Keyword Identification Using Text Graphlet Patterns

Ahmed Ragab Nabhan[1,2](✉) and Khaled Shaalan[3,4]

[1] Faculty of Computers and Information, Fayoum University, Faiyum, Egypt
ahmed.nabhan@gmail.com
[2] Member Technology, Sears Holdings, Hoffman Estates, USA
[3] The British University in Dubai, Dubai, United Arab Emirates
[4] School of Informatics, University of Edinburgh, Edinburgh, UK
khaled.shaalan@buid.ac.ae

Abstract. Keyword identification is an important task that provides useful information for NLP applications including: document retrieval, clustering, and categorization, among others. State-of-the-art methods rely on local features of words (e.g. lexical, syntactic, and presentation features) to assess their candidacy as keywords. In this paper, we propose a novel keyword identification method that relies on representation of text abstracts as word graphs. The significance of the proposed method stems from a flexible data representation that expands the context of words to span multiple sentences and thus can enable capturing of important non-local graph topological features. Specifically, graphlets (small subgraph patterns) were efficiently extracted and scored to reflect the statistical dependency between these graphlet patterns and words labeled as keywords. Experimental results demonstrate the capability of the graphlet patterns in a keyword identification task when applied to MEDLINE, a standard research abstract dataset.

Keywords: Word graphs · Pattern analysis · Graph features · Machine learning · MEDLINE

1 Introduction

Keywords are important meta data that is useful for many document processing tasks including indexing and retrieval, clustering, and classification. Keywords can act as relevancy indicators about documents that are retrieved given a search query. This meta data can be provided by authors, or by manual or automatic methods to provide better search and access experience when using scientific databases. Lists of keywords, as document fields, are of high quality considering they are provided manually either by authors or assigned by librarians. Although these lists are typically limited in size (three to five items), there is an opportunity to utilize them through supervised learning methods that aim to identify statistical dependencies between text features and word labels (e.g. keyword vs. non-keyword).

E. Métais et al. (Eds.): NLDB 2016, LNCS 9612, pp. 152–161, 2016.
DOI: 10.1007/978-3-319-41754-7_13

The problem of automatic extraction of keywords within text has been addressed through both NLP and graph-based methods. Several resources including manually-generated keywords, lexical, and syntactic annotations have been used to identify keywords within texts [1–5]. Machine learning techniques have been developed to extract useful features for the task of keywords identification. Witten et al. (1999) proposed a Naive Bayes method for learning features relevant to keyword extraction, see also [6]. Li et al. developed a semi-supervised approach to learning an indicator by assigning a value to each important phrase in the document title and to propagate that value to other phrases throughout the document [7]. Tomokiyo (2003) proposed an approach to key phrase extraction based on language models (cf. [8]).

Automatic methods for keyword identification utilize lexical, syntactic, semantic, and presentation features of texts. These features can be used to learn rules for identifying keywords in testing data. Counting-based methods (e.g. TF-IDF [9]) are built on lexical features (e.g. stemmed, lemmatized, or surface forms of the words). Morpho-Syntactic features (e.g. part-of-speech tagging information, base-phrase chunks) can be used as a filter to exclude words (e.g. adverbs and prepositions) or phrases (e.g. verb phrases) unlikely to be keywords.

Graph-based document representation can enable powerful methods such as random walks and frequent subgraph mining to capture relevant features for many NLP tasks. Ranking-based graph methods have been proposed to quantify the importance of a word in a document relative to neighboring words. TextRank [10], LexRank [11], and NeRank [12] are examples of methods that employed graph properties such as vertex centrality and ranking for keyword extraction and text summarization. Mining of frequent subgraphs and subtrees were demonstrated to be useful for document categorization [13]. Graph-based term weighting methods were proposed for information retrieval applications [14].

In this paper, a graph pattern mining method is proposed to identify keywords in word graphs constructed from text abstracts. Word graphs provide a flexible representation that enables exploration of complex, non-local features that can span multiple sentences. The method combined lexical features with graph substructures (e.g. graphlets) to explore contexts of keywords and identify siginificant patterns. Then, these graph-based features were used by a machine learning based classifier for testing data.

