Keyphrase Detection

(Spring 2017- Natural Language Processing)

Project Report by

Frenia Pinto

Instructor

David A. Smith

Abstract

Keyphrase extraction is the task of identifying single or multi-word expressions that represent the main topics of a document. Keyphrases are useful in many tasks such as information retrieval, document summarization or document clustering. Although scientific articles usually provide them, most of the documents have no associated keyphrases. Therefore, the problem of automatically assigning keyphrases to documents is an active field of research. In this project, we will evaluate the statistical approach such as TFIDF and graph approach known as the TopicRank on a corpus of technical documents.

Keyphrases: keyphrase extraction, tfidf, topicrank, graph

1. Introduction

Keyphrases are single or multi-word expressions that represent the main topics of a document. Automatic keyphrase extraction methods are divided into two categories: supervised and unsupervised methods. Supervised methods recast keyphrase extraction as a binary classification task, whereas unsupervised methods apply different kinds of techniques such as language modeling, clustering or graph-based ranking. While supervised approaches have generally proven to be more successful, the need for training data and the bias towards the domain on which they are trained remain critical issues.

The keyphrase extraction implementation can be described by the following steps:

1. Preprocessing the document
2. Candidate keyphrase extraction
3. Selection of relevant keyphrases

The corpus used for this project is the SemEval2000 corpus which has 144 technical documents. Link: https://github.com/snkim/AutomaticKeyphraseExtraction. For evaluations, we will be using the stemmed version of author and reader assigned keyphrases.

2. Literature Review

One of the algorithm that implements keyword extraction is the Rapid Automatic Keyword Extraction (RAKE) which is based on the observation that keywords frequently contain multiple words but rarely contain standard punctuation or stop words, such as the function words *and*, *the*, *of* or other words with minimal lexical meaning. RAKE begins keyword extraction on a document by parsing its text into a set of candidate keywords. First, the document text is split into an array of words by the specified word delimiters. This array is then split into sequences of contiguous words at phrase delimiters and stop word positions. Words within a sequence are assigned the same position in the text and together are considered a candidate keyword. After every candidate keyword is identified and the graph of word co-occurrences is complete, a score is calculated for each candidate keyword and defined as the sum of its member word scores. The

evaluated metrics for calculating word scores, based on the degree and frequency of word vertices in the graph:
(1) word frequency (number of occurrences),
(2) word degree (sum of lengths of the candidate keyphrase in which the word occurs)
and (3) ratio of degree to frequency.
Hence, frequency favors frequently occurring words and degree favors words that occur in longer candidate words.
Consider the abstract for the document I-54.txt. The candidate keywords along with their scores generated are:

{'negotiation': 2.2857142857142856, 'equilibrium offers': 4.2, 'different time points': 9.1, 'show': 1.0, 'present negotiation strategies': 8.035714285714285, 'pie': 1.0, 'approximate equilibrium strategies': 7.15, 'agents': 2.6, 'also analyze': 4.5, 'maximize': 1.0, 'parties': 1.0, 'polynomial time complexity categories': 13.1, 'allocated': 1.0, 'extend': 1.0, 'optimum': 1.5, 'relative error ie': 10.0, 'viewed': 1.0, 'approximately optimal': 4.0, 'difference': 1.5, 'valuations': 1.0, 'equilibrium': 2.2, 'subject descriptors': 4.0, 'complete information setting': 9.0, 'approximate': 2.2, 'deadlines': 1.0, 'onm': 1.0, 'relative error finally': 9.0, 'entirety': 1.0, 'priori': 1.0, 'allocate': 1.0, 'issues specifically': 4.75, 'finding': 1.0, 'negotiation deadline': 4.285714285714286, 'issue': 1.5, 'true optimum': 3.5, 'overcome': 1.0, 'split': 1.0, 'selfinterested autonomous agents': 8.6, 'form': 1.0, 'equilibrium strategies': 4.95, 'discount factors': 4.0, 'issues': 2.75, 'computationally efficient': 4.0, 'analyse': 1.0, 'decide': 1.0, 'known': 1.0, 'uncertain': 1.0, 'time complexity': 5.1, 'time constraints': 4.6, 'approximate strategies also': 7.45, 'case': 1.0, 'first obtain': 4.0, 'computational complexity': 4.5, 'size one': 4.0, 'different issues become available': 14.25, 'individual utilities': 5.5, 'online negotiation': 4.285714285714286, 'analysis': 1.0, 'n': 1.0, 'approximate equilibrium exists': 7.4, 'expected difference': 3.5, 'either agent': 4.0, 'np-hard problem even': 8.0, 'problem': 2.0, '-negotiations': 4.0, 'order': 1.0, 'issue must': 3.5}

On analyzing the keyphrases we find that the longer the candidate keyphrases, the higher is the calculated scores. This is because for each term in the candidate word, RAKE sums the individual term score (ratio of degree to frequency). Also, since RAKE splits the word at stop words, candidate keyphrases such as *gain from cooperation* are not extracted at the candidate extraction step.

