CitationAS: A Summary Generation Tool Based on Clustering of Retrieved Citation Content

Jie Wang[1], Shutian Ma[1], Chengzhi Zhang[1,2, *]

wangjie1342@qq.com , mashutian0608@hotmail.com, zhangcz@njust.edu.cn
[1] Department of Information Management, Nanjing University of Science and Technology, Nanjing, China, 210094
[2] Fujian Provincial Key Laboratory of Information Processing and Intelligent Control (Minjiang University), Fuzhou, China, 350108

Abstract

Usually, if researchers want to understand research status of any field, they need to browse a great number of related academic literatures. Luckily, in order to work more efficiently, automatic documents summarization can be applied for taking a glance at specific scientific topics. In this paper, we focus on summary generation of citation content. An automatic tool named CitationAS is built, whose three core components are clustering algorithms, label generation and important sentences extraction methods. In experiments, we use bisecting K-means, Lingo and STC to cluster retrieved citation content. Then Word2Vec, WordNet and combination of them are applied to generate cluster label. Next, we employ two methods, TF-IDF and MMR, to extract important sentences, which are used to generate summaries. Finally, we adopt gold standard to evaluate summaries obtained from CitationAS. According to evaluations, we find the best label generation method for each clustering algorithm. We also discover that combination of Word2Vec and WordNet doesn't have good performance compared with using them separately on three clustering algorithms. Combination of Ling algorithm, Word2Vec label generation method and TF-IDF sentences extraction approach will acquire the highest summary quality.

Conference Topic

Text mining and information extraction

Introduction

Currently, quantity of electronic academic literatures has reached a massive level. Challenges have shown up when people want to investigate research status quo in a field (Liu, 2013): (1) When searching in academic databases (e.g., CNKI[1]) or search engines (e.g., Google Scholar[2]), users are often given the relevant and ranked results which include many redundant information in themselves or among different platforms. (2) Although manual literature summaries can help researchers learn quickly about a new field, such summaries are in a small amount and their formation cycle is long which will definitely lead to hysteresis. Therefore, tools and systems are urgently needed to automatically generate a comprehensive, detailed and accurate summary according to the given topic words (Nenkova & McKeown, 2011). At the same time, such tools and systems should also help researchers retrieve relevant information in real time.

Obviously, the automatic summary tool can deal with problems mentioned above. When applying such tools, how to choose data for summary generation is another challenge. Firstly, if all literature contents are used to generate a summary, system cost will be increased and unimportant and redundant contents might be added. Secondly, if we only use abstracts for summary generation, there will be information loss compared with using full text. Hence, citation content can be chosen as dataset and the main reasons include: (1) Citation content is not only consistent with original abstract, but also can provide more concepts, such as entities and experimental methods (Divoli, Nakov & Hearst, 2012), and even retain some original

[*] Corresponding author: Chengzhi Zhang.

[1] http://www.cnki.net

[2] http://scholar.google.com/

information from cited articles. (2) Since citation content reflects author's analysis and summarization of other articles, it has objectivity and diversity (Elkiss, Shen, Fader, States & Radev, 2008). Some researchers have applied citation content to generate summaries. For example, Tandon and Jain (2012) generated structured summary by classifying citation content into one or more classes. Cohan and Goharian (2015) grouped citation content and its context at first, and then ranked sentences within each group, finally sentences were selected for summary. Yang et al. (2016) utilized key phrase, spectral clustering and ILP optimization framework to generate summary.

In this paper, we use citation content to do automatic documents summarization, and apply clustering algorithms to build an automatic summary generation tool, named CitationAS[3]. The main works include: (1) We build a demonstration website which can automatically generate summary under a given topic; (2) We optimize a search results clustering engine, Carrot2[4] (Osiński & Weiss, 2005), in three aspects, similar cluster label merging, important sentences extraction and summary generation.

