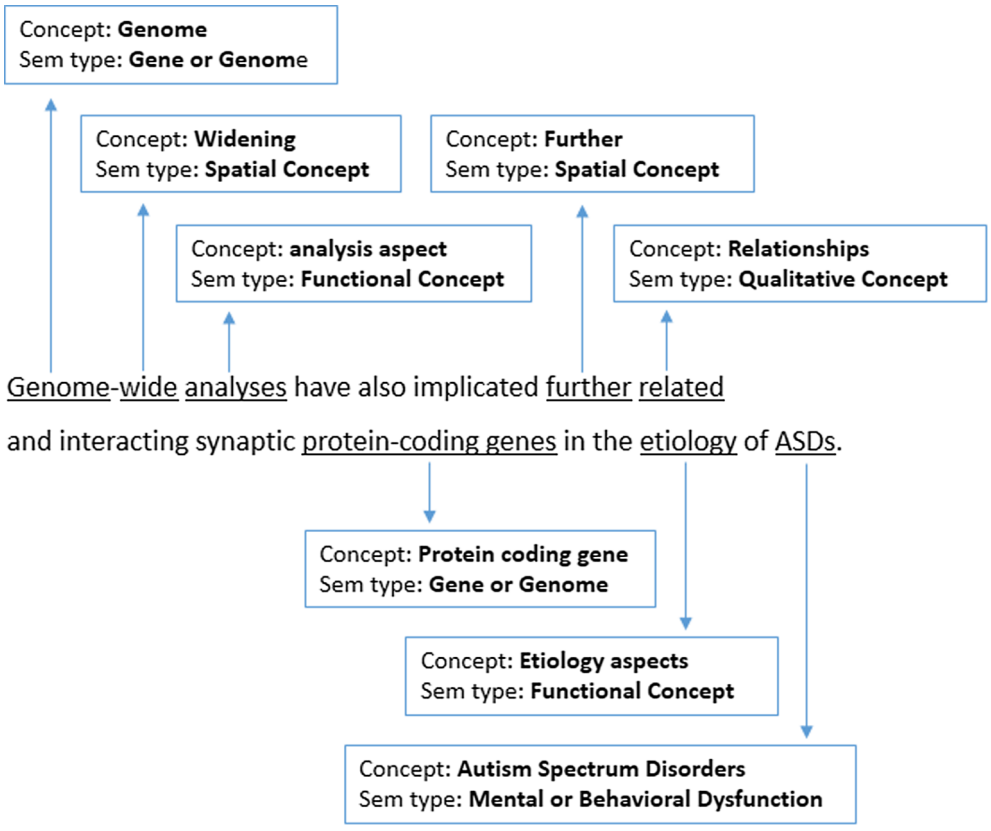

Fig. 2. A sample sentence and the UMLS concepts extracted in the preprocessing step.

and < Candidate > tags that hold candidate concepts. A semantic type is held by a < SemType > tag for each candidate concept. The summarizer reads the information contained within the relevant tags and organizes the input text in the form of a data structure in which individual sentences contain a list of paired concepts and semantic types.

3.1.2. Topic extraction

In this step, the summarizer discovers correlated concepts within the input documents and considers them as the main topics. There are multiple documents splitted into sentences after the preprocessing step. Each sentence contains a number of concepts which could be used to approximate topics referred to in the sentence. The summarizer deals with the input documents as an unified text and employs an itemset mining method to extract correlated and recurrent concepts. To this aim, our summarizer uses a variation of the Apriori algorithm [34].

The input text must be converted to an appropriated format to be prepared for the itemset mining algorithm. Creating a transaction for each sentence, the summarizer generates a transactional dataset. The concepts contained in each sentence make the items of the corresponding transaction. The itemset mining algorithm receives the transactional representation T and a minimum support threshold *min_sup*, then it returns a set of frequent itemsets *FI* as the output. Each item has a property called 'support' which is calculated as the proportion of transactions containing the item to the total number of transactions [35,36]. A k-itemset is a set of k distinct items. The support property is also defined for an itemset as the proportion of transactions containing all items of the itemset to the total number of transactions. The support property of an item must be greater than or equal to the threshold *min_sup* in order to the item to be identified as a frequent item. Likewise, a k-itemset is said to be frequent, if all subsets of the itemset are frequent.

Algorithm 1 gives a pseudo-code of the itemset mining algorithm employed by the CIBS summarizer. At the beginning, the set of frequent itemsets *FI* is initialized (line 1) and frequent 1-itemsets are specified (lines 2–6). Then, the process of discovering frequent k-itemsets begins with $k = 2$ (line 7). In each iteration of this process, a set of candidate itemsets *CI* is constructed by joining the set FI_{k-1} with itself (line 10). FI_{k-1} is a subset of *FI* containing itemsets with size k-1. Next, for each candidate itemset, the algorithm specifies whether all the subsets are frequent (lines 11–13). If so, the itemset is retained as a candidate (line 14), otherwise it is pruned (lines 15–16). Afterwards, each remaining candidate is tested as an unified itemset whether it is frequent (lines 19–20). If so, the candidate is added to the set *FI* (line 21). After extracting all frequent k-itemsets, k increases by one (line 24), and the next iteration runs if at least one itemset has been added to *FI* in the latest iteration (line 8).