2 Research Methods

The problem being addressed in this study can be defined as follows. Let a text document be mapped to a word graph where vertices represent words, and edges indicate word co-occurrences in sentences. That is, if the words w_i and w_j appear next to each other in a sentence, an edge is created to link the two vertices representing w_i and w_j. Given a set of word graphs $D = \{G_1, G_2, ..., G_n\}$ where each graph G_i represents a text document, the problem is to structurally analyze elements in D to extract significant graph substructures (e.g. graphlets, paths) in the neighborhood of each word vertex. Then, a binary classifier can be used to classify word vertices in a test corpus as keywords or non-keywords based on their graph substructure features.

Vertex labels in word graphs can represent content at some linguistic level. For instance, at the very basic level, a vertex can correspond to a primitive lexical unit (e.g. surface, lemmatized or stemmed word form), syntactic unit (e.g. a phrase) or a sentence (in which case a graph can represent a whole corpus). Edges in these graphs can represent a lexical relationship (e.g. neighborhood, lexical similarity) or syntactic relationships (e.g. subject-verb-object relationships). In this study, text abstracts were mapped to graphs where vertices represented words and edges between a pair of vertices were created if the pair of vertices appeared next to each other in a sentence. The edges did not have weights and the vertices had two features: a lexical label and a part-of-speech tag.

2.1 Notations

A graph is a structure $G = (V, E)$, where $V = \{v_1, v_2, ..., v_n\}$ is a set of vertices and $E \subseteq V \times V$ is the set of edges. A graph is said to be undirected iff $\forall e = (v_i, v_j) \in E, \exists e' = (v_j, v_i) \in E$. A graph $G(V, E)$ is isomorphic to graph $G'(V', E')$ if there exists a bijective function $f\colon V \rightarrow V'$ such that $e = (v_i, v_j) \in E$ iff $e'(f(v_i), f(v_j)) \in E'$. Graph automorphism is a symmetry property of a graph so that its vertex set can be mapped into itself while preserving edge-vertex connectivity. A subgraph $H(V', E')$ of G is defined such that $V' \subseteq V$ and $E' \subseteq E$. The size of a graph or a subgraph is defined as the number of items in the vertex set.

Graphlets are small-sized nonisomorphic subgraphs (typically $2 \leq |V| \leq 5$) within a given larger graph [15]. A graph can be characterized by its collection of graphlets [16]. Graphlet automorphism allows for modeling the relationship of a graphlet and its component vertices. *Automorphism orbits* are defined by distances of a vertex of interest (pivot) and the rest of vertices in a graphlet. Each vertex reachable from a pivot vertex p in a number of edges d are said to be in $d - orbit$.

2.2 The Graph-Based Method

In this study, we present a graph-based method for identification of keywords in research abstracts. Text preprocessing operations (e.g. removing punctuation marks and stop words, lower-casing) were applied. Text were processed to extract bigram patterns and these patterns were used to represent edges with end points representing vertices. The NLTK toolkit [17] part-of-speech tagger provided syntactic information that were used during classification to apply the graphlet methods only to vertices with nouns and adjectives part of speech tags. These helped in increasing the true negative (non-keywords) score.

The method relied on graphlets that were extracted from a word graph representation of the abstracts. For each word vertex, a graph search algorithm was designed to extract 3-graphlet and 4-graphlet subgraphs such that this word was a pivot vertex. The relationships between the pivot vertex (a candidate keyword)

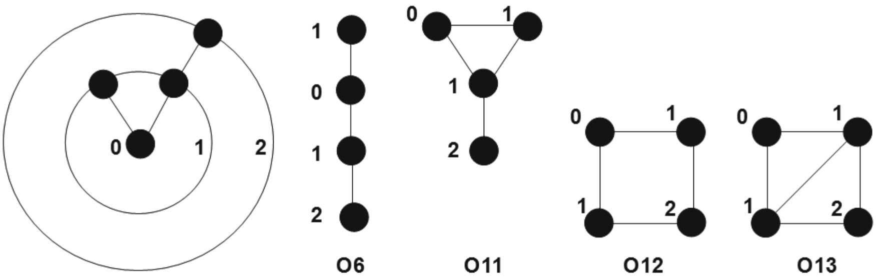

Fig. 1. An example 4-graphlet with four possible orbit forms of the configuration $< 0, 1, 1, 2 >$.

and its neighbors are defined based on orbit positions and inter-edges between these non-pivot vertices.

