International journal of scientific and technical research in engineering (IJSTRE)
ISSN: 2581-9941
www.ijstre.com Volume 6 Issue 1 ǁ 2021.

Single Document Viewpoint Summarization based on Triangle Identification in Dependency Graph

Yanting Li[1], Lianxin Xie[2], Linglong Chen[3]
[1, 2, 3]*(School of Information Engineering, Shaoguan University, P.R. China)*

ABSTRACT: *The task of viewpoint summarization is to extract a set of keywords or key phrases from a given machine readable document so that the main idea of the document can be automatically summarized. Most of the well-known viewpoint summarization approaches utilizes supervised learning technique with a collection of corpus as training data. In this paper, we propose a novel algorithm, called "TriangleIdent" for key sentence extraction from single document. A dependency graph is built for the given electronic document based on co-occurrence relations as well as syntactic dependency relations among words of the given document. The set of nodes represents words or phrases with high frequency. The set of edges represents dependency relations. The set of key-sentences can be highlighted in each paragraph, and extracted for summarizing the document based on counting the number of triangles in a sentence. For measuring the strength of connectivity, the clustering coefficient is employed as a metric due to the characteristics of triangle. The more triangles contains in a sentence, the more important of the sentence. The set of sentences containing abundant triangles is extracted for summarizing the main idea of a document. The results verify the high competitiveness of the proposed strategy in extracting key-sentences among state-of-the-art approaches.*

KEYWORDS - *Clustering coefficient, Document summarization, Dependency structure analysis, Key-sentence extraction, Triangle identification*

I. INTRODUCTION

The introduction of the paper should explain the nature of the problem, previous work, purpose, and the contribution of the paper. The contents of each section may be provided to understand easily about the paper.

The techniques of viewpoint summarization automatically concludes the main idea of an article by computer so that readers can quickly grasp the viewpoint. The approaches of viewpoint summarization address the issues of keyword or key-phrase extraction [16] [29] [25] [26] [27]. Two novel approaches are proposed for identifying keywords in an extractive document [16]. Graph-based syntactic representation of electronic documents is employed. A node of the graph indicates a word of the document. In the supervised approach, the classification algorithm is trained on a collection of corpus for inducing keyword identification model. The graph-based feature is considered for characterizing graph structure, such as degree and frequency. In the unsupervised approach, the HITS algorithm is applied onto extracting the most significant words or phrases from document graph. For extracting both the local information and the global information between the set of neighbor documents and a given document, a graph-based ranking algorithm is proposed [29]. This approach consists of two stages: neighborhood construction and key-phrase extraction. The neighborhood construction aims at seeking a few neighbor documents which have similar topic or viewpoint with the given document at the first step. The neighbor documents are obtained by employing document similarity search. The set of expanded documents is then built. The global affinity graph is built for reflecting the neighborhood level co-occurrence relations among all candidate words in the set of expanded documents at the second stage. The saliency score of each word is computed based on the global affinity graph for evaluating the quantity of information about the viewpoint. A set of salient phrases as the key-phrases is finally chosen from the given document. Similarly, the Kea Key-phrase Extraction algorithm is proposed [25] for increasing the coherence of the extracted key-phrases. The Kea consists of four feature sets: baseline feature set, key-phrase frequency feature set, coherence feature set and merged feature set. Candidate phrases are generated by seeking the consecutive words of the document. The candidate phrases are converted to lower case, and classified into either key-phrase or non-key-phrase with associated feature vectors. The set of candidates with the highest probability is then output as the extracted key-phrases. As a type of learning algorithms, the efficiency of C4.5 Decision Tree Induction algorithm and GenEx algorithm are examed [26]. A decision tree maps the relations between internal nodes and leaf nodes. The internal nodes are labeled with feature value. The leaf nodes are labeled with class. A set of key-phrases is generated as the output result by inputting a document to the extractor of the GenEx algorithm. The genitor genetic algorithm of the GenEx algorithm tunes the parameters of extractor to maximize the fitness of training data, such that the input document processing can be determined. The number of match pairs between machine-generated phrases and human generated phrases is the metric to measure the performance of key-phrase extraction algorithm. A human-generated key-phrase matches a machine-generated key-phrase if they correspond to the same sequence. To summarize the viewpoint of an article, the information contained in other articles is topically close to the given article [27]. A set of neighbor documents related to a given document is initially searched by adopting document similarity search. The document similarity search seeks documents topically similar to a query article in a text corpus. A knowledge context for the given document is built based on the informativeness of neighbor documents to supply more information and clues for summarizing the given document. The Graph-ranking based algorithm incorporates the relationship between the given document and neighbor documents among sentences. The set of collected sentences for summarizing the specified document can be modeled as an undirected graph by generating an edge between a pair of sentences if their affinity weight is larger than 0. Otherwise, no edge is generated. The edge among sentences in the affinity graph is categorized into within-document edge and cross document edge. The affinity weight between a pair of sentences can be calculated by using the cosine measurement. The feature of fixed-length of sentence sequence is employed for representing the distributed relationships among words of the word vector in each paragraph [14]. The Paragraph Vector is an unsupervised algorithm which studies the feature of fixed length. The feature of fixe length is represented by variable length segments in a specified document, such as words and phrases. Each word in a specified document is mapped to a unique vector, and represented by a column in a matrix. The position of the word in the vocabulary indexes the column. The next word in the same sentence can be predicted based on the total sum of the vectors. The fourth word can be predicted by using average of the vector within three words. The paragraph vector represents the missing information of the current context, and acts as a memory of the main idea of the paragraph at last. The approach of event-based extractive summarization aims at reorganizing a set of extracted sentences according to the important events described by the summary [17]. Initially, the graph based on the event term semantic relations is constructed. The event terms of the graph are then clustered by employing the modified DBSCAN clustering algorithm. An event term is selected from a cluster for presenting the main idea of the document. Lastly, the summary is generated by extracting a set of sentences which contains more informative.

Various proposed applications are for document summarization, such as sentence-similarity-based language modeling for machine translation [8], the sentiment analysis of documents [10] and document recognition [15]. The clue applying sentence-similarity-based on edit-distance is proposed to supplement N-gram-based splitting method [8]. Sentences are divided into grams for candidate generation. A split sentence consists of a set of words and phrases. Then, proper candidates can be selected based on the computation of sentence similarity. The similarity between any two sentences is defined as the edit-distence between a pair of word sequences, where $0 <$ edit-distance < 1. Furthermore, both the method of ConvLstm [10] and the Graph Transformer Networks [15] employ the framework of convolutional neural network. The neural network architecture of ConvLstm employs the convolutional neural network and long short-term memory as the pretrained word vectors. The graph transformer network is a novel learning paradigm for document recognition [15]. It allows multi-module system to be globally trained by applying the gradient-based method. For neural networks have become the state-of-the-art models of machine learning problem in recent years. To understand the role and utility of various computational components of LSTM variants leads to a renewed interest [31]. The tasks of LSTM including speech recognition, handwriting recognition and polyphonic music modeling, are optimized by using random search as parameters of LSTM variants. Somehow, the standard LSTM architecture cannot be significantly improved by any of the variants.