Table 2
Frequent itemsets extracted from a sample document.

Itemset	Support
{Schizophrenia}	0.352
{Bipolar Disorder}	0.235
{Bipolar Disorder, Schizophrenia}	0.223
{Autism Spectrum Disorders}	0.223
{Autistic Disorder}	0.200
{neuroligin}	0.176
{Deletion Mutation}	0.164
{Autistic Disorder, Schizophrenia}	0.152
{Genes}	0.129
{NRXN1 gene}	0.129
{Autistic Disorder, Bipolar Disorder, Schizophrenia}	0.105
{Autistic Disorder, Bipolar Disorder}	0.105
{Copy Number Polymorphism}	0.105
{Genome}	0.105
{Persons}	0.105

Algorithm 1. The frequent itemset mining algorithm employed by the CIBS summarizer.

Input: transactional representation T
Input: minimum support threshold *min_sup*
Output: set of frequent itemsets *FI*
1: $FI = \emptyset$
2: **for each** distinct item $item_i$ in T **do**
3: **if** $Support(item_i) >= min\_sup$ **then**
4: create a new itemset containing only $item_i$ and add it to *FI*
5: **end if**
6: **end for**
7: $k = 2$
8: **while** at least one frequent itemset has been added to *FI* in the latest iteration **do**
9: $CI = \emptyset$ /* *CI* is a set holding candidate k-itemsets in each iteration */
10: join FI_{k-1} with itself according to a rule stating that itemsets are joined if they have same first k-1 items, and add the resulting itemsets to *CI*/* FI_{k-1} is a subset of FI containing itemsets with size k-1 */
11: **for each** itemset CI_j in *CI* **do**
12: SB = all subsets of CI_j with size $k - 1$
13: **if** $Support(all\ itemsets\ in\ SB) >= min\_sup$ **then**
14: retain CI_j in *CI*
15: **else**
16: prune CI_j
17: **end if**
18: **end for**
19: **for each** remaining itemset CI_k in *CI* **do**
20: **if** $Support(CI_k) >= min\_sup$ **then**
21: add CI_k to *FI*
22: **end if**
23: **end for**
24: k = k + 1
25: **end while**
26: **return** *FI*

Each frequent itemset resulted from the itemset mining algorithm represents a set of concepts that appear frequently together within the input text. The summarizer considers these frequent itemsets as the main topics of the text. As an example, Table 2 presents the frequent itemsets extracted from a sample document, a scientific article[1] related to the genetic overlap of three mental disorders. In this case, the value of the threshold *min_sup* is set to 0.1. However, it is not the optimum value, and we will tune the parameter *min_sup* in Section 4.1. As can be seen in Table 2, 11 itemsets contain only one item. There are also three 2-itemests and one 3-itemset. The itemset {Schizophrenia} is the most frequent one denoting the concept *Schizophrenia* appears in 35% of sentences of the text. The most frequent itemset containing more than one item is {Bipolar Disorder, Schizophrenia}. This means the concepts *Bipolar Disorder* and *Schizophrenia* appear together in 22% of sentences of the text. Regarding the 3-itemset {Autistic Disorder, Bipolar Disorder, Schizophrenia}, we can find out that 10% of the sentences contain all the three concepts *Autistic Disorder*, *Bipolar Disorder*, and *Schizophrenia*. The frequent itemsets denote the main topics conveyed by the sentences. The summarizer utilizes these topics to cluster the semantically related sentences into separate groups (the next step in the summarization process).

3.1.3. Sentence clustering

After discovering the main topics of the input text, the summarizer groups the sentences according to the topics that each sentence covers. It identifies each set of sentences having some topics in common and puts them in the same group. Selecting sentences from all the groups, the summarizer can cover all the topics as much as possible. This grouping and selection strategy can lead to a summary with a high degree of information diversity and a low degree of redundancy.

In this step, the summarizer employs an agglomerative hierarchical

[1] Available at http://genomemedicine.biomedcentral.com/articles/10.1186/gm102

clustering algorithm [34] to group the sentences into a set of clusters. The similarity of sentences within each cluster should be maximum and the similarity to the sentences outside the cluster should be minimum. The summarizer uses the frequent itemsets (topics) shared between each pair of sentences to estimate the important content they have in common. The clustering algorithm begins with an initializing phase in which each sentence is considered as a cluster of its own. Then, the two closest clusters are merged together and form a new cluster in an iterative manner. In each iteration, the number of clusters is reduced by one. The algorithm proceeds until the number of clusters reaches a predefined value.

At the beginning of the clustering algorithm, before creating initial clusters, a coverage matrix $C_{M\times M}$ is constructed, where M is the total number of sentences within whole the input collection (input documents). Each element $C_{i,j}$ of the coverage matrix holds a coverage value between two sentences S_i and S_j. The coverage value denotes how much important information is shared between sentences S_i and S_j in terms of the important topics. This value is calculated by summing up the support values of frequent itemsets appearing in both sentences S_i and S_j, as follows:

$$C_{i,j} = Coverage(S_i, S_j) = \sum_p Support(Itemset_p) \quad (1)$$

where $Support(Itemset_p)$ is the support value of p^{th} frequent itemset appearing in both sentences S_i and S_j.