3. Implementation and Discussion

Preprocessing: Since the SemEval2000 corpus is a corpus of 144 technical documents and most of the relevant keyphrases that describes the content of each document is present in the abstract of the document, we consider only the content of the abstract to retrieve candidates keyphrases that can be used for extraction.

The steps performed in this stage are:

- Convert the text to lowercase.
- Remove all numbers from the text.
- Remove all non-ASCII characters (maybe used to describe a formula variable)
- Remove punctuations except - (eg. multi-agent is a more sensible keyword than multi and agent separately) and , , . (Since, they are used for POS tagging)

Candidate extraction: The RAKE approach identified candidate keywords by splitting at stop words and punctuation. Hulth (2003) stated that most keyphrases assigned by human readers are noun phrases. Hence, the most important topics of a document can be found by extracting their most significant noun phrases.

Interesting n-grams could also be chosen. However, deciding the length of n-gram is a constraint. Initially, it was decided to use unigram, bigram and trigram as candidate keywords since most of the keyphrases generated are mainly 1,2,3-gram. However, there is an overhead of additional computations. Also, each of these n-grams have to be handled separately. There is a lot of redundant keyphrases generated.

Eg. Consider the document C-44.txt:

One of the trigram generated is wireless sensor node which appears 2 times in the abstract. The bigram sensor node appears thrice. If any statistical based approach were to be used to extract keyphrases, mostly of relevant phrases generated would be node , sensor , sensor node , wireless sensor node which could be described by just one wireless sensor node phrase. Thus, allowing other candidates to feature in the relevant keyphrases set.

Each sentence from the abstract is sent to a POS tagger function which generates keyphrases that have the following POS patterns:

Named Entity (Noun), Adjective+ Named Entity (Noun), Named Entity + (Adjective/Verb) + Named Entity.

Consider the candidates generated for the document C-62.txt:

['todays', 'industry suffers from several well-known pathologies', 'none', 'long term', 'resistance', 'evolution', 'new services', 'isps', 'towards', 'commoditization', 'networks primitive system of contracts', 'incentives', 'study', 'networks lack of accountability', 'fundamental obstacle', 'problem',

'economic model', 'optimal routes', 'innovation', 'new monitoring capability', 'contracting system', 'minimum requirements', 'monitoring system', 'first-best routing', 'innovation characteristics', 'work', 'new protocol', 'specific guidance', 'design', 'systems', 'theoretical framework', 'factors', 'influence innovation', 'categories', 'subject descriptors', 'computer-communication networks', 'systems', 'behavioral sciences economics general terms economics', 'theory', 'measurement', 'design', 'legal aspects']

After the candidate keyphrases for each of the document are retrieved, each keyphrase is stemmed. The main reason to perform stemming is to avoid computations on similar words differently. For example, wireless sensor node and wireless sensor nodes are the same, but the latter is plural.

Selection of keyphrases: The two approaches used in this project are:

- **Statistical method TFIDF**

The list of candidate keyphrases that are generated for each of the document in the corpus from the previous step is used for the computation of the TF*IDF for each candidate.

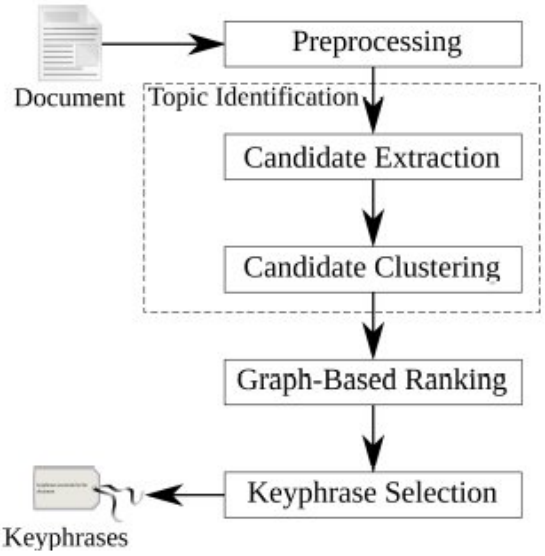
TF(t) = (Number of times term t appears in a document) / (Total number of terms in the document)

IDF(t) = log (Total number of documents / Number of documents with term t in it)

The candidates are then sorted by increasing TF*IDF values and only those n-candidates with top TF*IDF scores above average values.

- **Graph method TopicRank**

The diagrammatic representation of the TopicRank approach is as follows:



After the candidate keyphrases are retrieved, they have to be clustered into the topic they represent. A model could be trained to cluster each candidate keyphrase into individual topics using supervised approach. However, the approach used is to cluster each of the keyphrases if they have overlapping words. In this project, we have considered unigram and bigrams of keyphrases (Top 1/4th frequent bigrams) to represent topics of each cluster.