Graphlets can be enumerated systematically and thus are efficient to extract. *Automorphism orbits* allow for efficient comparison and counting of graphlets and thus they can be efficiently explored and scored. A string representation of graphlet determines (1) the layout of neighboring vertices in each orbit, and (2) an indicator of the configuration (e.g. how other non-pivot vertices are linked). Illustration of orbit configurations are given in Figs. 1 and 2. For instance, a 4-graphlet orbit 0 1 2 2 (Fig. 2) represents a pivot vertex at the origin (0) and the next neighbor is at distance 1, and then two vertices are connected to that neighbor at a two-edges distance from the pivot. In this case, two graphlet forms can be possible: (1) a form where the neighbors-of-a-neighbor can be linked by one edge, or (2) another form where there are no edges between those neighbors-of-a-neighbor.

In this paper, we followed the methodology proposed by Vacic et al. [16] for enumerating and counting graphlets using the configuration-form scheme. Table 1 shows the graphlet orbit configuration and form schemes used in this study. During feature extraction, graphlet patterns were discovered systematically by applying each scheme in Table 1, starting with a pivot vertex (the word for which the features were extracted). Two trivial graphlets (namely: 1-graphlet and 2-graphlets) were not used in our study.

Figure 1 shows four varieties of graphlets of the same configuration, only differing by the way non-pivot vertices are linked. The feature extraction algorithm stores and retrieves graphlets using the configuration string (e.g. $< 0, 1, 2, 2 >$), a named form (e.g. O7), together with labels (words surface forms) of the vertices ordered by the configuration string. Hash function values were generated using these compact string representations of graphlets and were used to read from and write to the feature table where features frequency were stored.

In this study, 3-graphlet and 4-graphlet subgraphs were used as features for classification of words as keywords vs. non-keywords. A minimum support value of 5 was used to reduce feature space size. Graphlets of size 5 were not used due to their somewhat large number of orbit configurations that would increase

Table 1. Orbit configuration and forms of 2-graphlet, 3-graphlet and 4-graphlet patterns

Orbit configuration	Form	Description
$< 0, 1 >$	O_1	Simple edge from pivot vertex to non-pivot vertex
$< 0, 1, 1 >$	O_3	Two edges drawn from pivot vertex to two non-pivot vertices that are not connected
$< 0, 1, 1 >$	O_4	Two edges drawn from pivot vertex to two non-pivot vertices that are connected (all three forms a triangle)
$< 0, 1, 2 >$	O_2	A simple path connecting pivot vertex to two other non-pivot vertices
$< 0, 1, 1, 1 >$	O_8	A star-like shape where pivot vertex is in the middle and no edges between non-pivot vertices
$< 0, 1, 1, 1 >$	O_{10}	A pivot vertex is in the middle and there exists one edge between a pair of the non-pivot vertices
$< 0, 1, 1, 1 >$	O_{14}	A pivot vertex is in the middle and there exists two edges between two pair of the non-pivot vertices
$< 0, 1, 1, 1 >$	O_{15}	A pivot vertex is in the middle and all non-pivot vertices connected to each other
$< 0, 1, 1, 2 >$	O_6	A simple path connecting four vertices and the pivot is any non-terminal vertex in that path (Fig. 1)
$< 0, 1, 1, 2 >$	O_{11}	A pivot vertex is connected to two non-pivot and a third vertex is connected to either of the non-pivot vertices (Fig. 2)
$< 0, 1, 1, 2 >$	O_{12}	A pivot vertex is connected to two non-pivot and a third vertex is connected to both of the non-pivot vertices (Fig. 1)
$< 0, 1, 1, 2 >$	O_{13}	A pivot vertex is connected to two non-pivot and a third vertex is connected to both of the non-pivot vertices (Fig. 1)
$< 0, 1, 2, 2 >$	O_7	A pivot vertex is connected to one non-pivot which is connected to two non-pivot vertices that are not connected (Fig. 2)
$< 0, 1, 2, 2 >$	O_9	A pivot vertex is connected to one non-pivot which is connected to two non-pivot vertices that are connected via an edge (Fig. 2)
$< 0, 1, 2, 3 >$	O_5	A simple path connecting the four vertices where the pivot is at either ends

the running time of the algorithm. A conditional probability model was used to quantify the correlation between graphlet features and class labels (keywords vs. non-keywords). Let $g_1, g_2, ..., g_n$ be a set of graphlets containing a given word v_i (as a pivot vertex) and that word has a class label c_k. The problem of assigning