In Eq. (1), it is not required to divide the sum of support values by the total number of shared itemsets. This is because the coverage value does not act as a similarity measure, hence, there is no need to normalize it. As mentioned above, the coverage value is interpreted as the amount of shared important information between two sentences. The more shared frequent itemsets between two sentences and the higher support values, the greater the coverage value.

After calculating all the coverage values within the coverage matrix C, the clustering algorithm creates initial clusters and begins the iterative merging process. In each iteration, the algorithm identifies a pair of clusters with the greatest between-cluster coverage value and merges them into a new cluster. Given the first cluster CL_q with only one sentence S_i, and the second cluster CL_r with only a sentence S_j, the between-cluster coverage is equal to the value of $C_{i,j}$. If there are more than one sentence in each one of the clusters, the between-cluster coverage is calculated as the arithmetic mean of coverage values between each member of the first cluster and each member of the second, as follows:

$$BCC(CL_q, CL_r) = \frac{\sum_{i,j} C_{i,j}}{NCV} \quad (2)$$

where $BCC(CL_q, CL_r)$ is the between-cluster coverage value of two clusters CL_q and CL_r. The value of $C_{i,j}$ is equal to the coverage value between two sentences S_i and S_j such that S_i is i^{th} member in CL_q and S_j is j^{th} member in CL_r. The value of NCV is equal to the total number of coverage values summed up in the numerator.

The clustering algorithm merges two clusters in each iteration. The algorithm proceeds until the number of clusters reaches a predefined value FC specifying the number of final clusters. If a sentence is highly unique and does not contain any important topics, eventually it is assigned to a random cluster. Since the sentence does not share any important information with other sentences in the cluster, there is no coverage value that can be computed for the sentence, hence, the sentence has no chance to be selected for the summary. Algorithm 2 gives a pseudo-code of the topic-based sentence clustering algorithm employed by our CIBS method. Firstly, initial clusters are constructed (lines 2–4). Next, in each iteration of the clustering algorithm, between-cluster coverage values are computed for every pair of clusters (lines 6–7), and two clusters with the largest value are merged (line 9). The algorithm stops when the size of CL reaches the number of final clusters FC (line 5).

Algorithm 2. The topic-based sentence clustering algorithm employed by the CIBS summarizer.

Input: set of sentences S
Input: coverage matrix C
Input: number of final clusters FC
Output: set of clusters CL
1: CL = ∅
2: **for all** S_i in S **do**
3: create a new cluster CL_i and add it to CL
4: **end for**
5: **while** the size of CL is greater than FC **do**
6: **for all** pairs of clusters CL_q and CL_r in CL **do**
7: calculate the between-cluster coverage using matrix C
8: **end for**
9: merge a pair of clusters having the maximum between-cluster coverage
10: **end while**
11: **return** CL

3.1.4. Summary generation

In the last step, the summarizer uses the clusters produced in the clustering phase to create the final summary. As explained earlier, the clustering algorithm groups sentences into clusters such that sentences within the same cluster have the maximum between-cluster coverage. In this way, each cluster contains a set of sentences that share some important topics of the whole text. Selecting sentences from all the clusters, the summarizer generates the final summary. It extracts summary sentences from all clusters to cover all the main topics within the input text. Each cluster contributes to the summary in proportion to its size, as follows:

$$N_q = N\frac{|CL_q|}{M} \quad (3)$$

where N_q is the number of sentences that should be selected from q^{th} cluster, $|CL_q|$ is the size of q^{th} cluster, M is the total number of sentences within whole the input text, and N is the number of sentences that must be selected for the final summary.

Next, the summarizer extracts the most related and informative sentences within each cluster. For each cluster CL_q, it computes the total coverage values between each sentence S_i and all the other sentences in CL_q. Then, it ranks the sentences of CL_q based on their total coverage values. The summarizer considers the top ranked sentences in each cluster as the most related and informative ones because they share the largest amount of information with other sentences within the cluster. Finally, the summarizer selects the top N_q sentences from the ranked list of each cluster CL_q, puts them together, and produces the final summary.

3.2. Evaluation method

3.2.1. Evaluation metrics

We use the ROUGE package [8] to evaluate the performance of automatic summarizers. ROUGE compares the system and model summaries and produces different scores, each one assesses the shared content based on different metrics. ROUGE evaluates the performance of summarization methods with respect to the informative content of summaries, hence, it can be an appropriate metric for our experiments since we need to investigate how much important content is conveyed by the summaries. In this research, we use two types of ROUGE metrics, i.e. ROUGE-2 (R-2) and ROUGE-SU4 (R-SU4). Both the metrics estimate the shared content in terms of bigrams. ROUGE-SU4 also allows a maximum skip distance of four between bigrams. ROUGE metrics are simple and this allows the results to be interpreted easily. In addition, ROUGE scores have shown significant correlation with the evaluations done by human assessors in the Document Understanding Conference (also known as DUC) [4].

Table 3
The topics of collections in the multi-document evaluation and parameterization corpora.

