Information Processing and Management 54 (2018) 888–902

Contents lists available at ScienceDirect

Information Processing and Management

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

Local word vectors guiding keyphrase extraction

Eirini Papagiannopoulou*, Grigorios Tsoumakas

School of Informatics, Aristotle University of Thessaloniki, 54124, Greece

ARTICLE INFO

Keywords:
Keyphrase extraction
Unsupervised method
GloVe
Local word vectors
Reference vector algorithm

MSC:
68T50

ABSTRACT

Automated keyphrase extraction is a fundamental textual information processing task concerned with the selection of representative phrases from a document that summarize its content. This work presents a novel unsupervised method for keyphrase extraction, whose main innovation is the use of *local* word embeddings (in particular GloVe vectors), i.e., embeddings trained from the single document under consideration. We argue that such local representation of words and keyphrases are able to accurately capture their semantics in the context of the document they are part of, and therefore can help in improving keyphrase extraction quality. Empirical results offer evidence that indeed local representations lead to better keyphrase extraction results compared to both embeddings trained on very large third corpora or larger corpora consisting of several documents of the same scientific field and to other state-of-the-art unsupervised keyphrase extraction methods.

1. Introduction

Keyphrase extraction is concerned with the selection of a set of phrases from within a document that together summarize the main topics discussed in that document (Hasan & Ng, 2014). Automatic keyphrase extraction is a fundamental task in digital content management as it can be used for document indexing, which in turns enables calculating semantic similarity between documents (and hence document clustering), and can improve browsing of digital libraries (Gutwin, Paynter, Witten, Nevill-Manning, & Frank, 1999; Witten, 2003). In addition, automatic keyphrase extraction offers an approach to document summarization. Keyphrase extraction is particularly important in academic publishing, where it is used as a technological building block to recommend articles to readers, to highlight missing citations to authors and to analyze research trends (Augenstein, Das, Riedel, Vikraman, & McCallum, 2017).

Supervised machine learning approaches for automatic keyphrase extraction rely on annotated corpora. However, manual selection of the keyphrases of each document by humans requires the investment of time and money and is characterized by great subjectivity. In many cases, the extracted keyphrases cover one or more non-core topics due to misunderstandings, or they miss one or more of the important topics discussed in the document. Using multiple annotators can partially address the problem of subjectivity by collecting more keyphrases (Chuang, Manning, & Heer, 2012; Sterckx, Caragea, Demeester, & Develder, 2016). This, however, comes at the expense of additional annotation effort.

In addition, supervised methods often fail to generalize well to documents coming from a different content domain than the training corpus, may require retraining to address concept drift, and are more susceptible to varying vocabularies across documents and different personal writing styles across authors.

In contrast, this work takes a novel unsupervised path to keyphrase extraction. To be able to take into account the semantic similarity among words we consider word embeddings, in particular the one generated by GloVe (Pennington, Socher, & Manning, 2014). Different however from past approaches that exploit word embeddings in keyphrase extraction (Wang, Liu, & McDonald,

[*] Corresponding author.
E-mail addresses: epapagia@csd.auth.gr (E. Papagiannopoulou), greg@csd.auth.gr (G. Tsoumakas).

https://doi.org/10.1016/j.ipm.2018.06.004
Received 19 November 2017; Received in revised form 10 April 2018; Accepted 8 June 2018

2014), we do not use pretrained vectors, but instead learn *local* GloVe representations in the context of *single documents*, in particular full-texts of academic publications. Our main hypothesis is that such local representations will be able to more accurately capture the semantic similarity of the different words and phrases in the context of each document, and help us extract more representative keyphrases, compared to global representations and other state-of-the-art unsupervised keyphrase extraction methods. Our research objective is to investigate whether this hypothesis holds.

Our approach extracts keyphrases from the title and abstract of an academic publication, which constitute a clear and concise summary of the whole publication, in order to avoid the noise and redundancy found in the full-text. Once local word vectors have been learned from the full-text of a given academic publication, we compute the mean vector of the words in its title and abstract, dubbed *reference vector*, which we can intuitively consider as a vector representation of the semantics of the whole publication. We then extract candidate keyphrases from the title and abstract, and rank them in terms of their cosine similarity with the reference vector, assuming that the closer to the reference vector is a word vector, the more representative is the corresponding word for the publication.