To summarize the viewpoint of an article, keywords extraction plays a significant role in human language process [6] [4] [5] [33] [20]. A unified neural network architechture and learning algorithm for part-of-speech tagging, chunking, named entity recognition and semantic role labeling are proposed [5]. It studies internal representation of unlabeled training data. At the first layer, raw words are given to the network with valuable knowledge, then, transformed into feature vector by a look-up table operation. A tag decision for each word in a sentence is produced by combining the feature vectors with subsequent layers of the neural network. Word features at higher level can be extracted from the word feature vectors. The neural networks are then trained by using stochastic gradient ascent and maximized likelihood. A tagging system can be built with minimal computational requirement. Meanwhile, many high frequent words are meaningless, but some words are meaningful with low frequency. Furthermore, many words have similar spelling but completely different meaning, such as the words ”hot” and ”hat”, the words ”cat” and ”cap”. A method for estimating the probability [6] employs available information based on distributional word similarity. A similarity-based model is used to improve the probability estimation for unseen bigrams in a back-off language model. The similarity-based model consists of three stages: scheme of word pair decision requiring similarity-based estimate, information combination technique of similar words and measurement function of word pair similarity. Assume two disjoint words for appropriate sets are considered. So that the pair of words has second element if the conditional probability is given. The similarity of the word pair is considered as the conditional events for estimating the probability of the word pair. Moreover, the word association is also extended for semantic relations and lexico-syntactic co-occurrence. An objective measurement based on the theoretic notion of mutual information is proposed [4] for estimating word association norms. The KeyGraph algorithm that based on graph segmentation represents the co-occurrence between a pair of words in a document [20]. The set of keywords is extracted without using corpus or parsing tools by the KeyGraph algorithm. The algorithm mainly consists of four steps: building construction metaphor, preparation of document, extraction of terms and columns and extraction of roofs. Initially, the set of words is clustered. Then, the ranking of terms is computed based on the relations between a term and the set of clusters, such that the term can be selected as a keyword of the document. Except electronic documents and corpus, websites on the internet display rich contextual information. It is a substantial source of revenue to automatically extract keywords from the texts embedded in websites. A method is described [33] for extracting keywords from websites with the purpose of advertisement targeting. This method employs numerous features, such as term frequency of each potential keyword, inverse document frequency, the occurrence of term in search query logs. Initially, the HTML document is transformed into an easy-to-process plain text as preprocessor. Each word or phrase within a length of 5 in the document is considered as a candidate keyword. But if a phrase crosses sentences, it could not be chosen as a candidate for eliminating trivial errors. This strategy is called candidate selector. The binary classifier is then trained for predicting the likelihood between a candidate and its label. A set of generated keywords is ranked by computing the probability when the label of a candidate is predicted by the classifier. New keywords can be extracted from previous unbrowsed websites at last.

II. RELATED WORKS

Additional to the unsupervised learning based keyword or key-phrase extraction techniques as discussed in Section 1 , the arise of neural network technologies provide an alternative way for viewpoint summarization as the state-of-the-art supervised learning technique [12] [11] [7] [35] [37] [34] [24] [36] [30] recently. A method for the tasks of sentence classification based on trained convolutional neural networks with word vectors is proposed [12]. A k-dimensional word vector $x_i \in R^k$ is defined according to the i^{th} word of a sentence. The filter is applied to words for generating new features, so that a feature map for the set of words in a sentence can be generated. Next, a max-over-time pooling operation over the feature map is adopted then. The maximum value is taken according to a specific filter. The most important feature with the highest value can be captured for each feature map. The penultimate layer of the neural network consists of the feature sets. The set of features is passed to the softmax layer. So two channels of word vectors keep static by continuous training and fine-tuned through backpropagation. Features of the words in a sentence play a vital role in sentence modelling and document analysis. Modelling sentences by convolutional neural networks with dynamic k-max pooling is proposed in [11]. The dynamic *k*-max pooling gives the wide convolutional layers and dynamic pooling layers with two feature maps respectively. Given a sentence s in document *D*, the embedded $w_i \in R^d$ is taken to obtain the fully connected layer. The matrix $s \in R^d$ of the given sentence *s* can be constructed. The values of w_i are optimised parameters for training the neural networks. By convolving a weighted matrix with activation below the layer, the convolutional layer is then obtained. The max pooling over the time dimension is employed as a pooling operation next. A value *k* and a sequence $P \in R^p$ are given, where $p \geq k$. The *k*-max pooling chooses *k* with the highest values of *P* for each subsequence $P^k_{max..}$

The order of the values in P^k_{max} is based on the original order of *P*. The values of *k* is not fixed, but dynamically selected for extracting higher order and longer-range features. So, the feature graph of the given sentence is induced by the succession of convolutional and pooling layers. The word relations within varying size can be captured by the feature graph. Moreover, the dynamic *k*-max pooling operator is used as a non-linear subsampling function of dynamic convolutional neural network. However, it is difficult to guarantee the stable rising order of the values of *k* due to the dynamical selection of value *k*. The k-max pooling is also used in the convolution architecture as a non-linear subsampling operator for extracting the most relevant global features from a given short text [30]. This proposed method combines the semantic clustering algorithm and the well pretrained convolutional architecture for modelling short text. The clustering algorithm [21] based on searching density peaks groups semantically related words, and seeks semantic cliques. Two quantities of an individual word *i* are computed: the local density p_i and the distance δ_i . The candidate Semantic Units (SU for short) appear in the given short text can be detected by performing the semantic position over n-gram. The output of this procedure is the pretrained word embedding for initializing the look-up table at the first layer of the convolutional architecture. More complexity features can be extracted from higher levels of the convolutional neural network. However, the determined meaningful key-phrases would appear at any position of the given short text. Thus, simple combination of all words in the given short text may result in unnecessary divergence and erroneous semantic representation. The task of short text sentiment analysis is challenging due to the limited contextual information. The short text could be a sentence or messages posted onto the social networks. The convolutional architecture of Character to Sentence Convolutional Neural Network (CharSCNN for short) exploits prior knowledge from character-level to sentence-level information for short text sentiment analysis [7]. The morphological, syntactic and semantic information of words is captured by transforming words into real-valued feature vectors at the first layer. Ignoring the sizes of different sentences and the position of important information in the sentence, the local features of each word is produced at the second layer. A max operation is then employed for creating a fixed-sized feature vector of the sentence, such that the feature vector and the word vector can be combined. Similarly, the CharSCNN network requires pretraining with using minimized negative likelihood over the training data. Except the short text sentiment analysis, the convolutional neural network is also applied onto sentence classification [35] and ontology classification [37]. One-layer of convolutional neural network is conducted for sentence classification based on sensitivity analysis. It is a standard baseline technique, which akin to the support vector machine and logistic regression due to its simplicity and empirical performance. The temporal convolutional neural network achieves amazing performance in understanding text without prior knowledge of words and sentences [37]. Through analyzing the applications of convolutional neural networks on document summarization, the contextual structures play a crucial role in capturing salient semantic information. Therefore, it is a critical issue of modeling sentences in a document due to its wide variety of tasks. The attention-based convolutional neural network (ABCNN for short) consists of three architectures based on the BCNN [34]. The ABCNN-1 is motivated by training convolution to learn "counterpart-biased" sentence representations. The ABCNN-1 employs the attention feature matrix to influence convolution, such that the units of a sentence can be highly weighted in convolution, where the unit represents words on the lowest level and phrases on higher levels of the network. The ABCNN-2 computes the weights of attention based on the convolutional output of ABCNN-1 with the purpose of reweighting this convolutional output. The ABCNN-2 pooling generates an output feature map within the same size as the input feature map so that the features of increasing abstraction can be extracted by stacking multiple convolutional pooling blocks. Finally, the ABCNN-3 combines the architectures of ABCNN-1 and ABCNN-2 by allowing the attention mechanism so that both the convolutional pooling and input/output granularity are operated. The proposed three attention architectures integrate mutual influence between a pair of sentences of a document into convolutional neural networks. The representation of each sentence is considered as its counterpart so that the interdependent sentence pair representations can be weighted in the attention feature matrix. Besides, a novel latent semantic model that incorporating a convolutional pooling structure over word sequences is proposed in [24]. Every word within a temporal context window in a word sequence captures the contextual features at the word n-gram level. The salient word n-gram features in the word sequence can be discovered. A sentence level feature vector is then formed. In order to generate a continuous vector representation of a given document, a non-linear transformation is applied to extract high-level semantic information finally. The proposed convolutional latent semantic model (CLSM) is trained on click through data. Likewise, the recurrent neural network is practically appliable to the task of document classification [13]. Traditional document classifiers rely on human-designed features, such as dictionaries and knowledge bases as aforementioned. The recurrent convolutional neural network for document classification without human-designed features is applied onto capturing contextual information [13]. The key role of word segmentation in capturing the key components of a document is automatically judged by a max-pooling layer.

III. PROBLEM DEFINITION

The triangle-based key-sentence extraction algorithm that proposed in [3] [33] extends the KeyGraph algorithm by the following ways. We refines the proposed algorithm and enriches relevant experiments for efficiency verification in this paper.

- Build a dependency graph where nodes represent words with high frequency, and edges represent both the co-occurrence and syntactic dependency relations
- Syntactic dependency relation is employed so that the keywords with certain relationship can be extracted
- Clustering coefficient is the measurement for the importance of nodes
- Strongly connected components are extracted in terms of triangles based on the transitivity of the dependency graph

A. Word Frequency

The word frequency, also called term frequency, is the occurrence frequency of a word *w* in *D*, where $w \in D$. The word frequency is denoted as *tf(w)*. Let *Stop* be the set of meaningless words, and the *HighFreq* be the set of words *w* with high occurrence frequency in *D*. Therefore, $w \in HighFreq$, but $w \notin Stop$ satisfies the parameter that $tf(w) > \delta$, where the given threshold $\delta > 0$.