Summary Generation Tool

Dataset

In this paper, we collected about 110, 000 articles in xml format from PLOS One[5] between 2006 and 2015, covering subjects such as cell biology, chemistry, mental health, computer science and so on. We identified citation sentences by rules, which discriminated whether a sentence contains reference marks (e.g., "*[1]*", "*[2]-[4]*") or not, and then xml labels were removed. 4, 339, 217 citation sentences were extracted to be used as citation content for automatic summary generation. Table 1 displays citation sentence examples.

Table 1. Citation Sentence Examples

No.	Citation sentence
1	*Gelatin zymography was performed as described previously [27].*
2	*Two studies in Drosophila subobscura found considerable differences [22], [42].*
3	*Even by knockout of a single VEGF-A allele mice were unable to survive [5]–[7].*
4	*The PCP signaling pathway determines planar polarity in a variety of tissues[4], [7]–[8].*

Framework of CitationAS

Framework of CitationAS is shown in Figure 1. Firstly, relevant citation sentences are retrieved from index files according to search terms from user interface. Then, we apply clustering algorithms to classify sentences into clusters which share same or similar topic. After that, we merge clusters whose labels are more similar with each other. Finally, summary is generated based on important sentences extracted from each cluster. And final evaluation is carried out by volunteers.

[3] http://117.89.118.178:8001/CitationAS/

[4] http://project.carrot2.org/

[5] http://journals.plos.org/plosone/

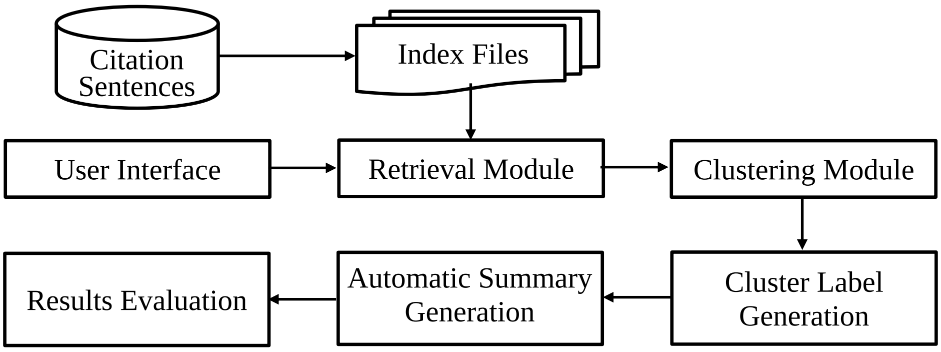


Figure 1. Framework of CitationAS

Retrieval module

We use Lucene[6] to index and retrieve dataset. When establishing index files, we add citation sentences and structure information (e.g., doi, cited count, position of one sentence and its first word in original article and paragraph). Our system also applies built-in algorithms of Lucene to obtain citation sentences associated with search terms and score sentences based on relevance. Finally, CitationAS ranks results which are used for the next step of clustering.

Clustering module

In this module, we firstly apply VSM (Yang & Pedersen, 1997) to represent citation sentences and use TF-IDF (Salton & Yu, 1973) to calculate feature weights. In VSM, each citation sentence is equivalent to a document and expressed as $s_j = s_j$ (t_1,w_{1j};…t_i,w_{ij}…;t_m,w_{mj}), where t_i is the i^{th} feature item, w_{ij} is feature weight of t_i in the j^{th} sentence, meanwhile, $1 \leq i \leq m$, $1 \leq j \leq N$, m and N are the number of feature item and citation sentences. The formula of TF-IDF is shown as (1).

$$w_{ij} = tf_{ij} * idf_i = tf_{ij} * \log(\frac{N}{n_i} + 0.01) \qquad (1)$$

Where tf_{ij} is frequency of t_i in sentence s_j, and n_i represents the number of sentences in which t_i is located.