Corpus	Topics
Evaluation (25 collections)	Aphasia, Arthritis, Asthma, Bipolar disorder, Botulism, Bronchiectasis, Cellulitis, Chickenpox, Conjunctivitis, Diabetes, Diphtheria, Eczema, Endometriosis, Gout, Hepatitis, Influenza, Measles, Meningococcal disease, Mumps, Norovirus, Osteoporosis, Rotavirus, Rubella, Tetanus, Tuberculosis
Parameterization (10 collections)	Anencephaly, Autism, Chorea, Dementia, Epilepsy, Gaucher disease, Hypersomnia, Microcephaly, Myotonia, Neurotoxicity

3.2.2. Evaluation corpus

In the performance evaluation of automatic summarization, systems are provided with a corpus of documents and model summaries. System-produced summaries are compared with model summaries to assess the performance of automatic summarizers. The more the shared content between the system and model summaries, the higher the performance of the summarizer. Since there is no standard corpus for biomedical documents and their model summaries, we create two types of corpora to conduct different evaluations. A description of the corpora is given in the following.

- **Multi-document corpus:** This corpus consists of 25 collections, each one contains 300 documents and a model summary. The topic of each collection is a disease. We searched the name of disease in PubMed and stored the first 300 abstracts retrieved in the result. We stored the definition of disease provided by Wikipedia as the model summary for the collection. This approach to creating a multi-document corpus was also previously adopted by Shang et al. [37]. The summarizer receives all the 25 collections and generates a summary for each one. According to the work done by Lin [38], when there is one reference (or model) summary for each document set in a corpus for multi-document summarization, the critical number of sets is 10 with respect to a Pearson's correlation analysis with a confidence interval of 95%. This means there must be at least 10 sets within the corpus in order to ROUGE scores obtained by the summarizer to be reliable. Therefore, the size of our evaluation corpus is large enough to allow the results of evaluations to be statistically significant. Likewise, we created a separate development corpus containing 10 collections for parameterization in multi-document experiments. Table 3 gives the topics of collections in each corpus.

The topics were selected randomly from a list of diseases. The documents in each collection cover both types of clinical trial and review. The 20 most frequent semantic types within both the corpora are *Disease or Syndrome*, *Finding*, *Sign or Symptom*, *Diagnostic Procedure*, *Health Care Activity*, *Activity*, *Human-caused Phenomenon or Process*, *Laboratory Procedure*, *Clinical Attribute*, *Organism Function*, *Phenomenon or Process*, *Behavior*, *Research Activity*, *Biologic Function*, *Experimental Model of Disease*, *Substance*, *Body Part Organ or Organ Component*, *Human*, *Body Substance*, and *Pathologic Function*.

For the evaluation corpus, the minimum, maximum, and average number of sentences in a collection is 1877, 2937, and 2540 respectively. For the development corpus, the minimum, maximum, and average number of sentences in a collection is 1868, 2925, and 2384 respectively.

- **Single-document corpus:** This corpus consists of 400 scientific biomedical articles randomly selected from the BioMed Central's corpus for text mining research [39]. The abstracts of articles are used as model summaries to evaluate the summarization methods. This approach was also adopted by some previous work [2,4,5,17]. Likewise, we create a separate development corpus containing 100 articles to conduct parameterization in single-document experiments. The minimum, maximum, and average number of sentences of a document in the evaluation corpus is 57, 401, and 173 respectively. The minimum, maximum, and average number of sentences of a document in the development corpus is 59, 387, and 177 respectively.

3.2.3. Comparison methods

We compare the performance of the CIBS method with four summarizers, i.e. the itemset-based summarizer, TexLexAn, SUMMA, and MEAD. The itemset-based summarizer [2] is a state-of-the-art biomedical summarization method based on concept extraction and itemset mining. It uses the support values of frequent itemsets to assign a score to each sentence. Selecting sentences with the highest scores, the method generates the final summary. TexLexAn [14] is an open source summarizer that uses keywords and cue expressions to extract the most important sentences. SUMMA [12] is a single- and multi-document summarizer that uses some measures including word frequency, the position of sentences, and similarity features for sentence selection. Another multi-document summarizer in our comparison is MEAD [13]. It employs sentence position, sentence length, similarity to the first sentence, and similarity to the centroid features. In the final evaluation experiments, we run all the comparison methods by their best settings specified on the single- and multi-document development corpora.

In multi-document experiments, each system extracts 40 sentences from each collection and returns them as the summary. In single-document experiments, each system extracts 30 percent of sentences within a document and returns them as the summary. We use a Wilcoxon signed-rank test with a confidence interval of 95% to test the statistical significance of the results obtained by the summarizers.

4. Results and discussion

4.1. Parameterization

There are two parameters involved in the topic extraction and sentence clustering steps. The *min_sup* threshold specifies the minimum support value for an itemset to be considered as a frequent one in the itemset mining algorithm. The clustering algorithm uses the parameter *FC* to determine the number of final clusters. Conducting two sets of separate experiments on single- and multi-document development corpora, we evaluate the performance of the CIBS method under different values of *min_sup* in the range [0.01, 0.2] and *FC* in the range [1,8]. In addition to evaluating ROUGE scores, we also assess the occurrence of concepts within summaries in order to investigate both the coverage and redundancy.

