Natural Language Engineering (2020), **26**, pp. 293–318
doi:10.1017/S1351324919000342

CAMBRIDGE UNIVERSITY PRESS

ARTICLE

Learning keyphrases from corpora and knowledge models

R. Silveira[1,*], V. Furtado[2] and V. Pinheiro[2]

[1]Eixo de Informação e Comunicação, Instituto Federal de Educação do Ceará (IFCE), Brazil and
[2]Programa de Pós-graduação em Informática Aplicada, Universidade de Fortaleza (UNIFOR), Brazil.
*Corresponding author. Email: raquel.vsilveira@hotmail.com

(Received 8 August 2018; revised 19 March 2019; accepted 22 March 2019; first published online 10 September 2019)

Abstract

Extraction keyphrase systems traditionally use classification algorithms and do not consider the fact that part of the keyphrases may not be found in the text, reducing the accuracy of such algorithms a priori. In this work, we propose to improve the accuracy of these systems with inferential mechanisms that use a knowledge representation model, including symbolic models of knowledge bases and distributional semantics, to expand the set of keyphrase candidates to be submitted to the classification algorithm with terms that are not in the text (not-in-text terms). The basic assumption we have is that not-in-text terms have a semantic relationship with terms that are in the text. To represent this relationship, we have defined two new features to be represented as input to the classification algorithms. The first feature refers to the power of discrimination of the inferred not-in-text terms. The intuition behind this is that good candidates for a keyphrase are those that are deduced from various textual terms in a specific document and that are not often deduced in other documents. The other feature represents the descriptive strength of a not-in-text candidate. We argue that not-in-text keyphrases must have a strong semantic relationship with the text and that the power of this semantic relationship can be measured in a similar way as popular metrics like TFxIDF. The method proposed in this work was compared with state-of-the-art systems using five corpora and the results show that it has significantly improved automatic keyphrase extraction, dealing with the limitation of extracting keyphrases absent from the text.

Keywords: Keyphrases extraction; Keyphrases absent from the text; Inference of keyphrases; Knowledge models

1. Introduction

Keyphrases are single or multi-word linguistic units that represent the salient aspects of a document (Debanjan *et al.* 2018a). Keyphrases are used in scientific and literature works to enable the cataloging, organization, and searching of texts (Cândido Júnior *et al.* 2015). Keyphrase extraction is a fundamental task in natural language processing (NLP) that facilitates the mapping of documents with a set of representative phrases (Debanjan *et al.* 2018a).

It is assumed that when writing a text, writers assign keyphrases that represent the meaning of the text, so that the reader can have a quick sense of what the text refers to. With the advent of the Internet and the proliferation of digital documents, keyphrases have become essential for information retrieval (Arampatzis *et al.* 1998; Gutwin *et al.* 1998; Turney 2000; Nguyen and Kan 2007), opinion mining (Berend 2011), and NLP tasks, such as categorization, summarization, and clustering of documents (Hulth and Megyesi 2006; Han, Kim, and Choi 2007; Litvak and Last 2008). The relevance of this activity has attracted the attention of researchers who have used classification to automatically select, among the terms of a text, those that are more appropriate to become keyphrases (Frank *et al.* 1999; Witten *et al.* 1999; Nguyen and Kan 2007; Medelyan 2009).

© Cambridge University Press 2019

https://doi.org/10.1017/S1351324919000342 Published online by Cambridge University Press

Manually assigning keyphrases can be time-consuming, tedious, and expensive. Thus, automatic extraction of keyphrases is necessary (Liu *et al.* 2010). The scientific literature has been extensive in proposing methods for automatic extraction of keyphrases (Frank *et al.* 1999; Witten *et al.* 1999; Medelyan, Frank, and Witten 2009; Meng *et al.* 2017), almost always using inductive learning. These works limit the candidate sets to keyphrases, those of which are present in the text or in the training corpus of the classification algorithm, reducing the accuracy of any classifier a priori. An improvement of this is the work of Meng *et al.* (2017), which infers keyphrases that are not necessarily in the text from terms present in other documents of the training corpus. One might assume that there are inferences made by authors that lead them to choose keyphrases that are not in the text. This intuition was strengthened by the fact that when checking corpora used in previous research, we calculated the percentage of missing keyphrases in the document text, as well as the percentage of absent keyphrases in the texts of all the documents in the corpus, it was found that they all contained keyphrases that were not in all the corpus used in the training process. Kim *et al.* (2010) used a corpus of news with keyphrases assigned by humans, where approximately 19% of the keyphrases are not in the text and 11% are not present in the complete corpus. Extending the analysis to the corpus of CiteULike (Medelyan 2009), which contains scientific papers, 21% of the keyphrases attributed by the authors are also not found in the text of the works and 14% are not found in the corpus. In other words, the maximum coverage that traditional keyphrase extraction methods can achieve in these corpora is 81% and 79%, respectively, considering as candidates only the terms present in the text. This fact has been repeatedly found in other corpora in which keyphrases are determined by people (Hulth 2003; Nguyen and Kan 2007; Medelyan 2009).