Next, bisecting K-means, Lingo and STC, built-in Carrot2, are used to cluster citation sentence respectively. Since VSM will represent documents in a high dimension, which will cost efficiency of clustering algorithms, we adopt NMF algorithm (Lee, 2000) to reduce dimensions. This algorithm obtains the non-negative matrix after decomposing the term-document matrix. It can be described as that for non-negative matrix A_{m*n}, we need to find non-negative matrix U_{m*r} and V_{r*n}, which should satisfy the following formula:

$$A_{m*n} \approx U_{m*r} \times V_{r*n} \qquad (2)$$

Where U_{m*r} is the base matrix, V_{r*n} is the coefficient matrix, and r is the number of new feature item. When r is less than m, we can replace A_{m*n} with V_{r*n} to achieve dimensionality reduction.

In bisecting K-means, we will use coefficient matrix to calculate similarity between citation sentence and clustering centroid. Each sentence is assigned to the most similar cluster. Labels of each cluster are individual words which are three feature items with the greatest weight in term-document matrix.

Lingo algorithm firstly extracts key phrases by the suffix sorting array and the longest common prefix array. Then it builds term-phrase matrix based on the key phrases, where feature weight is calculated by TF-IDF. Thirdly, it constructs base vectors according to the

[6] http://lucene.apache.org/

term-phrase matrix and the base matrix through NMF. Finally, each base vector gets corresponding words or phrases to form one cluster label, and sentence containing label’s words will be assigned to the corresponding cluster.

STC algorithm (Zamir & Etzioni, 1999) is based on Generalized Suffix Tree which recognizes key words and phrases that occurred more than once in citation sentences. Then each such words and phrases are used to come into being one base clusters. There may be many same citation sentences in two clusters, while the cluster labels are different. So, we merge these base clusters to form final clusters in order to reduce overlap rate of citation sentences between clusters.

Among the three algorithms, Lingo and STC have two common characteristics. They both create overlapping clusters that means one document can be assigned to more than one cluster. Besides, their cluster labels may appear phrases. While bisecting K-means is non-overlapping clustering algorithm, and words included in the generated cluster labels may not correspond with all cluster’s documents.

Cluster label generation

It is possible that some cluster labels are semantic similar to each other, for example, labels like ‘*data mining method*’ and ‘*data mining approach*’ for the search terms ‘*data mining*’. In order to improve experimental accuracy, similar cluster labels are merged in experiments. We apply three methods to calculate semantic similarity between labels by using Word2Vec (Mikolov, Le & Sutskever, 2013) and WordNet (Fellbaum & Miller, 1998).

(1) Similarity Computation Based on Word2Vec

Word2Vec is a statistical language model based on corpus. It applies neural network to get word vectors, which can be used to compute similarity between words. Given phrase P, we assume that it is made up of word A, B and C. Then we can get the i^{th} dimensional representation in the phrase P, namely $\frac{1}{L}\sum_{i=1}^{Len}(a_i + b_i + c_i)$, where L means the number of words in P. Finally, we use cosine value to compute similarity between phrases. The formula is shown as (3).

$$sim(p_1, p_2) = \frac{\sum_{i=1}^{n} p_{1i} \times p_{2i}}{\sqrt{\sum_{i=1}^{n} p_{1i}^2} \times \sqrt{\sum_{i=1}^{n} p_{1i}^2}} \qquad (3)$$

(2) Similarity Computation Based on WordNet

WordNet is a semantic dictionary and organizes words in a classification tree, so semantic similarity between words can be calculated by path in the tree. The formula is shown as (4).

$$sim(w_1, w_2) = 1/distance(w_1, w_2) \qquad (4)$$

Where $distance(w_1, w_2)$ denotes the shortest path between words in the tree.

Then, similarity between phrases uses formula (5) to calculate.

$$sim(p_1, p_2) = \sum_{i=1}^{L_{p_1}} \sum_{j=1}^{L_{p_2}} \frac{sim(p_{1i}, p_{2j})}{L_{p_1} \times L_{p_2}} \qquad (5)$$

Where p_1 and p_2 represents phrases, L_{p_1} and L_{p_2} means the number of words in phrases, $sim(p_{1i}, p_{2j})$, calculated via formula (4), means the similarity between words in p_1 and p_2.