Table 4 presents the ROUGE scores obtained by the CIBS method for multi-document summarization, using different values of *FC*. The scores are presented only for the best value of *min_sup* specified for each *FC* value. The summarizer obtains the highest scores when the number of final clusters is equal to 2, 3, or 4. The scores obtained for $FC = 3$ are significantly ($p < 0.05$) higher than those of other values of *FC* except 2 and 4. In the following, we discuss the results with respect to the trade-off between information coverage and redundancy in terms of three different cases. We use the ROUGE scores as an indicator of information coverage and the average number of unique concepts as an indicator of redundancy in the summaries.

Case 1-**small number of clusters**. When *FC* is equal to 1, all the sentences of a collection are assigned to the same cluster. In this case,

Table 4
Parameterization results for multi-document summarization. Scores are presented for the best minimum support thresholds using different values of the *FC* parameter. The highest scores are shown in bold type.

FC	Best *min_sup*	ROUGE-2	ROUGE-SU4
1	0.14	0.2637	0.3005
2	0.16	**0.2711**	**0.3092**
3	0.16	**0.2823**	**0.3204**
4	0.15	**0.2779**	**0.3143**
5	0.15	0.2688	0.3081
6	0.15	0.2656	0.3057
7	0.16	0.2624	0.3018
8	0.17	0.2603	0.3004

the top ranked sentences within the single cluster are selected for the summary. These sentences usually share similar information, therefore, the produced summary contains redundant information while it may not cover all the main ideas. When $FC = 1$, the average number of unique concepts within a summary is 254.

Case 2-large number of clusters. When *FC* is equal to 8, the sentences are grouped into multiple clusters. In this case, the selected sentences cover a variety of topics, while some of them may not be indeed informative and related. When $FC = 8$, the average number of unique concepts within a summary is 321.

Case 3-optimum number of clusters. When FC is equal to 3, the summarizer can select sentences more efficiently than the two other cases. In this case, the average number of unique concepts within a summary is 289. This shows that, compared to the case of small number of clusters, the redundancy decreases because the summarizer selects from a wider range of topics and the number of unique concepts in summaries increases. According to the ROUGE scores, the informative content of summaries is improved because more main ideas are included. On the other hand, in comparison to the case of large number of clusters, the summaries may include more redundant information since the average number of unique concepts is smaller. This shows that for large values of *FC*, more topics are covered in summaries. However, some of the topics may not be indeed important. In this way, the informative content of summaries decreases as can be perceived by the ROUGE scores in Table 4.

Table 5 presents the ROUGE scores obtained by the CIBS method for single-document summarization, using different values of *FC*. The scores are presented only for the best value of *min_sup* specified for each *FC* value. Similar to the multi-document parameterization, the highest scores are reported when the number of final clusters is equal to 2, 3, or 4. The scores obtained for $FC = 3$ are significantly ($p < 0.05$) higher than those of other values of *FC* except 2 and 4. The discussion presented above that assessed the impact of three different cases of small, large, and optimum values of *FC* can also be applied to the single-document results. It is worth to note that, in the single-document experiments, the average number of unique concepts within summaries is 231 when *FC* takes the optimum value. When *FC* takes a small value, e.g. 1, or a large value, e.g. 8, the average number of unique concepts in summaries is 206 and 254 respectively. Regarding these numbers and the scores given by Table 5, the same trade-off between coverage and redundancy mentioned for multi-document experiments can also be applied to the single-document case. The impact of the parameter *min_sup* on the quality of summaries was discussed thoroughly in an initial work of using itemset mining for biomedical text summarization [2]. However, we give a brief discussion in the following.

Table 5
Parameterization results for single-document summarization. Scores are presented for the best minimum support thresholds using different values of the *FC* parameter. The highest scores are shown in bold type.

FC	Best *min_sup*	ROUGE-2	ROUGE-SU4
1	0.08	0.3337	0.3855
2	0.09	**0.3411**	**0.3942**
3	0.09	**0.3493**	**0.4008**
4	0.1	**0.3389**	**0.3915**
5	0.09	0.3367	0.3886
6	0.08	0.3214	0.3723
7	0.07	0.3151	0.3659
8	0.07	0.3092	0.3587

For small values of *min_sup*, e.g. 0.3 or 0.4, a large number of frequent itemsets are discovered for a document (in single-document experiments) or a collection (in multi-document experiments). In this case, the quality of summaries can be negatively affected because sentences are clustered based on many unimportant topics. On the other hand, when the itemset mining algorithm uses a large value for *min_sup*, e.g. 0.2, many important itemsets are discarded and the summarizer is not provided with sufficient knowledge about important topics. As a result, sentences cannot be clustered efficiently and the summary cannot convey the most informative and related contents.

As the results show, using optimum values for the parameters *min_sup* and *FC*, the CIBS method can produce a summary containing a wider range of important information of the input text. These two parameters can play an important role in establishing a trade-off between the information coverage and redundancy of summaries.

4.2. Multi-document evaluation

Table 6 presents the results of multi-document evaluations. The CIBS method obtains the highest scores using the three optimum values of the parameter *FC*. Using all the three settings, CIBS significantly performs better than MEAD, SUMMA, and TexLexAn for both the R-2 and R-SU4 scores ($p < 0.05$). When *FC* is equal to 3 or 4, the improvement obtained by the CIBS method is also significant for both the metrics compared to the itemset-based summarizer ($p < 0.05$).