It is also worth noting that in the keyphrase extraction task, the amount of terms assigned as keyphrases in a document by the authors is significantly less than the number of terms that are not keyphrases in the corresponding document, making unbalancing the training set and making it difficult for machine learning algorithms to learn the examples related to the minority class (terms that are keyphrases). In Medelyan (2009), only 0.6% of the terms present in the text are attributed as keyphrases by the authors, while the other terms are not keyphrases. This fact is repeated in other corpora (Hulth 2003; Kim *et al.* 2010; Nguyen and Kan 2007; Krapivin, Autayeu, and Marchese 2008).

In this work, we describe our research that concentrates on defining a method that aims at increasing the accuracy of the traditional systems of keyphrase extraction, expanding the training set with not-in-text terms obtained from an inference process using knowledge models, and overcoming the limitation of identifying keyphrases that do not appear in the text and/or in the corpus. Furthermore, because of the number of positive terms (terms that are keyphrases) being significantly smaller than negative terms (terms that are not keyphrases), the proposed method also involves the application of existing strategies of data balancing.

Our basic hypothesis is that, in a similar way to people, it is necessary to learn inference mechanisms that allow for the expansion of the candidate set to keyphrases, which are not in the text (not-in-text terms, for short), dealing with the limitation of obtaining candidates presented in the text and in the corpus. We also hypothesize that when people assign keyphrases that are not present in the text, they capture some form of semantic relation (synonym, generalization, etc.) between not-in-text terms with the subject of the text and consequently there should be an association between the term that is not in the text and the other terms that are in the text.

In order to capture this semantic relationship, we have defined features to represent the relevance of the not-in-text terms, in a way that the candidate set, which is the input of the classification algorithm, is sufficiently informative to improve the accuracy of the task for keyphrase extraction, allowing the algorithm to learn the not-in-text keyphrase. For our approach to function, it is necessary to choose a knowledge representation model where the inferences of not-in-text terms should be applied. Two traditional options are used: distributional semantics, such as latent semantic analysis (LSA) (Deerwester *et al.* 1990), and symbolic models of knowledge bases,

https://doi.org/10.1017/S1351324919000342 Published online by Cambridge University Press

Table 1. Heuristics to capture the relevance of terms

Heuristic	Description
TFxIDF (Salton and McGill 1983)	Value proportional to the frequency of the term in the document and inversely proportional to the occurrence of the term in the corpus.
First occurrence (Frank *et al.* 1999)	Distance from the first occurrence of the term from the beginning of the document.
Spread (Medelyan *et al.* 2009)	Distance between the first and last occurrence of a term in the document, normalized by the size of the document.
Phrase length (Medelyan and Witten 2008)	Size of the words of a term.
Keyphraseness (Frank *et al.* 1999)	Quantifies the frequency of a term as a keyphrase in the training set.
Wikipedias keyphraseness (Mihalcea and Andra 2007)	Probability that a term is a link in Wikipedia.
Inverse Wikipedia Linkage (IWL) (Medelyan *et al.* 2009)	Quantifies the number of other articles in Wikipedia that link to the most likely Wikipedia article for the term.
Node degree (Medelyan and Witten 2008)	The number of hyperlinks that link the most likely Wikipedia article of the term to other Wikipedia pages which represent another candidate term.
Semantic relatedness (Medelyan *et al.* 2009)	average number of hyperlinks in Wikipedia in common with other candidate terms.
Inference Semantic Relatedness (ISR) (Silveira, Furtado, and Pinheiro 2015)	Captures if the semantic relationship of a not-in-text term (in relation to the other terms of the text) is greater than the semantic relationship of the term that inferred it.

such as InferenceNet (Pinheiro *et al.* 2010), ConceptNet (Liu and Singh 2004), and WordNet (Miller 1995).