(3) Similarity Computation Based on Combination of Word2Vec and WordNet

We linearly combine Word2Vec and WordNet to obtain a new similarity calculation method. The formula is shown as (6), where α is a weight and we set it to be 0.5.

$$sim(p_1, p_2) = \alpha sim_{word2vec}(p_1, p_2) + (1 - \alpha) sim_{wordnet}(p_1, p_2) \qquad (6)$$

Automatic summary generation

Clusters are sorted according to their size and each cluster is taken as a paragraph in the final summary. To choose important citation sentences from each cluster, we design two methods to measure sentence scores.

(1) TF-IDF based Sentences Extraction

Since each citation sentence is represented by the term-document matrix, we can obtain the sentence weight. For the sentence $s = s\ (t_1, w_1; \ldots t_i, w_i \ldots; t_m, w_m)$, its weight is computed via the following formula (7):

$$w_S = \sum_{i=1}^{m} w_i\ /m \qquad (7)$$

Thereby, we rank citation sentences in each cluster based on its weight. The sentences with higher weight will be used as summary sentences.

(2) MMR based Sentences Extraction

MMR (Carbonell, Jaime & Goldstein, 1998) method considers similarity of selected sentence to search items and redundancy to sentences in summary.

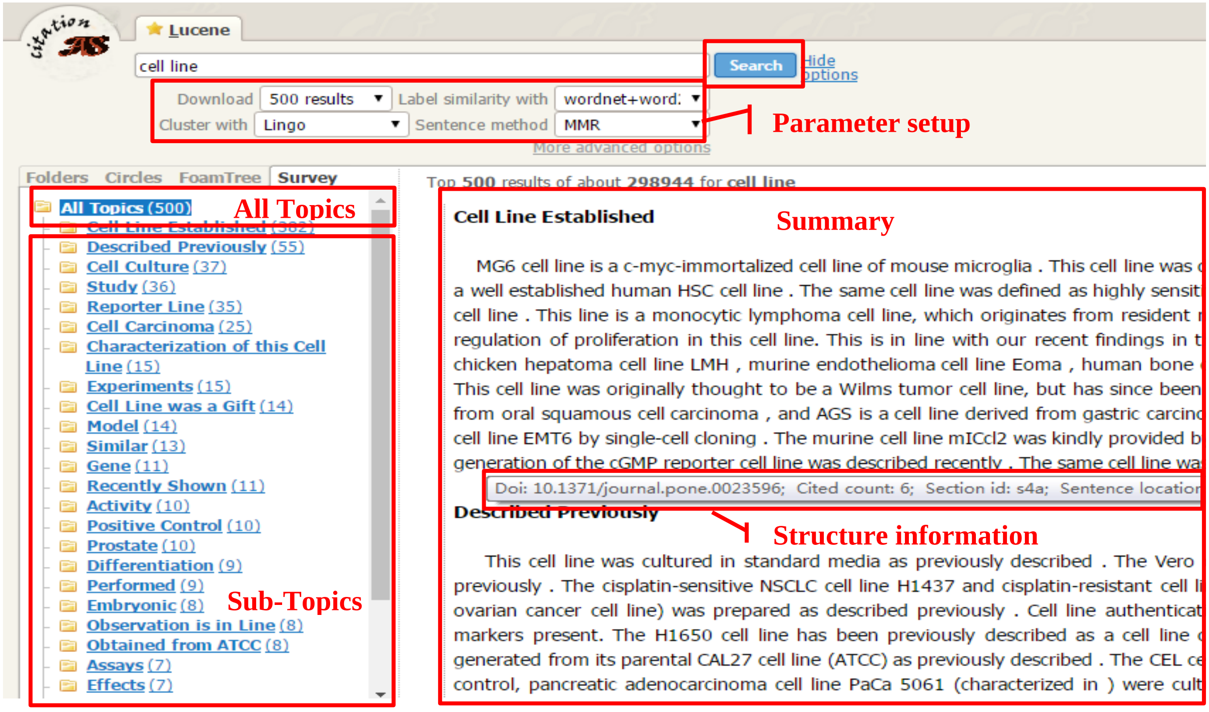
$$s = \max_{s_i \in C - S}\left[\beta sim(s_i, q) - (1-\beta) \max_{s_j \in S} sim(s_i, s_j)\right] \qquad (8)$$