B. Co-occurrence Frequency

Co-occurrence is a crucial indicator in linguistic analysis for it defines the connectivity frequency of word pairs. Since co-occurrence in linguistic sense can be interpreted as an indicator of semantic proximity, it has various applications, such as keyword-brand associations, search volume co-occurrence, keywords search and terms discovery. Classification of co-occurrence is listed below:

- Global: extracted from databases
- Local: extracted from individual documents or sentences
- Fractal: extracted from self-similar, scaled distribution

The local meaning of co-occurrence is employed for modelling the occurrence frequency between two words w_i and w_j due to its importance in computing the dependency frequency. The word w_j co-occurs with word w_i if word w_i connects to word w_j . The co-occurrence can be formulated as *(1)*.

$$co(w_i, w_j) = \begin{cases} 1, \ w_i \ \textit{is adjacent to} \ w_j \\ 0, \ \textit{Otherwise} \end{cases} \qquad (1)$$

The table I describes the employed definitions.

TABLE I. FREQUENTLY USED NOTATIONS

Notation	Description
G	Dependency graph
V	The set of nodes of G
E	The set of edges of G
D	A document
w	A word in the document
$tf(w)$	Term frequency of words
Stop	The set of meaningless words
HighFreq	High occurrence frequency of words
$co(w_i, w_j)$	Co-occurrence frequency of two words
$dep(w_i, w_j, s_k)$	Dependency frequency of word pair
$dp(w_i, w_j, s_k)$	Undirected dependency relation
$df(w_i, w_j)$	Dependency frequency
$df(G)$	Dependency weight of graph G
$N(v_i)$	Neighbor nodes of node v_i
$ccf(v_i)$	Clustering coefficient of node v_i
δ	Threshold of term frequency
λ	Threshold of dependency frequency
τ	Number of triples of a node
μ	Threshold of clustering coefficient

C. Dependency Frequency

A document D consists of a set of sentences, denoted as $D = \{s_1,s_2,s_3.....s_k\}$. A word w_i depends on the word w_j in sentence s_k if w_i is syntactically modified by w_j , or w_i co-occurs with w_j . The dependency relation between words w_i and w_j is denoted as $w_i \rightarrow w_j$. Take a sentence ”Tom sent three letters to Jim in last week.” as an example. The noun ”Tom” depends on the verb ”sent”, such that the dependency relation between ”Tom” and ”sent” is indicated by Tom → sent. The verb ”sent” depends on the noun ”letters” grammatically for the noun ”letters” describes the object and purpose of the action ”sent”. Therefore, the dependency relation between ”sent” and ”letters” is indicated as sent → letters. Similarly, the dependency relation between the noun ”letters” and ”Jim” can also be denoted by letters → Jim. Let dep(w_i, w_j, s_k) be the indicator function of dependency relation defined as formula *(2)*.

$$dep(w_i, w_j, s_k) = \begin{cases} 1, & w_i \ \text{dependson} w_j \ \text{in} \ s_k \\ 0, & \text{Otherwise} \end{cases} \tag{2}$$

The direction of dependency relation can be ignored due to the characteristic of triangle as fully connected cycle. Therefore, the dependency relation between w_i and w_j in sentence s_k can be described as:

$$dp(w_i,w_j,s_k)=dep(w_j,w_i,s_k)\vee(w_i,w_j,s_k)$$

The dependency frequency between w_i and w_j in document D can be defined as:

$$df(w_i,w_j)=\sum_{k=1}^{m} co(w_i,w_j,s_k)\vee dp(w_i,w_j,s_k)$$

D. Dependency Graph

Dependency is one-to-one correspondence between words, for every element (e.g. word or morph) in a sentence *s*, a node in the structure of the sentence s exactly corresponds to the element. So the result of one-to-one correspondence is word (or morph) grammar.

A dependency graph *G* of a given document *D* is defined as $G = (V, E)$, where V is the set of nodes, and E is the set of edges. The $V = HighFreq$ and $E = \{(w_i, w_j) \mid \mathrm{df}(w_i, w_j) > \lambda$, where $\lambda > 0\}$. Therefore, *G* is a weighted graph. The set of nodes represents high frequent words in *D*. The set of edges represents the dependency relations between two words in a sentence *s*. The dependency weight (*df*) of graph $G = (V, E)$ is denoted as following:

$$df(G)=\sum_{(v_i,v_j)\in E} df(v_i,v_j)$$

E. Clustering Coefficient

The clustering coefficient (ccf for short), as one of the applications of triangle, measures the degrees of nodes tending to cluster together as a small group in a graph [31]. There are two versions of the ccf measurement: the global and the local. The global version is designed to give an overall indication of clustering in a graph, whereas the local gives an indication of connectivity of single node. The local clustering coefficient is considered by the proposed method for key-sentences extraction in a single document *D*.

An $e_{ij} = (v_i, v_j)$, where $e_{ij} \in G$, connects nodes v_i and v_j. The neighborhood $N_{(vi)}$ for the node v_i is defined as its immediately connected neighbor:

$$N_{(v_i)}=\{v_j|e_{ij}=(v_i,v_j)\in E\}$$

Let $d_{(vi)}$ be the degree of node v_i so that $\mathrm{d}_{(vi)} = |\mathrm{N}_{(vi)}|$. The degree $\mathrm{d}_{(vi)}$ is the number of nodes adjacent to v_i. The measurement of local clustering coefficient is defined as the probability, which a random pair of its neighbor nodes is connected by an edge.

$$ccf(v_i)=\frac{|e_{(jk)}\,|\,(v_j,v_k)\in N_{(v_i)},e_{(jk)}\in E\,|}{\binom{|V_{(v_i)}|}{2}}$$

A triangle is a complete subgraph that contains three fully connected nodes. Let $\lambda_{(vi)}$ be the number of triangles including node v_i . A triple at the node v_i is a path of length 2 for v_i is the centralized node. Let $\tau_{(vi)}$ be the number of triples on $v_i \in V$. $\tau_{(vi)}$ is the number of subgraphs containing two edges and three nodes, one of which is v_i , such that v_i is adjacent to both edges. The clustering coefficient of node v_i can be computed by the formula, where $0 < \mathrm{ccf}(v_i) < 1$:

$$ccf(v_i)=\frac{\lambda_{(v_i)}}{\tau_{(v_i)}}$$

The maximal value ccf is 1 when every neighbor node connects to v_i is also connected by every other node within the neighborhood. The minimal value of ccf is 0 if none of the neighbor nodes of v_i connect to each other. The Fig. 1 shows the computation of clustering coefficient of a node in *G*.

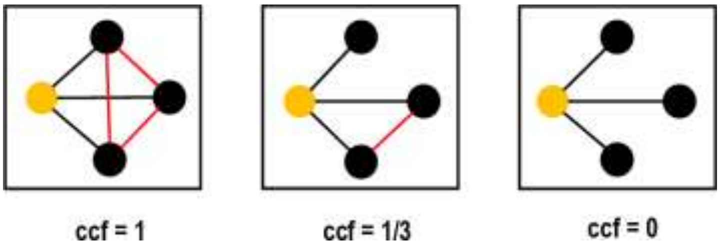


Fig. 1. The degree of node vi in yellow is 3 for it has three neighbors nodes in black. The number of edges in red among three neighbor nodes is 3, 1 and 0 from left to right. Thus, the values of ccf are 1, 1/3 and 0, respectively.

IV. SINGLE DOCUMENT SUMMARIZATION

Given a document *D*, all of the meaningless words, which defined as *Stop* words with high occurrence frequency are filtered out initially in order to improve the precision of co-occurrence frequency and dependency frequency. The procedure of summarizing the viewpoint of the given document *D* begins from syntax analysis of all remained words in *D*. A dependency graph *G* is then built for *D*. The set of nodes in *V* represents the set of remained words. And the set of edges in *E* represents the dependency relations among the set of remained words. However, not all of the remained words is useful and meaningful for summarizing the viewpoint of *D*. Hence, the local clustering coefficient is employed for measuring the clustering coefficient value of every node in *G*. The nodes with the value of clustering coefficient that smaller than a threshold μ are removed from *G*. We can obtain the graph *G'*, where $G' \in G$. The set of words in graph *G'* holds stronger connectivity than the set of words in graph *G*. All triangle in *G'* are extracted, where $T = \{T_1, T_2, T_3....T_i\}$. More triangles anchor in a sentence s_k , more importance of the sentence s_k in *D*.