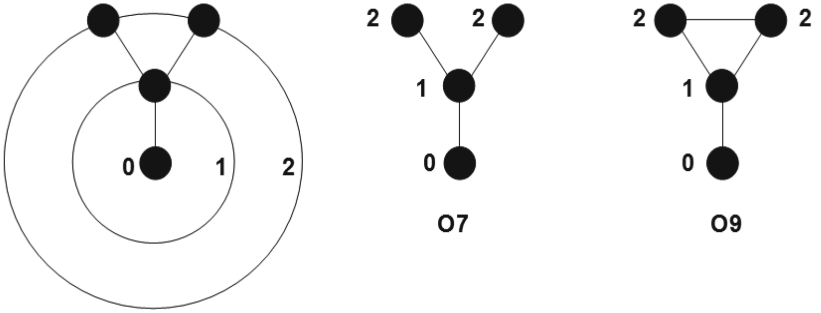

Fig. 2. An example 4-graphlet with two possible orbit forms of the configuration $< 0, 1, 2, 2 >$. There are two cases with the outer vertices on orbit 2: first case is where no edges between these vertices and second case where there is an edge linking them.

a class label c_k to a word v_i given a set of graphlets with v at the origin can be stated as estimating the probability value $P(c_k|v_i, g_1, g_2, ..., g_n)$, as a function of graphlet features. Using Bayes' rule,

$$\begin{aligned} P(c_k|v_i, g_1, g_2, ..., g_n) &\propto P(c_k) \times P(v_i, g_1, g_2, ..., g_n|c_k) \\ &= P(c_k) \times P(v_i|c_k) \times P(v_i, g_1, g_2, ..., g_n|c_k) \\ &= P(c_k) \times P(v_i|c_k) \times \prod_{i=1}^{n} P(g_i|c_k), \end{aligned} \tag{1}$$

assuming graphlet features are conditionally independent on class label c_k. The probability value $P(c_k)$ represents prior information about class distribution in the data set. The probability model $P(v_i|c_k)$ can be viewed as a unigram probability of word vertex v_i. During initial phases of the study, there was no observed performance improvements of using the unigram probability $P(v_i|c_k)$ and thus this probability value was not used in the final model. Finally, a label was assigned to a word such that it maximized the probability value that is a function of graphlet features:

$$\hat{y} = \underset{k \in \{1,2\}}{\text{argmax}} P(c_k) \times \prod_{i=1}^{n} P(g_i|c_k) \tag{2}$$

Using the above model, a Naive Bayes classifier was used to decide on whether or not a test word vertex represented a keyword, given graphlets that were explored for that test word vertex.

2.3 TF-IDF Method

In this study, TF-IDF scores were used in the baseline system. TF-IDF is a frequency-based method that takes into account word as well as document frequencies [18]. For a given word i in a document j, the w_{ij} score was computed as follows.

$$w_{ij} = \frac{n_{ij}}{d_j} log_2 \frac{N}{n_i}, \tag{3}$$

where n_{ij} is the frequency of word i in document j, $|dj|$ is total words in document j, N is the total number of documents in the corpus, and n_i is the size of set of document containing word i. After computing w_{ij} for all words, a normalized score was computed as

$$W_{ij} = \frac{w_{ij}}{\sqrt{\sum_i wij}} \tag{4}$$

The TF-IDF score was computed for words after applying the syntactic filter where words other than nouns and adjectives were excluded. Words were ranked according to the score and top five high scored words were labeled as keywords.

3 Experiments and Results

3.1 Data Description

The dataset of the present study was drawn from MEDLINE research abstract database. MEDLINE is a high quality medical publication database supported by the National Library of Medicine (NLM) in the United States. One of the fields of an abstract record is the Other Term (OT) field that indicates author-supplied list of keywords. Not all MEDLINE abstracts have that field and the data was collected such that only abstracts with non-empty OT fields were included in the test dataset. A corpus of around 10,000 abstracts was collected. Text preprocessing steps were applied. For each abstract, a word graph was constructed and 3-and 4-graphlets were extracted for every word that pass the syntactic filter (only nouns and adjectives were considered). Other word types (e.g. adverbs, conjunctions) were included in the graphs to serve as hubs to connect other words, but no specific graphlets are generated for these category of words.