Where C denotes the set of citation sentences in cluster, S denotes the set of summary sentences, so $s_i \in C - S$ denotes the set of not selected as summary sentences. s_i means current citation sentence and q means search items. β is a parameter and generally set it to be 0.7.

This method firstly selects maximum score of sentence as a summary sentence from the candidate sentence set, then it recalculates MMR value of the left sentences. When the candidate sentence set is empty, this algorithm ends.

User interface of CitationAS

As shown in Figure 2, users can input search terms and set parameters to get a summary. The parameters (‘*Parameter setup*’ scope) are about summary generation methods and the number of citation sentences for clustering. When users click ‘*search*’, sub-topics, which are cluster labels and the number, will appear in the summary frame. Then users can click ‘*All Topics*’, the automatic summary will be presented on the right side, where the bold fonts are titles and others are content in summary’s paragraph. Summary sentence’s structure information will be displayed, when users put the mouse on it.



cell line

Download 500 results ▾ Label similarity with wordnet+word: ▾
Cluster with Lingo ▾ Sentence method MMR ▾

Top 500 results of about 298944 for cell line

- All Topics (500)
- Cell Line Established (302)
- Described Previously (55)
- Cell Culture (37)
- Study (36)
- Reporter Line (35)
- Cell Carcinoma (25)
- Characterization of this Cell Line (15)
- Experiments (15)
- Cell Line was a Gift (14)
- Model (14)
- Similar (13)
- Gene (11)
- Recently Shown (11)
- Activity (10)
- Positive Control (10)
- Prostate (10)
- Differentiation (9)
- Performed (9)
- Embryonic (8)
- Observation is in Line (8)
- Obtained from ATCC (8)
- Assays (7)
- Effects (7)

Cell Line Established

MG6 cell line is a c-myc-immortalized cell line of mouse microglia . This cell line was
a well established human HSC cell line . The same cell line was defined as highly sensit
cell line . This line is a monocytic lymphoma cell line, which originates from resident
regulation of proliferation in this cell line. This is in line with our recent findings in t
chicken hepatoma cell line LMH , murine endothelioma cell line Eoma , human bone
This cell line was originally thought to be a Wilms tumor cell line, but has since been
from oral squamous cell carcinoma , and AGS is a cell line derived from gastric carcin
cell line EMT6 by single-cell cloning . The murine cell line mICcl2 was kindly provided b
generation of the cGMP reporter cell line was described recently . The same cell line wa

Doi: 10.1371/journal.pone.0023596; Cited count: 6; Section id: s4a; Sentence locatio

Described Previously

This cell line was cultured in standard media as previously described . The Vero
previously . The cisplatin-sensitive NSCLC cell line H1437 and cisplatin-resistant cell li
ovarian cancer cell line) was prepared as described previously . Cell line authenticat
markers present. The H1650 cell line has been previously described as a cell line
generated from its parental CAL27 cell line (ATCC) as previously described . The CEL ce
control, pancreatic adenocarcinoma cell line PaCa 5061 (characterized in) were cult

Figure 2. User Interface of CitationAS

Experimental Results Analysis and Discussion

Since the summary is based on user's search terms in CitationAS, we choose 20 high-frequency phrases from dataset as search terms and use them for experiments. Phrases are shown in Table 2. Here, the frequency refers to the number of phrases presenting in citation content dataset. We divide them into ten high frequency 2-gram and 3-gram separately. We also find that phrases are related to medical field, this is because articles about biology and mental health have a large proportion.