The details of TriangleSum algorithm is described in algorithm 1. With the set of extracted triangles, summarization of a document can be achieved in two ways:

- **Entrance sentence**: extract sentence from the entrance of a paragraph containing more triangles. Because bushy paths are more likely to contain information, which centralize to the viewpoint of the document *D*.
- **Anchored sentence**: extract sentences that anchored with more triangles. Triangles indicate the importance of a sentence in the document *D*.

Given a document taken from technology column of bbc news that includes three sentences as an example. The dependency graph *G* for all words of the given document *D* is constructed by applying the TriangleSum algorithm shown as Fig. 2. Three triangles are identified in the dependency graph *G*. The T_1 = {Google, tool, rolling}, T_2 = {rolling, tool, android} and T_3 = {android, view, users}. A key sentence as a summarization can be extracted based on the three triangles T_1, T_2 and T_3.

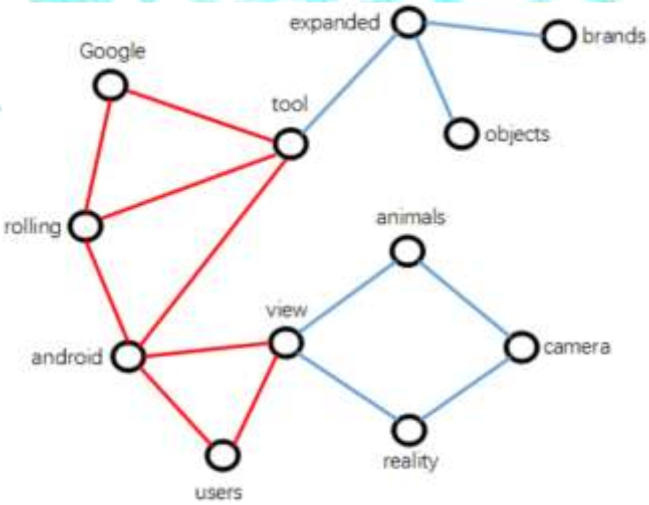
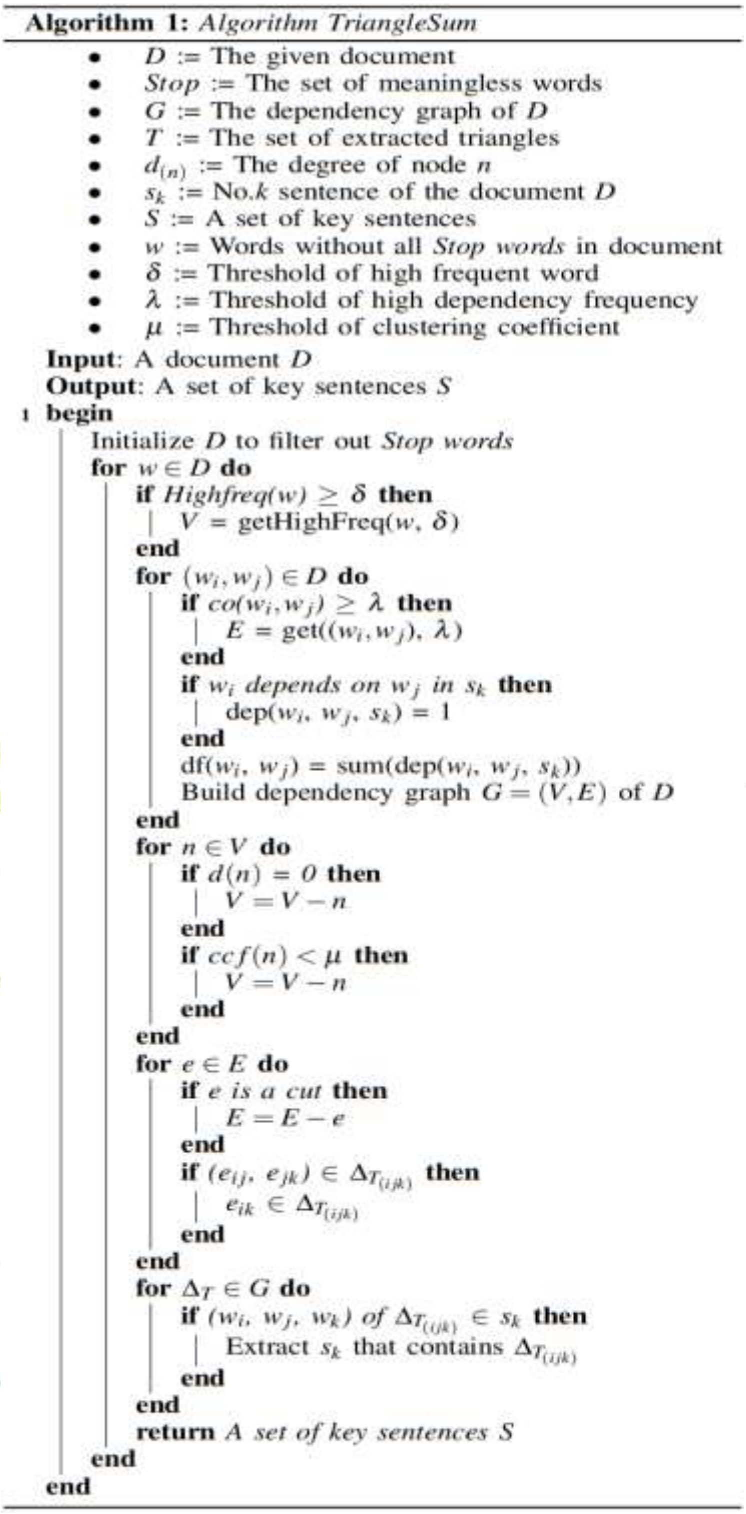


Fig. 2. The Dependency Graph of Words in Document *D*

Exapmle Document D of Fig.2: Google is rolling out a tool that allows Android users to view moving animals which make sounds in augmented reality through their device's camera. It currently only works with some animals but could be expanded to include objects and brands in future. It was unveiled last month at the firm's annual event for developers who create apps for Android devices.

To implement the TriangleSum algorithm, two functions are required: efficient computation of local clustering coefficient for all nodes in *V*, and the identification of all triangles from each connected subgraphs. A triangle is denoted as $T=\{u, v, w\}$, where $\{u, v, w\}$ V. All nodes are fully connected to each her as a completely closed cycle. The Breadth-first Search simplifies the identification of triangles by traversing the entire dependency graph G.



Algorithm 1: *Algorithm TriangleSum*

- D := The given document
- $Stop$:= The set of meaningless words
- G := The dependency graph of D
- T := The set of extracted triangles
- $d_{(n)}$:= The degree of node n
- s_k := No.k sentence of the document D
- S := A set of key sentences
- w := Words without all $Stop$ $words$ in document
- δ := Threshold of high frequent word
- λ := Threshold of high dependency frequency
- μ := Threshold of clustering coefficient

Input: A document D
Output: A set of key sentences S

```
begin
    Initialize $D$ to filter out $Stop$ $words$
    for $w \in D$ do
        if $Highfreq(w) \geq \delta$ then
            $V$ = getHighFreq($w$, $\delta$)
        end
        for $(w_i, w_j) \in D$ do
            if $co(w_i, w_j) \geq \lambda$ then
                $E$ = get(($w_i, w_j$), $\lambda$)
            end
            if $w_i$ depends on $w_j$ in $s_k$ then
                dep($w_i$, $w_j$, $s_k$) = 1
            end
            df($w_i$, $w_j$) = sum(dep($w_i$, $w_j$, $s_k$))
            Build dependency graph $G = (V, E)$ of $D$
        end
        for $n \in V$ do
            if $d(n)$ = $0$ then
                $V = V - n$
            end
            if $ccf(n) < \mu$ then
                $V = V - n$
            end
        end
        for $e \in E$ do
            if $e$ is a cut then
                $E = E - e$
            end
            if $(e_{ij}, e_{jk}) \in \Delta_{T_{(ijk)}}$ then
                $e_{ik} \in \Delta_{T_{(ijk)}}$
            end
        end
        for $\Delta_T \in G$ do
            if $(w_i, w_j, w_k)$ of $\Delta_{T_{(ijk)}} \in s_k$ then
                Extract $s_k$ that contains $\Delta_{T_{(ijk)}}$
            end
        end
        return A set of key sentences $S$
    end
end
```