The summaries produced by the itemset-based summarizer cover the most important ideas within collections but still contain redundant information. This is because the method always gives the highest priority to the sentences that convey the most important topics. This strategy may lead to selecting redundant sentences, and this issue is more challenging in multi-document summarization where similar information is distributed among multiple documents. In comparison to the itemset-based summarizer, the summaries generated by CIBS contain less redundant sentences and cover a wider range of topics. This is because the CIBS method groups the sentences of a collection into multiple clusters, where each cluster contains sentences sharing similar ideas. The summarizer selects sentences from all the clusters in proportion to their size. As a result, all the main topics of the collection can be covered in the summary, and the informative content of summaries improves as can be perceived by the ROUGE scores.

Compared to MEAD, SUMMA, and TexLexAn that employ generic features, the CIBS method can produce more informative and related summaries. The results show that the quality of summaries is improved when the informativeness of sentences is assessed based on their

Table 6
ROUGE scores obtained for multi-document evaluation by the CIBS method and the other summarizers. The best score for each ROUGE metric is shown in bold type.

	ROUGE-2	ROUGE-SU4
CIBS ($FC = 3$)	**0.2791**	**0.3179**
CIBS ($FC = 4$)	0.2730	0.3114
CIBS ($FC = 2$)	0.2654	0.3058
Itemset-based summarizer	0.2589	0.3002
MEAD	0.2447	0.2875
SUMMA	0.2408	0.2841
TexLexAn	0.2385	0.2793

Table 7
ROUGE scores obtained for single-document evaluation by the CIBS method and the other summarizers. The best score for each ROUGE metric is shown in bold type.

	ROUGE-2	ROUGE-SU4
CIBS (*FC* = 3)	**0.3475**	**0.3978**
CIBS (*FC* = 2)	0.3392	0.3914
CIBS (*FC* = 4)	0.3321	0.3836
Itemset-based summarizer	0.3293	0.3805
SUMMA	0.3080	0.3579
MEAD	0.3002	0.3511
TexLexAn	0.2916	0.3415

approximated semantics rather than generic measures such as the position and length. Our sentence clustering and selection approach can lead to an increase in the information coverage of summaries as demonstrated by the ROUGE scores.

4.3. Single-document evaluation

Table 7 presents the results of single-document evaluations. The CIBS method obtains the highest scores using the three optimum values of the parameter *FC*. CIBS significantly performs better than SUMMA, MEAD, and TexLexAn in terms of both the ROUGE metrics when it uses all the three settings ($p < 0.05$). Compared to the itemset-based summarizer, the improvement obtained by CIBS is significant for both R-2 and R-SU4 scores when the parameter *FC* is equal to 3 ($p < 0.05$). The improvement is only significant for R-SU4 when *FC* is equal to 2 ($p < 0.05$) and is not significant for $FC = 4$ in terms of both the metrics ($p > 0.05$).

Investigating the output of systems in single-document evaluations, we observe that summaries produced by CIBS cover a wider range of topics compared to the comparison methods. Although the redundancy in single-document summarization is not as problematic as in multi-document summarization [6], the quality of summaries generated by the itemset-based summarizer can be negatively affected by sentences containing redundant information. The itemset-based summarizer assigns higher scores to sentences containing itemsets with higher support values. Therefore, the summary may tend to cover repeated information because itemsets with smaller support values have a low chance for appearing in the summary. To deal with this problem, CIBS puts the sentences sharing similar information into the same cluster. Each cluster contributes to the summary in proportion to its significance. In this way, even the low-supporting itemsets appear in the summary and the coverage increases.

According to the results reported by the domain-independent methods in both single- and multi-document evaluations, the generic features cannot be considered as efficient measures for extracting informative sentences in this type of summarization. However, they can be more useful for summarizing specific types of inputs such as newswire articles or documents where the position of sentences may be indicative of their importance [15].

5. Conclusion

In this paper, we proposed the CIBS method, a multi-document biomedical text summarizer. Mapping the input text to the UMLS concepts in the preprocessing step, CIBS makes an approximation of the semantics behind the sentences. It employs an itemset mining method to discover the main topics within the text. Using the extracted topics and a clustering method, the summarizer groups the sentences into multiple clusters such that each cluster contains sentences sharing similar information. It generates the final summary by selecting sentences from all the clusters. Conducting a set of experiments, we evaluated the performance of the CIBS method. The results show that the topic-based clustering approach utilized by CIBS can perform better than the comparison methods. Our method assesses the informative content of sentences according to their approximated semantics. Moreover, it discovers main ideas within the text and tries to cover all the important information. As a result, the amount of redundant information in the summary can decrease and the information coverage can increase.

Regarding the parameterization results, we observed that the number of clusters can be adjusted to establish a trade-off between information coverage and redundancy. When the optimum value is assigned to the parameter *FC*, summaries cover a wider range of topics including the most important ones and other subsidiary topics. In this way, information coverage improves as shown by the ROUGE scores, and the redundant content decreases as perceived from the number of unique concepts included in summaries. As explained thoroughly in the work devoted to the itemset-based summarizer [2] and as we briefly discussed in Section 4.1, the parameter *min_sup* plays an important role in identifying the important topics within the input text, such that extreme values can lead to discovering many unimportant ideas or disregarding a set of truly important ones. These important topics affect the quality of summaries since the summarizer utilizes them to quantify the shared content between sentences and select the most related ones. We can also observe that generic measures employed by the general-purpose and publicly available summarizers may not relatively provide an acceptable performance in biomedical text summarization. However, they have shown their suitability for summarizing other genres of text [15].