In the cluster label generation test, we apply Davies-Bouldin (DB) and SC clustering index (Fahad et al., 2014) to find the best label generation method for each clustering algorithm. SC index is equal to the ratio of clusters' separation and compactness. If DB value is lower and SC value is higher, clusters are more compact and further from each other. The more number of search terms for consistency between DB and SC, the better clustering results obtained by the method will be. Through experiments, we find combination of Lingo and Word2Vec has better clustering results with 8 search terms. When combining STC with WordNet, there are 6 search terms. If combining bisecting K-means with Word2Vec, we find a total of 9 search terms. However, combination of Word2Vec and WordNet doesn't have good performance compared with applying them separately on the three clustering algorithms. The quality of some cluster results based on this method is between WordNet and Word2Vec. The reason may be that we only use linear function and set equal weights to combine them, which is too simple to bring out their strengths. In a word, we use these methods to carry out the final automatic summary generation experiment.

Table 2. Top 20 Phrases According to High Frequency

Phrase (Frequency）	Phrase (Frequency）
cell line (37507)	reactive oxygen species (5160)
gene expression (37001)	central nervous system (4418)
amino acid (35165)	smooth muscle cell (3439)
transcription factor (25626)	protein protein interaction (3286)
cancer cell (25605)	single nucleotide polymorphism (2535)
stem cell (22567)	tumor necrosis factor (2482)
growth factor (17531)	genome wide association (2386)
signaling pathway (16597)	case control study (2269)
cell proliferation (14203)	false discovery rate (2209)
meta analysis (12647)	innate immune response (2133)

In this paper, we choose 20 search terms and each of them generates summaries in 6 different approaches. Finally, 120 summaries are produced. Compression ratio is set to be 20%, which means the final summary length equals the number of retrieved citation sentences multiplies by 20%. Then we invite 2 volunteers to make manual evaluation and apply 5 points system to score. The evaluation standards are described in Table 3.

In the evaluation process, we give volunteers 120 produced summaries and the corresponding search words for each summary, but we do not let them know the generated method behind each summary. Volunteers are demanded to mark each paragraph in the summary, thus we can get average score of each summary. Since each summary is obtained by one method, we can calculate average score of each method. In order to sketch the selected summary generation approaches, we omit Word2Vec and Wordnet in the Table 4 and Figure 3. For example, method Lingo-Word2Vec-TF-IDF will be described as Lingo-TF-IDF.

Table 3. Evaluation Standards

Score	Evaluation standards
5	Sentences are very smooth. Paragraphs and summaries are very comprehensive, exist very small redundancy and can fully reflect retrieval topics. The logical structure of summary is reasonable.
4	Sentences are relatively smooth. Paragraphs and summaries are relatively comprehensive, exist relatively small redundancy and can relatively reflect retrieval topics. The logical structure of summary is relatively reasonable.
3	Sentences are basically smooth. Paragraphs and summaries are basically comprehensive, exist certain redundancy and can basically reflect retrieval topics. The logical structure of summary is basically reasonable.
2	Sentences are not smooth enough. Paragraphs and summaries are not comprehensive, exist relatively high redundancy and cannot reflect retrieval topics enough. The logical structure of summary is confusing.
1	The smoothness of sentences becomes very poor. Paragraphs and summaries are far from comprehensive, exist very high redundancy and cannot fully reflect retrieval topics. There is no logical structure in the summary.

Table 4. Six Methods Rankings Based on Two Volunteers

Ranking	Volunteer A	Volunteer B
1	STC-TF-IDF	Lingo-TF-IDF
2	Lingo-TF-IDF	Lingo-MMR
3	STC-MMR	STC-MMR
4	Lingo-MMR	STC-TF-IDF
5	bisecting K-means-MMR	bisecting K-means-MMR
6	bisecting K-means-TF-IDF	bisecting K-means-TF-IDF

We rank the six methods according to average score of each method shown in Table 4. We can find that rankings of STC-WordNet-MMR, bisecting K-means-Word2Vec-TF-IDF and bisecting K-means-Word2Vec-MMR are same in two volunteers scores. They both think summary quality is poor by bisecting K-means algorithm, especially the combination of bisecting K-means, Word2Vec and TF-IDF. Reasons of this phenomenon may be that bisecting K-means is hard clustering and each sentence must belong to one cluster. Some sentences in same cluster may not be subject to the cluster's topic. And cluster labels may also not effectively reflect the topic of citation sentences in cluster. Volunteers give different rankings for the rest of methods, which indicates each of these approaches has its own advantages and disadvantages.