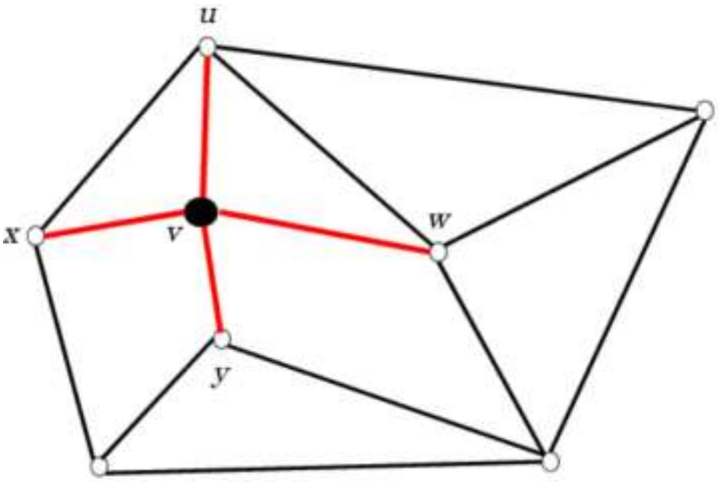
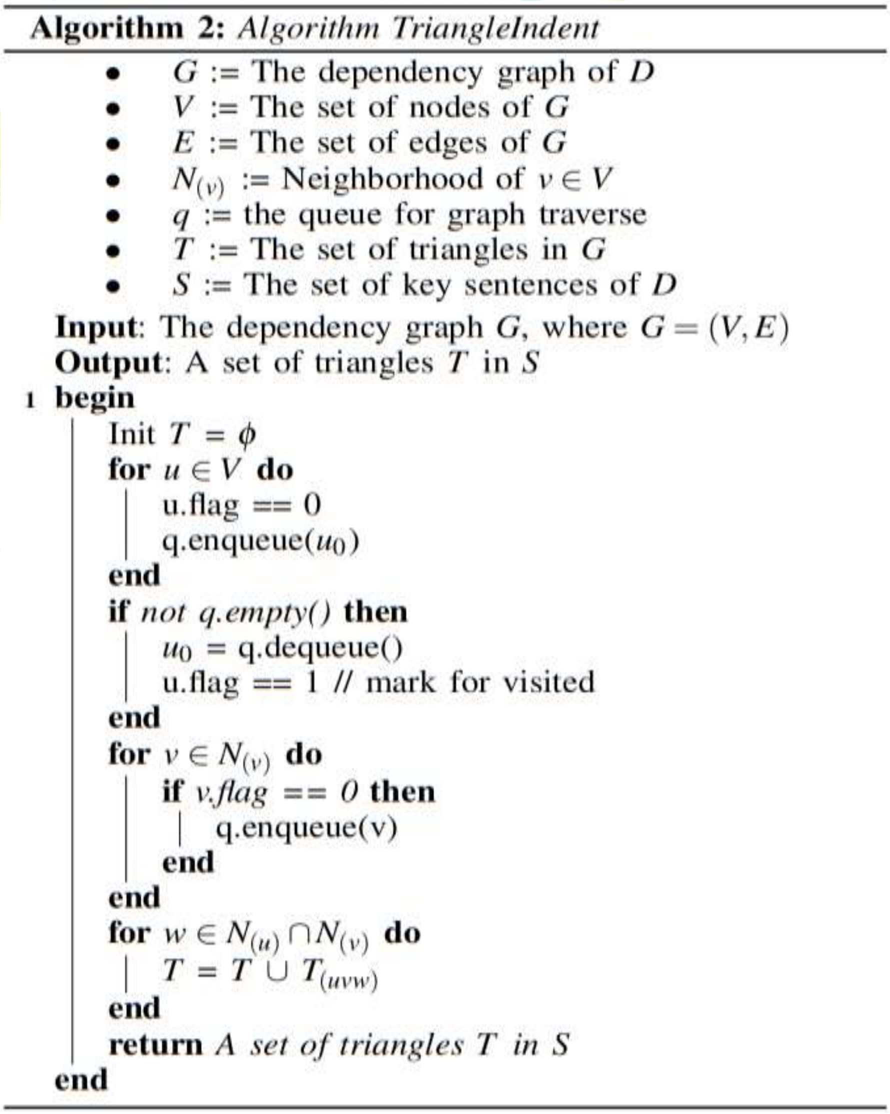


Fig. 3. Identification of Triangle based on Breadth-first Search

In Fig. 3, the neighborhood $N_{(v)}$ of node v enters the queue when node v is visited. The connection among the neighbor nodes of the root node is visited. Triangles are identified if there is an edge between two neighbor nodes of the root node. An edge connects the nodes u and w, such that the triangle $T = \{u, v, w\}$ is identified since both nodes u and w are the neighbors of node v.



```
Algorithm 2: Algorithm TriangleIndent
    • $G$ := The dependency graph of $D$
    • $V$ := The set of nodes of $G$
    • $E$ := The set of edges of $G$
    • $N_{(v)}$ := Neighborhood of $v \in V$
    • $q$ := the queue for graph traverse
    • $T$ := The set of triangles in $G$
    • $S$ := The set of key sentences of $D$
Input: The dependency graph $G$, where $G = (V, E)$
Output: A set of triangles $T$ in $S$
begin
    Init $T = \phi$
    for $u \in V$ do
        u.flag == 0
        q.enqueue($u_0$)
    end
    if not q.empty() then
        $u_0$ = q.dequeue()
        u.flag == 1 // mark for visited
    end
    for $v \in N_{(v)}$ do
        if v.flag == 0 then
            q.enqueue(v)
        end
    end
    for $w \in N_{(u)} \cap N_{(v)}$ do
        $T = T \cup T_{(uvw)}$
    end
    return A set of triangles $T$ in $S$
end
```

The algorithm 2 named TriangleIndent shows that from the root node, all child nodes obtained by expanding one node are added to the queue. The unvisited nodes for their neighbor nodes are placed in containers as the queue or linked list called ”open”. Once examined, the nodes are then moved into the containers called ”closed” until all nodes are visited.

V. EXPERIMENTAL EVALUATION

The proposed algorithms are applied onto several bench- marks with real-world corpus for efficiency verification. The set of corpus that released by the Text Analysis Conference is a set of BBC News collection in different topics. We mainly record experimental results within three aspects: the number of extracted triangles, the word frequency and the number of key-sentences, as well as the comparative results of recall among the proposed method, KeyGraph algorithm [20] and the keyphrase extraction algorithm based on neighborhood knowledge [29]. The experimental results are concluded in detail in section V.

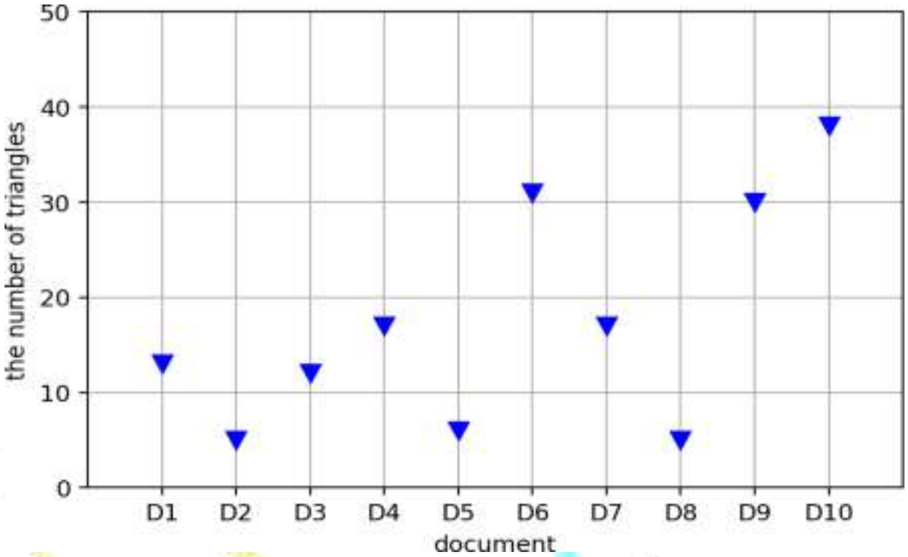


Fig. 4. The Number of Counted Triangles

Fig. 4 records the numbers of extracted triangles in each document. The variable of the x-axis represents the set of documents, and the y-axis records the numbers of triangles in each document. The triangle counting results prove the effectiveness of the proposed algorithm in extracting trianglges from each document. The total numbers of triangles depend on the total numbers of sentences in a document and the grammatical dependence relationship among words.