The major limitation of this work is related to the evaluation corpora. Although the approaches to creating the corpora for experiments were also adopted by previous work in the field, the model summaries used in both the single- and multi-document experiments might not appropriately convey all the important content within the original texts. It is truly needed to develop standard corpora for both single- and multi-document biomedical text summarization, such that methods can be evaluated using a benchmark.

This research may be extended further to deal with texts from other domains and related challenges. The ever-increasing volume of textual information in the biomedical domain led us to developing a domain-specific text summarizer, taking into account that the text analysis should be done at a semantic level to achieve a better summarization performance. The challenge of establishing a trade-off between coverage and redundancy had not been addressed in the context of multi-document biomedical summarization. This was our motivation for developing the CIBS method and evaluating it against other domain-specific and general-purpose summarizers. However, there is much work to do in the field. Such summarization methods can be specialized further by considering the singularities of other types of documents such as medical records, clinical notes, clinical trial reports, multimedia documents, web documents, and so on. It can also have potential for future research to develop methods and measures that assess the informativeness of sentences based on some task-specific criteria. For example, if the clinical relevance of sentences is an indicator of their importance in a specific summarization task, a method can be developed to assess the informative content in terms of clinical relevance. It should be taken into account that the above mentioned types of work require an extensive effort to construct standard benchmark corpora to have an accurate performance evaluation.

In a previous work [5], we addressed the probabilistic modeling of single-document summarization based on the idea stating that if the probability distribution of important concepts within the summary follows the distribution of those concepts within the original text, the quality of summaries can improve. Future work may include adopting such type of probabilistic modeling for multi-document summarization. However, there are some challenges that need to be taken into account. First, establishing a trade-off between information coverage and redundancy would be a challenging task. Second, every document in a

collection may have a different probability distribution in terms of important concepts. The concepts also have an overall distribution across the whole collection. Regarding these various distributions, more complex statistical and probabilistic methods are needed for extending the single-document summarization to the multi-document counterpart. Furthermore, it may be useful to develop new methods and measures to identify essential concepts within a collection of documents.

6. Mode of availability

The Java source code of the CIBS method is accessible for research purposes at http://github.com/mmoradi-iut/CIBS-biomedical-text-summarizer.