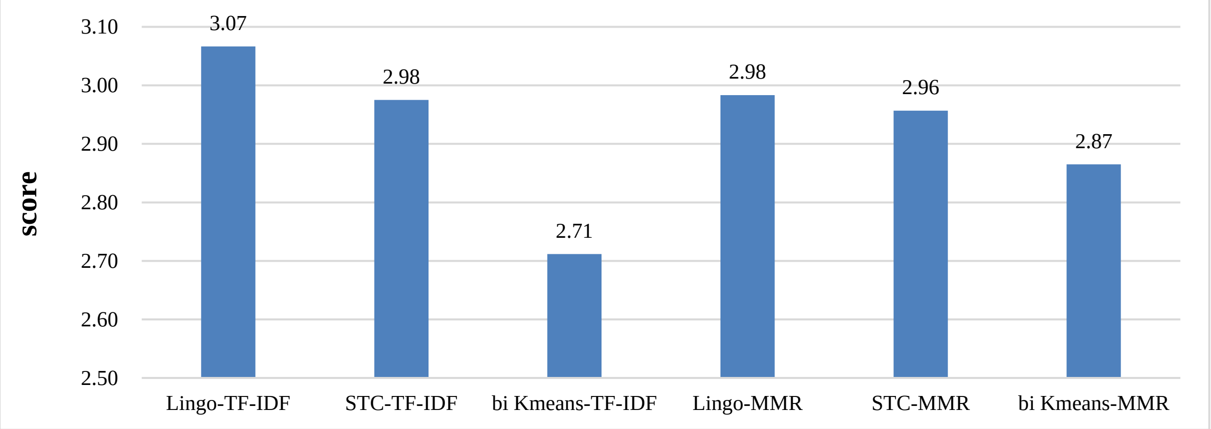


Figure 3. Average Scores of Six Different Methods

In order to make a comprehensive analysis about six methods, we average the scores of two volunteers. As illustrated in Figure 3, scores obtained by 6 methods are close to 3, indicating the generated summaries are comprehensive. Among them, combination of Lingo, Word2Vec and TF-IDF acquires the highest summary quality which is 3.07. When it comes to TF-IDF or MMR, summary quality obtained based on combination of Lingo and Word2Vec is higher. The reason may be that Lingo algorithm uses abstract matrix and the longest common prefix array when obtaining clustering labels, so that it can get more meaningful labels. In addition, citation sentence is assigned to the cluster containing corresponding labels, instead of calculating similarity between sentence and cluster centroid. This may be one of the reasons for using TF-IDF method to get a better summary. Compared to TF-IDF, we also find that summary quality is higher based on MMR after using combination of bisecting K-means and Word2Vec. Bisecting K-means algorithm divides citation sentences according to similarity between cluster centroid and sentences. Meanwhile, MMR also considers similarity between citation sentences. However, TF-IDF ranks sentences only by their weight. Summary quality obtained by combination of STC, WordNet and TF-IDF or MMR is almost same, which indicates that sentences selection approaches do not have much impact on summary quality based on this clustering algorithm.