3.2 Performance Evaluation

A 5-fold cross validation was conducted for two systems. A system based on TF-IDF method was implemented as a baseline. The evaluation of both experimental and baseline systems were based on top five scoring candidate keywords. For the experimental model, for each document, the five top scoring keywords in the positive class (keyword) were considered if ($\hat{y}$) probability score is higher than that value of the negative class (non-keyword) according to Eq. 2. When, for a test document, the top five model scores given positive class labels were lower than scores corresponding to negative class labels, no keyword was generated.

A frequent pattern mining system was implemented to extract 3- and 4-graphlets based on a minimum support value of five (a graphlet has to appear in five documents to be selected as a feature). A probability model of graphlets conditionally independent on word class label (keyword vs. non-keyword) was build

Table 2. Keyword classification accuracy

System	Recall	Precision
TFIDF	0.55	0.73
graph-based	0.6	0.79

Table 3. A sample of graphlet patterns

Orbit Configuration	Form	Vertex Labels
$< 0, 1, 2 >$	O2	vascular,endothelial,growth
$< 0, 1, 2, 2 >$	O7	control,and,cervical,prevention
$< 0, 1, 2, 2 >$	O7	cervical,cancer,early,screening
$< 0, 1, 2, 2 >$	O7	overall,survival,(pfs),(os)
$< 0, 1, 1, 2 >$	O6	cancer,early,screening,detection

by counting pattern occurrences and normalization. A uniform prior probability (0.5) was used for $P(c_k)$. A small probability value $\epsilon = 1 \times 10^{-5}$ was used for unseen graphlet parameters. Recall and precision scores were computed based on the five top scoring candidate keywords in each test document. Performance results for the two systems are shown in Table 2.

As a by-product of the feature extraction process, a number of significant patterns were discovered. Here, we present some of the graphlet patterns that achieved high scores. A number of these patterns is shown in Table 3.

4 Discussion and Conclusions

This study addressed the problem of keyword identification in scientific abstracts using a novel graph-based method. An important advantage of the proposed method is the use of efficient enumeration of significant graphlet patterns in word graphs. A Bayes' rule-based model was applied to measure the statistical dependency between significant graphlet patterns and word class labels (e.g. keywords vs. non-keywords labels). The pivot vertices in the graphlets feature set represent candidate keywords. The proposed method was compared to standard, frequency-based method (TF-IDF) and experimental results showed competitive performance in terms of Recall and Precision.

This study advances the state-of-the-art of keyword identification methods by introducing an efficient method for exploring significant patterns in word graphs. Graph-based representation of text has an important advantage over traditional methods of text analysis. These graphs allow access to a wider context of the word vertex and hence enables capturing of non-local dependencies that might not be straightforward to obtain through syntactic and lexical analysis.

A major contribution of this study is an efficient algorithm for mining significant patterns in word graphs. While graph-based algorithms are known to be computationally expensive, the methods presented in this study enabled calculation of pattern frequency without resorting to testing of graph isomorphism. This was possible through usage of an orbit-based scheme representation of graphlets. This graphlet-based method is different from sequence-based methods (e.g. n-gram models) in two aspects. First, one graphlet pattern may contain words from multiple sentences in a given utterance (e.g. a text abstract), whereas sequence-based methods only contain words that co-occur within sentences. Second, sequence-based methods cannot capture some subtle topological patterns of a candidate keyword (e.g. as illustrated by the patterns shown in Fig. 1).

The presented method utilized a simple text structure, that is of words co-occurrences, to identify vertices and edges of text graphs. This simplicity of graph representation ignores the syntactic structure of sentences. One way to capture syntactic relationships is to identify graph vertices at a multi-word or phrase level (e.g. noun phrases). However, this might affect graph topology in a way that can prevent capturing of some significant, micro-level (e.g. word level) graph substructures. Another limitation of the presented method is lack of edge information (i.e. edges were unlabeled, unweighted). One potential future enhancement of the work would include, in addition to adjacency, some information such as bigram frequency and syntactic dependency.