More specifically, the main contribution of this paper is the elaboration of a method for keyphrase extraction that is able to assign keyphrases absent from the text and from the training corpus and consequently outperform (considering the f-measure metric) the state-of-the-art keyphrase systems. More specifically, the proposed method (i) defines an inference strategy to improve the set of keyphrase candidates by also including candidate terms that are not present in the text and (ii) elaborates new heuristics to characterize the relevance of not-in-text terms to be keyphrases. Our method was validated by comparing the f-measure of the proposed method with the state-of-the-art systems using five traditional benchmark corpora.

2. Background knowledge

2.1 Heuristics to capture keyphrase relevancy

Medelyan (2005) points out that the main challenge of keyphrase extraction systems that use classification algorithms is to define informational heuristics that can be determined for each term and used to distinguish terms that are keyphrases and that are not keyphrases between the set of candidate terms. Table 1 shows a summary of the heuristics used in automatic keyphrase extraction systems to capture the relevance of a keyphrase term.

2.2 Systems for automatically extracting keyphrases

The literature in NLP is rich in proposals of systems that, based on corpora with texts having keyphrases labeled by people, try automatically extracting keyphrases. The systems for automatically extracting keyphrases from text are predominantly divided into two categories: supervised

https://doi.org/10.1017/S1351324919000342 Published online by Cambridge University Press

and unsupervised. Supervised methods uncover the mapping between features about the syntax, structure, and semantics of the text and certain keyphrases that best summarize the meaning of the text, whereas the unsupervised methods are mostly based on clustering and graph-based ranking.

A wide variety of keyphrase extraction methods have been presented in the literature. Some methods are based on supervised learning (Frank *et al.* 1999), unsupervised learning (Mihalcea and Tarau 2004; Bougouin, Boudin, and Beatrice 2013), and semi-supervised learning (Ammar *et al.* 2017; Ye and Wang 2018). In addition, word representation has been used as word embeddings which has also been used in keyphrase extraction (Wang, Liu, and McDonald 2015; Debanjan *et al.* 2018ab). The main challenge of keyphrase extraction is that sometimes keyphrases are not present in the text, so some methods will fail to extract those keyphrases. The literature of keyphrase extraction has demonstrated some methods that extract keyphrases missing from the text (Silveira, Furtado, and Pinheiro 2015; Meng *et al.* 2017; Wang, Liu, and McDonald 2015).

Keyphrase extraction algorithm (KEA) (Frank *et al.* 1999; Witten *et al.* 1999) is one of the first systems that extracts keyphrases from documents. It applied a Naive Bayes (Domingos and Pazzani 1997) classifier for keyphrase extraction as a classification task in which the textual terms are represented by three features: (i) TFxIDF, (ii) position of the first occurrence, and (iii) keyphraseness. Medelyan and Witten (2006) improved the original KEA and proposed the KEA++, a method based on machine learning and which works in two main stages: (i) candidate identification, which identifies thesaurus terms that relate to the documents content and (ii) filtering, which uses a learned model to identify the most significant terms based on certain properties or features.

Wikify! Mihalcea and Andra (2007) uses the property of Wikipedia's keyphraseness, which quantifies the probability of the term appearing as a Wikipedia tag, and was inspired in reproducing the publishers' editorial style in Wikipedia and select keyphrases.

The multi-purpose automatic topic indexing (MAUI) system (Medelyan, Frank, and Witten 2009) enhances KEA's framework by proposing features that capture a semantic knowledge from Wikipedia. MAUI divides the process of keyphrase extraction into two steps: (i) the selection of the candidates to be labeled a keyphrase and (ii) the application of the learning algorithm. At the candidate selection step, all *n*-grams up to a maximum length of three words that do not begin or end with a stop word and that appear more than once in the text are selected as candidate terms. Moreover, MAUI has extended the set of features that characterize the text by creating six new ones: (i) spread, (ii) semantic relatedness, (iii) Inverse Wikipedia Linkage, (iv) phrase length, (v) Wikipedias keyphraseness, and (vi) node degree. MAUI uses bagged decision trees.