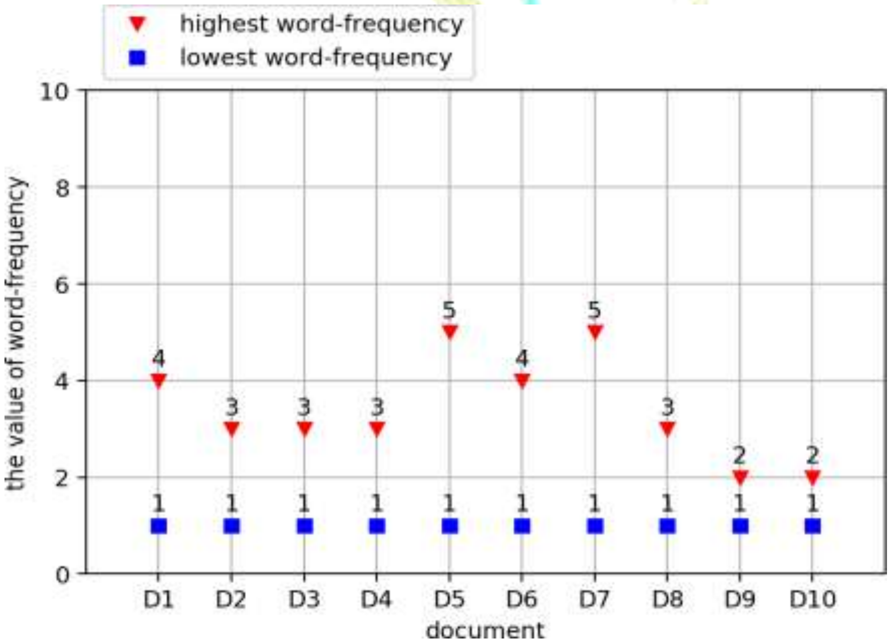


Fig. 5. The Word Frequency

Fig. 5 illustrates the occurrence frequency of words in a document. The *x*-axis indicates the set of documents. The *y*-asix shows the occurrence frequency of words in a document. The red graph records the highest occurrence frequency, as well as the blue graph records the lowest occurrence frequency. The value of blue graph remains the same of all documents due to the similar size of all documents. Moreover, the minimal occurrence frequency of words remains 1 for most of words merely occurs once in a document. The table II records the set of words within highest frequency in each document.

TABLE II. FREQUENTLY USED NOTATIONS

Document	Word with highest frequency
D_1	Huawei
D_2	Labour, Education
D_3	Growth
D_4	Music
D_5	Food
D_6	Kim
D_7	Trade
D_8	Baby
D_9	Police, Vehicle, Crash, Car, One
$D10$	Store, Jobs, Risk, Boohoo, Brands

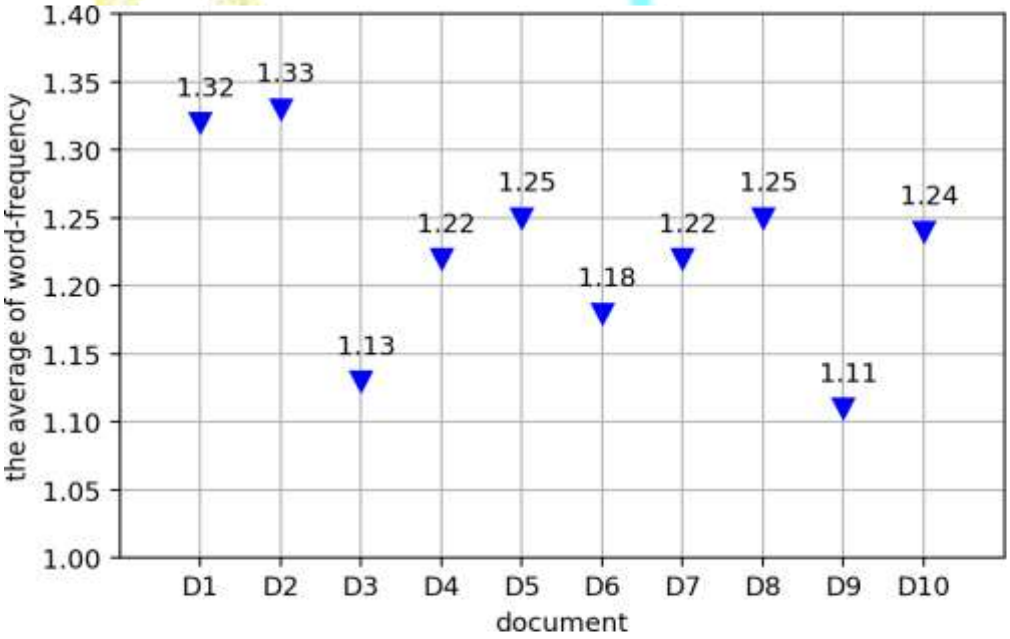


Fig. 6. The average word frequency

Both recall and precision are employed to evaluate the proposed approach. The *x-axis* of Fig. 7 indicates the set of documents. The *y-axis* of Fig. 7 records both the results of recall and precision. The recall is defined as the formula below.

$$recall = \frac{keysentence_{(by\ human)} \cap keysentence_{(by\ system)}}{number\ of\ keysentences\ identified\ \ by\ human}$$

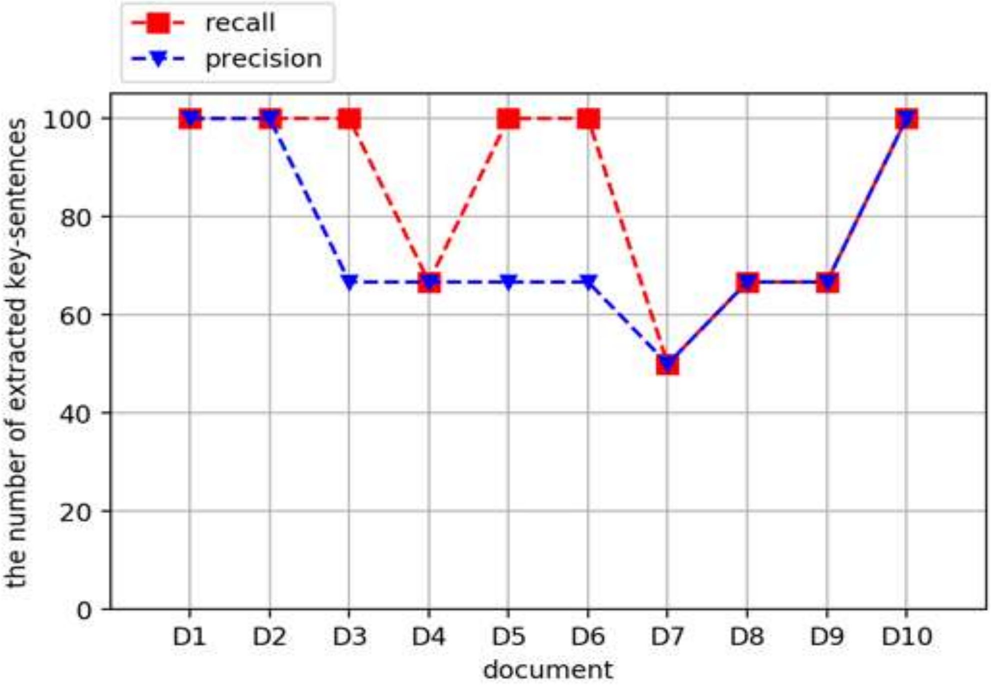


Fig. 7. The Number of Extracted Key-sentences

To evaluate the efficiency and accuracy, the intersection set of key-sentences manually identified by human and key-sentences extracted by system increases, the recall of the proposed approach increases. Likewise, the precision is the same as the recall. The precision is calculated by using the following formula.

$$precision = \frac{keysentence_{(by\ human)} \cap keysentence_{(by\ system)}}{Total\ number\ of\ sentences\ in\ D}$$

The process of removing the set of meaningless words in a document *D* by using the notion of Stop words could reduce the interference of meaningless words on extracting key-sentence. Besides, the set of given thresholds could also affect the performance of the proposed approach.