While the method was designed to solve the problem of identification of word vertices that are keywords, as a by-product, a set of significant graphlet patterns are produced. These patterns can be useful in other tasks such as exploration, text categorization, and document clustering. The proposed probability model allows for quantification of pattern importance and thus these patterns are already measured as significant. Future work includes designing a text categorization method that rely on text graphlets.

References

1. Andrade, M.A., Valencia, A.: Automatic extraction of keywords from scientific text: application to the knowledge domain of protein families. Bioinformatics **14**(7), 600–607 (1998)
2. Matsuo, Y., Ishizuka, M.: Keyword extraction from a single document using word co-occurrence statistical information. Int. J. Artif. Intell. Tools **13**(1), 157–169 (2004)
3. Hammouda, K.M., Matute, D.N., Kamel, M.S.: CorePhrase: keyphrase extraction for document clustering. In: Perner, P., Imiya, A. (eds.) MLDM 2005. LNCS (LNAI), vol. 3587, pp. 265–274. Springer, Heidelberg (2005)
4. Hasan, K.S., Ng, V.: Automatic keyphrase extraction: a survey of the state of the art. In: ACL, pp. 1262–1273 (2005)
5. Daiber, J., Jakob, M., Hokamp, C., Mendes, P.N.: Improving efficiency and accuracy in multilingual entity extraction. In: Proceedings of the 9th International Conference on Semantic Systems, pp. 121–124. ACM (2013)
6. Witten, I.H., Paynter, G.W., Frank, E., Gutwin, C., Nevill-Manning, C.G.: KEA: practical automatic keyphrase extraction. In: Proceedings of the Fourth ACM Conference on Digital Libraries, pp. 254–255. ACM (1999)

7. Liu, F., Pennell, D., Liu, F., Liu, Y.: Unsupervised approaches for automatic keyword extraction using meeting transcripts. In: Proceedings of Human Language Technologies, pp. 620–628. Association for Computational Linguistics (2009)
8. Tomokiyo, T., Hurst, M.: A language model approach to keyphrase extraction. In: Proceedings of the ACL 2003 Workshop on Multiword Expressions: Analysis, Acquisition and Treatment, vol. 18, pp. 33–40. Association for Computational Linguistics (2003)
9. Zhang, Y., Zincir-Heywood, N., Milios, E.: Narrative text classification for automatic key phrase extraction in web document corpora. In: Proceedings of the 7th Annual ACM International Workshop on Web Information and Data Management, pp. 51–58. ACM (2005)
10. Mihalcea, R., Tarau, P.: TextRank: Bringing Order into Texts. Association for Computational Linguistics, Stroudsburg (2004)
11. Erkan, G., Radev, D.R.: LexRank: graph-based lexical centrality as salience in text summarization. J. Artif. Intell. Res. **22**, 457–479 (2004)
12. Bellaachia, A., Al-Dhelaan, M.: Ne-rank: a novel graph-based keyphrase extraction in twitter. In: Proceedings of the 2012 IEEE/WIC/ACM International Joint Conferences on Web Intelligence and Intelligent Agent Technology, vol. 01, pp. 372–379. IEEE Computer Society (2012)
13. Jiang, C., Coenen, F., Sanderson, R., Zito, M.: Text classification using graph mining-based feature extraction. Knowl. Based Syst. **23**(4), 302–308 (2010)
14. Blanco, R., Lioma, C.: Graph-based term weighting for information retrieval. Inf. Retrieval **15**(1), 54–92 (2012)
15. Pržulj, N.: Biological network comparison using graphlet degree distribution. Bioinformatics **23**(2), e177–e183 (2007)
16. Vacic, V., Iakoucheva, L.M., Lonardi, S., Radivojac, P.: Graphlet kernels for prediction of functional residues in protein structures. J. Comput. Biol. **17**(1), 55–72 (2010)
17. Bird, S.: NLTK: the natural language toolkit. In: Proceedings of the COLING/ACL on Interactive Presentation Sessions, pp. 69–72. Association for Computational Linguistics (2006)
18. Aizawa, A.: An information-theoretic perspective of TF-IDF measures. Inf. Process. Manage. **39**(1), 45–65 (2003)