References

[1] M. Gambhir, V. Gupta, Recent automatic text summarization techniques: a survey, Artif. Intell. Rev. 47 (2016) 1–66.
[2] M. Moradi, N. Ghadiri, Quantifying the informativeness for biomedical literature summarization: An itemset mining method, Comput. Methods Programs Biomed. 146 (2017) 77–89.
[3] R. Mishra, J. Bian, M. Fiszman, C.R. Weir, S. Jonnalagadda, J. Mostafa, et al., Text summarization in the biomedical domain: a systematic review of recent research, J. Biomed. Inform. 52 (2014) 457–467.
[4] L. Plaza, A. Díaz, P. Gervás, A semantic graph-based approach to biomedical summarisation, Artif. Intell. Med. 53 (2011) 1–14.
[5] M. Moradi, N. Ghadiri, Different approaches for identifying important concepts in probabilistic biomedical text summarization, Artif. Intell. Med. 84 (2018) 101–116.
[6] R. Ferreira, L. de Souza Cabral, F. Freitas, R.D. Lins, G. de França Silva, S.J. Simske, et al., A multi-document summarization system based on statistics and linguistic treatment, Expert Syst. Appl. 41 (2014) 5780–5787.
[7] S.J. Nelson, T. Powell, B. Humphreys, The unified medical language system (umls) project, Encyclopedia Library Inform. Sci. (2002) 369–378.
[8] C.-Y. Lin, Rouge: A package for automatic evaluation of summaries, Text summarization branches out: Proceedings of the ACL-04 workshop, 2004.
[9] J.G. Yao, X. Wan, J. Xiao, Recent advances in document summarization, Knowl. Inf. Syst. (2017) 1–40.
[10] Y. Sankarasubramaniam, K. Ramanathan, S. Ghosh, Text summarization using Wikipedia, Inf. Process. Manage. 50 (2014) 443–461.
[11] R.M. Alguliev, R.M. Aliguliyev, N.R. Isazade, DESAMC+DocSum: Differential evolution with self-adaptive mutation and crossover parameters for multi-document summarization, Knowl.-Based Syst. 36 (2012) 21–38.
[12] H. Saggion, SUMMA: A robust and adaptable summarization tool, Traitement Automatique des Langues 49 (2008).
[13] D.R. Radev, H. Jing, M. Styś, D. Tam, Centroid-based summarization of multiple documents, Inf. Process. Manage. 40 (2004) 919–938.
[14] (Accessed 01.10.2017). TexLexAn: An Open Source Text Summarizer. < http://texlexan.sourceforge.net/ > .
[15] V. Gupta, G.S. Lehal, A survey of text summarization extractive techniques, Journal of Emerging Technologies in Web Intelligence 2 (2010) 258–268.
[16] H.D. Menéndez, L. Plaza, D. Camacho, Combining graph connectivity and genetic clustering to improve biomedical summarization, 2014 IEEE Congress on Evolutionary Computation (CEC), 2014, pp. 2740–2747.
[17] L.H. Reeve, H. Han, A.D. Brooks, The use of domain-specific concepts in biomedical text summarization, Inf. Process. Manage. 43 (2007) 1765–1776.
[18] Z. Lu, K.B. Cohen, L. Hunter, Finding GeneRIFs via gene ontology annotations, Biocomputing 2006, World Scientific, 2006, pp. 52–63.
[19] Z. Lu, K. BRETONNEL COHEN, L. Hunter, GeneRIF quality assurance as summary revision, *Biocomputing 2007*, World Scientific, 2007, pp. 269–280.
[20] N. Elhadad, M.Y. Kan, J.L. Klavans, K.R. McKeown, Customization in a unified framework for summarizing medical literature, Artif. Intell. Med. 33 (2005) 179–198 2005/02/01/.
[21] M. Fiszman, D. Demner-Fushman, H. Kilicoglu, T.C. Rindflesch, Automatic summarization of MEDLINE citations for evidence-based medical treatment: A topic-oriented evaluation, J. Biomed. Inform. 42 (2009) 801–813 2009/10/01/.
[22] Y. Cao, F. Liu, P. Simpson, L. Antieau, A. Bennett, J.J. Cimino, et al., AskHERMES: An online question answering system for complex clinical questions, J. Biomed. Inform. 44 (2011) 277–288 2011/04/01/.
[23] L. Plaza, J. Carrillo-de-Albornoz, Evaluating the use of different positional strategies for sentence selection in biomedical literature summarization, BMC Bioinf. 14 (2013) 1.
[24] R. Pivovarov, N. Elhadad, Automated methods for the summarization of electronic health records, J. Am. Med. Inform. Assoc. 22 (2015) 938–947.
[25] H. Moen, L.-M. Peltonen, J. Heimonen, A. Airola, T. Pahikkala, T. Salakoski, et al., Comparison of automatic summarisation methods for clinical free text notes, Artif. Intell. Med. 67 (2016) 25–37.
[26] (Accessed 01.10.2017). National Library of Medicine. UMLS Specialist Lexicon fact sheet. < http://www.nlm.nih.gov/pubs/factsheets/umlslex.html ≥ .
[27] (Accessed 01.10.2017). National Library of Medicine. UMLS Metathesaurus fact sheet. < http://www.nlm.nih.gov/pubs/factsheets/umlsmeta.html ≥ .
[28] (Accessed 01.10.2017). National Library of Medicine. UMLS Semantic Network fact sheet. < http://www.nlm.nih.gov/pubs/factsheets/umlssemn.html ≥ .
[29] M. Mampaey, N. Tatti, J. Vreeken, Tell me what i need to know: succinctly summarizing data with itemsets, Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining, 2011, pp. 573–581.
[30] C. Ordonez, N. Ezquerra, C.A. Santana, Constraining and summarizing association rules in medical data, Knowl. Inf. Syst. 9 (2006) 1–2.
[31] E. Baralis, L. Cagliero, A. Fiori, P. Garza, Mwi-sum: A multilingual summarizer based on frequent weighted itemsets, ACM Transactions on Information Systems (TOIS) 34 (2015) 5.
[32] (Accessed 01.10.2017). National Library of Medicine. MetaMap Portal. < http://mmtx.nlm.nih.gov/ > .
[33] L. Plaza, M. Stevenson, A. Díaz, Resolving ambiguity in biomedical text to improve summarization, Inf. Process. Manage. 48 (2012) 755–766.
[34] D.T. Larose, Discovering knowledge in data: an introduction to data mining, John Wiley & Sons, 2014.
[35] R. Agrawal, T. Imieliński, A. Swami, Mining association rules between sets of items in large databases, ACM SIGMOD Record 22 (1993) 207–216.
[36] R. Agrawal, H. Mannila, R. Srikant, H. Toivonen, A.I. Verkamo, Fast discovery of association rules, Adv. Knowledge Discovery Data Min.ng 12 (1996) 307–328.
[37] Y. Shang, Y. Li, H. Lin, Z. Yang, Enhancing biomedical text summarization using semantic relation extraction, PLoS one 6 (2011) e23862.
[38] C.-Y. Lin, Looking for a few good metrics: Automatic summarization evaluation-how many samples are enough? NTCIR (2004).
[39] (Accessed 01.03.2017). BioMed Central's open access full-text corpus for text mining research. < http://old.biomedcentral.com/about/datamining > .
[40] M. Nasr Azadani, N. Ghadiri, E. Davoodijam, Graph-based biomedical text summarization: an itemset mining and sentence clustering approach, J. Biomed. Inform. 84 (2018) 42–58, https://doi.org/10.1016/j.jbi.2018.06.005.
[41] M. Nasr Azadani, N. Ghadiri, Evaluating different similarity measures for automatic biomedical text summarization, in: International Conference on Intelligent Systems Design and Applications, Springer, Cham, 2017, pp. 305–314, DOI: https://dx.doi.org/10.1007/978-3-319-76348-4_30.