The rest of the paper is organized as follows. Section 2 gives a review of the related work in the field of keyphrase extraction as well as a brief overview of methods that produce word embeddings. Section 3 presents the proposed approach. Section 4 describes empirical results highlighting different aspects of our approach and comparing it with other state-of-the-art unsupervised keyphrase extraction methods. Finally, Section 5 presents the conclusions of this work and points to future work directions.

2. Related work

2.1. Automatic keyphrase extraction

Automatic keyphrase extraction is a well-studied task and a variety of techniques have been proposed in the past. In this section, we present both supervised and unsupervised methods in a comprehensive and structured way.

2.1.1. Unsupervised approaches

Unsupervised keyphrase extraction approaches typically follow a standard three-stage process (Hasan & Ng, 2010; 2014). The first stage concerns choosing the candidate lexical units with respect to some heuristics, such as the exclusion of stop words or the selection of words that are nouns or adjectives. The second stage concerns ranking these lexical units by measuring their *importance* through co-occurrence statistics or syntactic rules. The final stage concerns keyphrase formation, where the top-ranked lexical units are used either as keywords or as components of keyphrases.

The baseline approach for unsupervised keyphrase extraction is *TfIdf* (Jones, 1972). It ranks phrases in a particular document according to their frequency in this document (tf), multiplied by the inverse of their frequency in all documents of a collection (idf). Recently, Florescu and Caragea (2017a) proposed an approach for combining TfIdf with any other word-scoring approach. In their approach, a phrase's score is computed by multiplying its frequency within the document (tf) with the mean of the scores of the phrase's words.

Graph-based ranking algorithms are based on the following idea: first, a graph from a document is created that has as nodes the candidate keyphrases, and then edges are added between *related* candidate keyphrases. The final goal is the ranking of the nodes using a graph-based ranking method, such as PageRank (Brin & Page, 1998), Positional Function (Herings, Laan, & Talman, 2005), and HITS (Kleinberg, 1999). *TextRank* (Mihalcea & Tarau, 2004) builds an undirected and unweighted graph with candidate lexical units as nodes for a specific text and adds connections (edges) between those nodes that co-occur within a window of N words. The ranking algorithm runs iteratively until it converges. Once the algorithm converges, nodes are sorted by decreasing order and the top T nodes form the final keyphrases. Variations of TextRank include *SingleRank* (Wan & Xiao, 2008), where edges have a weight equal to the number of co-occurrences of their corresponding nodes within a window, and *ExpandRank* (Wan & Xiao, 2008), where the graph includes as nodes not only the lexical units of a specific document but also the lexical units of the k nearest neighboring documents of the initial document. In ExpandRank, an edge between two nodes exists if the corresponding words co-occur within a window of W words in the whole document set. Once the graph is constructed, ExpandRank's procedure is identical to SingleRank. Recently, another unsupervised graph-based model, called PositionRank, was proposed by Florescu and Caragea (2017b). This method tries to capture frequent phrases taking into account, at the same time, their corresponding position in the text. More specifically, it incorporates all word's positions into a biased PageRank. Finally, the keyphrases are scored and ranked. Wang et al. (2014) propose a graph-based ranking model that takes into consideration information coming from distributed word representations. In particular, again a graph of words is initially created with edges that represent the co-existence between the words within a window of W consecutive words. Then, a weight (the *word attraction score*) is assigned to every edge, which is the product of two individual scores: (a) the *attraction force* between two words which uses the frequencies of the words as well as the distance between the corresponding word embeddings, and (b) the *dice coefficient* (Dice, 1945; Stubbs, 2003). Once more, a weighted PageRank algorithm is utilized to rank the words. A similar approach that uses a personalized weighted PageRank model with pre-trained word embeddings, but with different edge weights is proposed in Wang, Liu, and McDonald (2015).

RAKE (Rose, Engel, Cramer, & Cowley, 2010) is a domain-independent and language-independent method for extracting keyphrases from individual documents. Given a list of stop words, a set of phrase delimiters, and a set of word delimiters, RAKE cuts the document text up to candidate sequences of content words and then builds a graph of word co-occurrences. Afterwards, word scores are calculated for each candidate keyword. The basic difference in comparison with the previous approaches is that RAKE is able to identify keyphrases that contain *interior* stop words. Specifically, RAKE detects pairs of keywords that adjoin one another at least