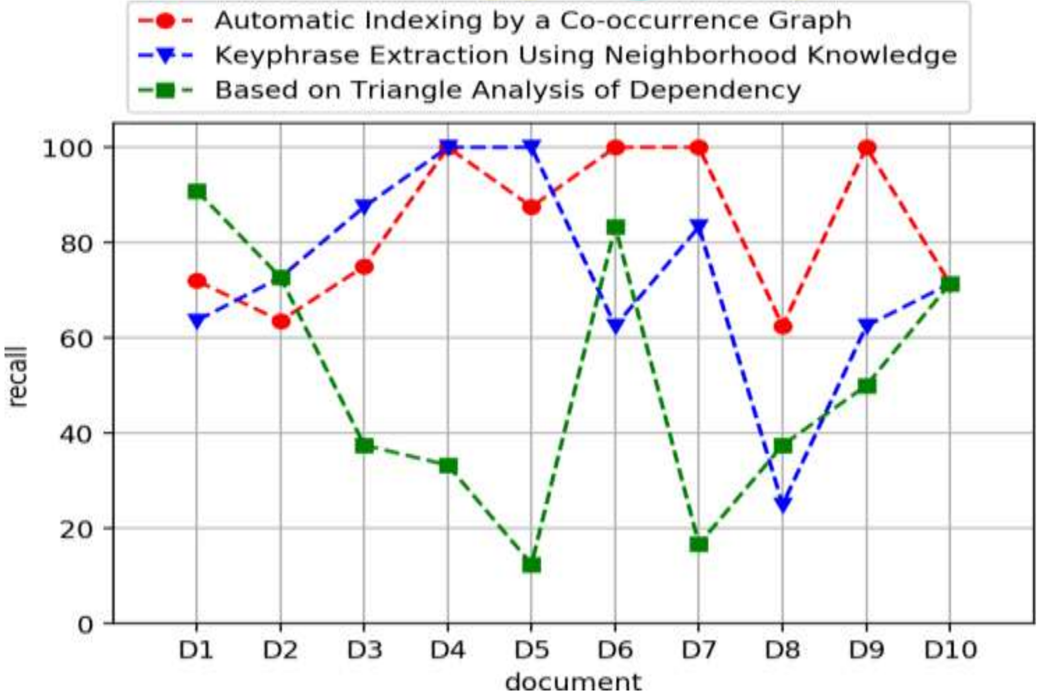


Fig. 8. The results of Recall

We finally exam the performance of keyword extraction by using the proposed algorithm and other two algorithms. The results are recorded by the Fig. 8. The purpose of the proposed algorithm is to extract key-sentence from document *D*, so that the keyword extracted by the proposed algorithm does not perform well as other two algorithms. For not all words that contained in a sentence are keywords, nevertheless, the set of words can locates the key-sentences in a given document *D*.

VI. CONCLUSION

In this paper, we refine the proposed algorithm for automatical extraction of key-sentences from single document, and perfect the experimental results. The proposed approach does not require pretraining process. The algorithm efficiently extracts triangles as anchor points of key-sentences from an input document. The set of triangles can be then used to identify key-sentences that centralize to the main idea of a document. Moreover, the proposed approach relies on high quality results of morphology parsing and syntactic parsing. However, for some thresholds are used onto the computation of word frequency and dependency frequency, the efficiency of the proposed approach can be improved.

For the future works, we will develop a document summarization approach by combining the architecture of convolutional neural networks and the technique of fuzzy search.

VII. Acknowledgements

This research is supported by the research fund of Shaoguan University under Grant No.SY2017KJ06.

REFERENCES

[1] Al-Abdallah, R. Z., & Al-Taani, A. T. (2017). Arabic single-document text summarization using particle swarm optimization algorithm. Proce- dia Computer Science, 117, pp.30-37.

[2] Al-Sabahi, K., & Zuping, Z. (2019). Document Summarization Us- ing Sentence-Level Semantic Based on Word Embeddings. Interna- tional Journal of Software Engineering and Knowledge Engineering, Vol.29(02), pp.177-196.

[3] Cheng, K., Li, Y., & Wang, X. (2013). Single Document Summarization Based on Triangle Analysis of Dependency Graphs. In 2013 16th International Conference on Network-Based Information Systems, pp.38- 43.

[4] Church, K. W., & Hanks, P. (1990). Word association norms, mutual in- formation, and lexicography. Computational linguistics, Vol.16(1), pp.22- 29.

[5] Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., & Kuksa, P. (2011). Natural language processing (almost) from scratch. Journal of machine learning research, pp.2493-2537.

[6] Dagan, I., Lee, L., & Pereira, F. C. (1999). Similarity-based models of word cooccurrence probabilities. Machine learning, Vol.34(1-3), pp.43- 69.

[7] Dos Santos, C., & Gatti, M. (2014). Deep convolutional neural networks for sentiment analysis of short texts. In Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Techni- cal Papers, pp.69-78.

[8] Doi, T., & Sumita, E. (2004). Splitting input sentence for machine translation using language model with sentence similarity. In COLING 2004: Proceedings of the 20th International Conference on Computa- tional Linguistics, pp.113-119.

[9] Greff, K., Srivastava, R. K., Koutnk, J., Steunebrink, B. R., & Schmid- huber, J. (2016). LSTM: A search space odyssey. IEEE transactions on neural networks and learning systems, Vol.28(10), pp.2222-2232.

[10] Hassan, A., & Mahmood, A. (2017). Deep learning approach for sentiment analysis of short texts. The 3rd International Conference on Control, Automation and Robotics, pp. 705-710.

[11] Kalchbrenner, N., Grefenstette, E., & Blunsom, P. (2014). A con- volutional neural network for modelling sentences. arXiv preprint arXiv:1404.2188.

[12] Kim, Y. (2014). Convolutional neural networks for sentence classifica- tion. arXiv:1408.5882.

[13] Lai, S., Xu, L., Liu, K., & Zhao, J. (2015). Recurrent convolutional neural networks for text classification. The 29th AAAI International Conference on Artificial Intelligence.

[14] Le, Q., & Mikolov, T. (2014). Distributed representations of sentences and documents. The International Conference on Machine Learning, pp.1188-1196.

[15] LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient- based learning applied to document recognition. Proceedings of the IEEE, Vol.86(11), pp.2278-2324.

[16] Litvak, M., & Last, M. (2008). Graph-based keyword extraction for single-document summarization. Proceedings of the Workshop on Multi- source Multilingual Information Extraction and Summarization, pp.17- 24.

[17] Liu, M., Li, W., & Ji, D. (2010). Multi-document summarization based on event term semantic relation graph clustering. The Journal of Chinese information processing.

[18] Mallick, C., Dutta, M., Das, A. K., Sarkar, A., & Das, A. K. (2019). Extractive Summarization of a Document Using Lexical Chains. In Soft Computing in Data Analytics, pp.825-836.

[19] Naserasadi, A., Khosravi, H., & Sadeghi, F. (2019). Extractive multi- document summarization based on textual entailment and sentence compression via knapsack problem. Natural Language Engineering, Vol.25(1), pp.121-146.

[20] Ohsawa, Y., Benson, N. E., & Yachida, M. (1998). KeyGraph: Auto- matic indexing by co-occurrence graph based on building construction metaphor. Proceedings of the IEEE International Forum on Research and Technology Advances in Digital Libraries, pp.12-18.

[21] Rodriguez, A., & Laio, A. (2014). Clustering by fast search and find of density peaks. Science, Vol.344(6191), pp.1492-1496.

[22] Saini, N., Saha, S., Jangra, A., & Bhattacharyya, P. (2019). Extractive single document summarization using multi-objective optimization: Ex- ploring self-organized differential evolution, grey wolf optimizer and wa- ter cycle algorithm. Proceedings of Knowledge-Based Systems, Vol.164, pp.45-67.

[23] Shafiee, F., & Shamsfard, M. (2018). Similarity versus relatedness: A novel approach in extractive Persian document summarisation. Journal of Information Science, Vol.44(3), pp.314-330.