Conclusions

In this paper, we establish an automatic summary generation tool, named CitationAS. Our tool mainly contains three components. The first is clustering algorithms including bisecting K-means, Lingo and STC. The second is cluster label generation methods, Word2Vec, WordNet and the combination of them. The last is automatic summary generation approaches which are TF-IDF and MMR. Citation sentences are applied as summary generation data. Through experiments, we choose the best label generation approach for each clustering algorithm from semantic level, and then they are used in automatic summary generation. We find that combination of Word2Vec and WordNet doesn’t improve system performance compared with using them separately. Finally, automatic summary obtained by 6 methods are comprehensive, which means that sentences are basic smooth, summary content is basic comprehensive and reflects the retrieval topic, but it has redundancy. For soft cluster, such as Lingo and STC, quality of summary obtained by TF-IDF may be better. The generated summary by CitationAS may not completely reflect the topic, but people can refer to it.

In future work, we will apply Ontology to calculate semantic similarity between labels and use deep learning to improve quality of generative summary. We will also select new approach to combine WordNet and Word2Vec in order to play their advantages. Besides, automatic evaluation can be made to avoid wrong judgements by human.

Acknowledgments

This work is supported by Major Projects of National Social Science Fund (No. 16ZAD224), Fujian Provincial Key Laboratory of Information Processing and Intelligent Control (Minjiang University) (No. MJUKF201704) and Qing Lan Project.

References

Cohan, A., & Goharian, N. (2015). Scientific article summarization using citation-context and article's discourse structure. *Conference on Empirical Methods in Natural Language Processing*, 390-400.

Carbonell, Jaime, & Goldstein. (1998). The use of MMR, diversity-based reranking for reordering documents and producing summaries. *In Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, 335-336.

Divoli, A., Nakov, P., & Hearst, M. A. (2012). Do Peers See More in a Paper Than Its Authors? *Advances in Bioinformatics*, 2012(2012), 750214.

Elkiss, A., Shen, S., Fader, A., States, D., & Radev, D. (2008). Blind Men and Elephants: What do Citation Summaries Tell Us about a Research Article? *Journal of the American Society for Information Science and Technology,* 59(1), 51-62.

Fahad, A., Alshatri, N., Tari, Z., Alamri, A., Khalil, I., Zomaya, A. Y., Foufou, S. & Bouras, A. (2014). A survey of clustering algorithms for big data: taxonomy and empirical analysis. *IEEE Transactions on Emerging Topics in Computing,* 2(3), 267-279.

Fellbaum, C., & Miller, G. (1998). WordNet: An Electronic Lexical Database. *MIT Press.*

Lee, D. D. (2000). Algorithms for nonnegative matrix factorization. *Advances in Neural Information Processing Systems,* 13(6), 556-562.

Liu, X. (2013). Generating metadata for cyberlearning resources through information retrieval and meta-search. *Journal of the American Society for Information Science and Technology,* 64(4): 771-786.

Mikolov, T., Le, Q. V., & Sutskever, I. (2013). Exploiting Similarities among Languages for Machine Translation. *Computer Science,* 1-10.

Nenkova, A., & McKeown, K. (2011). Automatic summarization. *Meeting of the Association for Computational Linguistics*, 5(3), 1-42.

Osiński, S., & Weiss, D. (2005). Carrot2: Design of a Flexible and Efficient Web Information Retrieval Framework. *In Proceedings of the Third International Atlantic Web Intelligence Conference,* 439-444.

Salton, G., & Yu, C. T. (1973). On the Construction of Effective Vocabularies for Information Retrieval. *Acm Sigplan Notices*, 10(1), 48-60.

Tandon, N., & Jain, A. (2012). Citation context sentiment analysis for structured summarization of research papers. *In Proceedings of 35th German Conference on Artificial Intelligence*, 1-5.

Yang, S., Lu, W., Yang, D., Li, X., Wu, C., & Wei, B. (2016). KeyphraseDS: Automatic generation of survey by exploiting keyphrase information. *Neurocomputing*, 224, 58-70.

Yang, Y., & Pedersen, J. O. (1997). A Comparative Study on Feature Selection in Text Categorization. *In Proceedings of the 14th International Conference on Machine Learning,* 412-420.

Zamir O., & Etzioni O. (1999). Grouper: A dynamic clustering interface to Web search results. *Computer Networks*, 31(11), 1361-1374.