Consider the keyphrase: wireless sensor node

Topics: wireless, sensor, node, wireless sensor, sensor node. (We consider an imaginary START and END symbol before and end of each keyphrase, but exclude in our bigram model)

We construct the graph as follows: Let G = (V, E) be a complete and undirected graph where V is a set of vertices and the edges E a subset of V ×V. Vertices are topics and the edge between two topics ti and tj is weighted according to the strength of their semantic relation. ti and tj have a strong semantic relation if their keyphrase candidates often appear close to each other in the document. Therefore, the weight w (i, j) of their edge is defined as follows:

$$w_{i,j} = \sum_{c_i \in t_i} \sum_{c_j \in t_j} \text{dist}(c_i, c_j)$$

$$\text{dist}(c_i, c_j) = \sum_{p_i \in \text{pos}(c_i)} \sum_{p_j \in \text{pos}(c_j)} \frac{1}{|p_i - p_j|}$$

where dist (ci, cj) refers to the reciprocal distances between the offset positions of the candidate keyphrases ci and cj in the document and where pos (ci) represents all the offset positions of the candidate keyphrase ci.

Only those candidate keyphrases in cluster topics which have weights>0 are selected to be relevant keyphrases of the document.

4. Results

For the technical document J-37.txt:

Author+Reader assigned stemmed keyphrases:

equilibrium, sequenti game, imperfect inform, comput game theori, order game isomorph, relat order game isomorph abstract transform, order signal space, observ action, nash equilibrium, gameshrink, signal tree, game theori, norm framework, ration behavior, strategi profil, sequenti game of imperfect inform, autom abstract, equilibrium find, comput poker

TFIDF stemmed keyphrases:

fundament problem in comput game theori, observ action, close-to-optim strategi, gameshrink, poker game, f. theori of comput, nontrivi game, relat order game isomorph abstract transform, sever electron commerc applic for gameshrink, nash equilibrium, space complex, multi-play sequenti game of imperfect inform, transform,number of node,approxim method, origin game, equilibrium, larg game, game, game tree, yield ex post, signal tree, order signal space, order game isomorph, extens form game of imperfect inform, isomorph (P = 0.307)

TopicRank stemmed keyphrases:

size, nash equilibrium, origin game, equilibrium, gameshrink, poker game, fundament problem in comput game theori, game, game tree, relat order game isomorph abstract transform, extens form game of imperfect inform, multi-play sequenti game of imperfect inform, sever electron commerc applic for gameshrink, order game isomorph (P = 0.26)

For the technical document I-53.txt:

Author+Reader assigned stemmed keyphrases:

coalit format, interact, multi-agent system, cooper game theori, shaplei valu, uniqu and fair solut, polynomi time, mean of reach consensu+reach consensu mean, randomis method, gener function, approxim, game-theori

TFIDF stemmed keyphrases:

polynomi time, problem of comput complex, main problem, number of player, shapley valu, approxim shapley valu, gener case, empir studi, vote game, method,time complex, specif vote game, fair solut, mani coalit game, other gener vote game, key solut concept for coalit game, percentag error, main advantage (P = 0.05)

TopicRank stemmed keyphrases:

problem of comput complex, paper, design, main problem, problem, method (P = 0.0)

From the results of the I-53.txt, we can deduce that many a times, the abstract of a technical document does not give much idea about the topic (or keyphrases) since the reader+author have assigned different keyphrases which are not present in the abstract. Also, the keyphrases from TFIDF provide similar idea what the reader+author have to say. Instead of complete matching of keyphrases, topic matching would give us high precision values.

The precision and recall values (expressed in percentages) for the two approaches are described as follows:

Approach	**Precision**	**Recall**
TF*IDF	10.93	14.07
TopicRank	10.06	4.7

On closely inspecting the results of precision and recall values of both the approaches, the count of values of precision =0.0 and recall =0.0 is more in the TopicRank approach than the TFIDF approach.

The values of precision and recall for TopicRank could be improved if we would use the entire technical document to generate candidate words to be added to each of the topic clusters. Adding more words to the cluster would add more co-occurrence values and thus, add more weights to the edges associated with each of the clusters.

5. Future Work

In future work, we will implement a supervised approach to cluster keyphrases into topics. This would eliminate redundant keyphrases in other clusters and improve cluster quality. Also, we will evaluate the performance of other approaches such has TextRank, SingleRank and other baseline approaches available with multiple corpus. We will also propose approaches to select a candidate keyphrase that best represents the topic instead of returning all candidates keyphrases in the topic cluster.

6. References

- Stuart Rose, Dave Engel, Nick Cramer and Wendy Cowley, Automatic keyword extraction from individual documents

- Martin Dostal and Karel Jezek, Automatic Keyphrase Extraction based on NLP and Statistical Methods, *V. Snasel, J. Pokorny, K. Richta (Eds.): Dateso 2011, pp. 140{145, ISBN 978-80-248-2391-1*.

- Adrien Bougouin and Florian Boudin and Beatrice Daille, TopicRank: Graph-Based Topic Ranking for Keyphrase Extraction, *International Joint Conference on Natural Language Processing.*