[24] Shen, Y., He, X., Gao, J., Deng, L., & Mesnil, G. (2014). A latent semantic model with convolutional-pooling structure for information retrieval. Proceedings of 23rd ACM International Conference on Infor- mation and Knowledge Management, pp.101-110.

[25] Turney, P.D. (2003). Coherent keyphrase extraction via web mining. arXiv preprint cs/0308033.

[26] Turney, P.D. (2000). Learning algorithms for keyphrase extraction. Information retrieval, Vol.2(4), pp.303-336.

[27] Wan, X., Yang, J., & Xiao, J. (2007, July). Single document summariza- tion with document expansion. The Proceedings of AAAI, pp.931-936.

[28] Wan, X., Luo, F., Sun, X., Huang, S., & Yao, J. G. (2019). Cross- language document summarization via extraction and ranking of mul- tiple summaries. Proceedings of Knowledge and Information Systems, Vol.58(2), pp.481-499.

[29] Wan, X., & Xiao, J. (2008, July). Single Document Keyphrase Extrac- tion Using Neighborhood Knowledge. AAAI, Vol.8, pp.855-860.

[30] [30] Wang, P., Xu, B., Xu, J., Tian, G., Liu, C. L., & Hao, H. (2016). Semantic expansion using word embedding clustering and convolutional neural network for improving short text classification. Neurocomputing, Vol.174, pp.806-814.

[31] Watts, D. J., & Strogatz, S. H. (1998). Collective dynamics of small- worldnetworks. nature, Vol.393(6684).

[32] Yang, K., Al-Sabahi, K., Xiang, Y., & Zhang, Z. (2018). An Integrated Graph Model for Document Summarization. Information, Vol.9(9).

[33] Yanting Li, Kai Cheng. (2011). Key Sentence Extraction from Single Document based on Triangle Analysis in Dependency Graph. Information Science Journal of Kyushu Sangyo University, Vol.10(1), pp.62-66.

[34] Yih, W. T., Goodman, J., & Carvalho, V. R. (2006, May). Finding adver- tising keywords on web pages. In Proceedings of the 15th International Conference on World Wide Web, ACM, pp.213-222.

[35] Yin, W., Schtze, H., Xiang, B., & Zhou, B. (2016). Abcnn: Attention- based convolutional neural network for modeling sentence pairs. Trans- actions of the Association for Computational Linguistics, Vol.4, pp.259- 272.

[36] Zhang, Y., & Wallace, B. (2015). A sensitivity analysis of (and practitioners' guide to) convolutional neural networks for sentence clas- sification. arXiv:1510.03820.

[37] Zhang, Y., Er, M. J., Zhao, R., & Pratama, M. (2016). Multiview con- volutional neural networks for multidocument extractive summarization. IEEE transactions on cybernetics, Vol.47(10), pp.3230-3242.

[38] Zhang, X., & LeCun, Y. (2015). Text understanding from scratch. arXiv:1502.01710.

APPENDIX

The set of real-world corpus that records BBC News in different topics is listed in the section of appendix.

1) Huawei: Government decision on 5G rollout delayed. A decision on whether controversial Chinese firm Huawei should be excluded from the rollout of 5G mobile phone networks in the UK has been postponed. Culture secretary Jeremy Wright said the government is ”not yet in a position” to decide what involvement Huawei should have in the 5G network. Mr. Wright said, the implication of the recent US ban on its companies from dealing with Huawei was not clear. Until it was, he said the government would be ”wrong” to make a decision.(Technology and Science Column)

2) Labour calls for action on ’catastrophic fall’ in adult learning. Labour has renewed calls for a ”cradle-to-grave” National Education Service, after a ”catastrophic fall” in adult learners. Shadow education minister Gordon Marsden has called for action, saying government cuts have led to fewer adults choosing to study later in life. Labour says millions of adults in the UK lack basic skills and are unable to access education and training. The government says further education and skills is a priority.(Education Column)

3) Dwarfism drug aims to boost healthy growth. Scientists hope a new type of medication could boost healthy growth in children born with dwarfism. Sam Short, nine, from south-west London has been on the treatment for three years as part of a global trial. It is experimental but experts hope the drug can stop some of the medical complications linked to stunted growth. The researchers behind the work,published in the New England Journal of Medicine, say the goal is to improve health, not just increase height.(Medicine Column)

4) The music duo defying description. Goldsmiths College in London has some pretty famous former music students. Mercury-winner James Blake went there, Blur were formed there (though they were called Seymour at the time) and other well-known graduates include the Velvet Underground’s John Cale and Placebo’s Brian Molko. It’s also the birthplace of the uniquely-named duo Jockstrap (be careful searching it on Google). ”I like the fact that it’s quite shocking,” says singer and violinist Georgia Ellery, who is studying for a degree in jazz music and is one half of the band. ”But it’s quite anonymous. ”I don’t think people really think it’s me and Taylor behind it. It’s just a bit of fun, really. ”Also, there was no Jockstrap on Spotify.”(Entertainment Column)

5) Venezuela crisis: Vast corruption network in food programme, US says. US officials have accused Venezuelan President Nicols Maduro and his allies of profiting from a food subsidy scheme as the country suffers acute shortages. For years, a ”vast corruption network” made money from overvalued contracts while only a fraction of the food for the state-run programme was imported, the US treasury department said. It imposed sanctions on 10 people, including Mr Maduro’s three stepsons. Mr. Maduro called the measures a sign of ”desperation” by ”the gringo empire”. Oil-rich Venezuela has faced chronic shortages of food and medicine as a result of a years-long political and economic crisis, and a large number of people say they do not have access to enough food. (International News Column)

6) North-Korea has developed software designed to teach ideology to party. Members and workers, according to North Korean party daily Rodong Sinmun. Called Chongseo 1.0, it contains writings by the country’s founder Kim Il-sung and his son former leader Kim Jong-il. The paper explains that the encyclopaedia-like electronic book program aggregates classical works and anecdotes about the two Kims, as well as material related to current leader Kim Jong-un. The program works on different devices and operating systems - including Windows and North Korea’s Linux-based Red Star - and the plan is to distribute it nationwide. It won’t stop there. Developers are already working on the next version Chongseo 2.0, adding various functions, including voice reading. (Politics Column)

7) Trump escalates trade war with more China tariffs. US President Donald Trump has said he will impose a fresh 10% tariff on another $300 of Chinese goods, in a sharp escalation of a trade war between the two countries. ”Adding tariffs is definitely not a constructive way to resolve economic and trade frictions, it’s not the correct way,” Mr Wang said on the sidelines of a meeting of South East Asian ministers in Bangkok. Mr. Trump says his trade tactics are working, and that Beijing is feeling the pain. But China isn’t the only country that is hurting. The International Monetary Fund has warned that the US-China trade war is the biggest risk to the global economy. (Business Column)

8) Paisley babysitter jailed for causing baby brain bleed. A man who threw a 12-week-old baby in the air causing bleeding near the brain has been jailed for two years. Charlie Boyle, 23, from Paisley, was babysitting the boy in Neilston, near Glasgow, when he tossed him up three times. The baby was rushed to hospital after becoming floppy and unresponsive. Medical experts told the. High Court in Glasgow that the child, who cannot be named for legal reasons, may have died, but for urgent medical help. The court heard that Boyle had earlier been warned by the baby’s mother about throwing the child in the air. Boyle insisted he was just playing with the child, who he claimed was laughing. (Society News Column)

9) Police vehicle and car in crash on M5 northbound. Avon & Somerset Police said the crash happened at 21:40 BST as officers in a marked car attempted to stop a vehicle linked to a suspect. One officer was taken to hospital with non life-threatening injuries and a 27-year-old man was arrested on suspicion of dangerous driving. Highways England reopened the road shortly after 07:50 but delays remain. One motorist tweeted how they made it home after a ”13 hour journey.” (Society News Column)

10) 1,100 jobs at risk after Karen Millen online deal. More than 1,000 jobs could be at risk after fashion firm Boohoo bought the online business of UK brands Karen Millen and Coast for f18.2m. Boohoo, an online-only retailer, said acquiring the website operations of the two brands ”would represent highly complementary additions”. The firms’ 32 UK High Street stores and 177 concessions, employing 1,100 people, now appear set to close. Administrators Deloitte said the stores would trade for a ”short time”.