Recently, Zhang *et al.* (2016) tackle the problem of keyphrase extraction in a different context using an innovative approach. The context is that of extracting keyphrases from tweets. The approach is based on deep recurrent neural network (RNN) in order to rank the keyphrase candidates. Instead of looking for a pattern from features of the text, the approach identifies the potential keyphrases exploring the sequence of words in the text. Zhang *et al.* (2018) present a neural keyphrase extraction framework that exploits conversation context. They proposed four types of neural encoders (averaged embedding, RNN, attention, and memory networks) for capturing salient content to help in indicating keyphrases in target posts.

On many occasions in the keyphrase extraction task, candidate terms that are not assigned as a keyphrase by the authors are labeled as non-keyphrases and used as negative examples in training by supervised classifiers. Often these negative examples outnumber the positive examples, creating an unbalanced training set. The literature presents data balancing techniques (Chawla *et al.* 2002) in order to reduce the negative effects that unbalanced data can cause in training. Sterckx *et al.* (2016) address this problem by creating a large test collection of articles with many different opinions per article, evaluating the effect on extraction performance, and presenting a procedure for supervised keyphrase extraction with noisy labels.

The literature review also contemplates approaches that use unsupervised learning for keyphrase extraction. Graph-based ranking methods follow an unsupervised approach for

https://doi.org/10.1017/S1351324919000342 Published online by Cambridge University Press

keyphrase extraction. Mihalcea and Tarau (2004) proposed a graph-based ranking model called TextRank, which builds a graph that represents the interconnection between the words or other text entities with meaningful relations. The principle of the model is the fact that when one vertex links to another one, it is casting a vote for the other vertex. The higher the number of votes that are cast for a vertex, the higher the importance of the vertex, and this information is also taken into account by the ranking model.

Grineva, Grinev, and Lizorkin (2009) used a model to build a graph and analyze the interconnectivity of the vertices. However, instead of using statistics gathered from a training set, they use semantic information derived from Wikipedia. Bougouin *et al.* (2013) improved TextRank and presented TopicRank, a keyphrase extraction method based on graphs that classify candidate sets of keyphrases in topics and use them as vertices in a complete graph. A ranking model based on a graph is applied to allocate the significance score for each topic. Keyphrases are generated selecting a candidate for each topic with the largest score value. Bougouin, Boudin, and Daille (2016) propose TopicCoRank, a keyphrase annotation method built on the existing TopicRank method (Bougouin, Boudin, and Beatrice 2013). The authors used two graph representations, one for the document and one for the specialized domain. Then, they applied a co-ranking algorithm to perform both keyphrase extraction and assignment in a mutually reinforcing manner.

The literature review also has semi-supervised approaches for keyphrase extraction. Danesh, Sumner, and Martin (2015) developed the SGRank using an unsupervised keyphrase extraction approach by combining statistical heuristics with a graph-based algorithm in which candidate *n*-grams are ranked based on a modified version of TFxIDF as used in KP-Miner (El-Beltagy and Rafea 2010). Wang and Li (2017) constructed a system that uses external knowledge to enrich heuristics that represent a candidate for the keyphrase, including Wikipedia, IEEE taxonomy, and pre-trained word embeddings, among others. The system involves three steps: (i) generation of candidates, (ii) ranking of keyphrases, and (iii) classification of the type of keyphrase. To assign candidates as keyphrases, two unsupervised algorithms TextRank (Mihalcea and Tarau 2004) and SGRank (Danesh, Sumner, and Martin 2015) are incorporated into a Random Forest, providing their output as complementary heuristics. Ammar *et al.* (2017) developed a system using semi-supervised learning based on the extraction of end-to-end relationships of Miwa and Bansal (2016) for keyphrase extraction and identification of types and semantic relationships between keyphrases of the same type.

Ye and Wang (2018) proposed semi-supervised methods for extracting keyphrases using labeled and unlabeled data for learning. Two strategies are proposed: (i) unlabeled documents are first tagged with synthetic keyphrases obtained from unsupervised keyphrase extraction methods or a self-learning algorithm, and then combined with labeled samples for training and (ii) a multi-task learning framework is investigated to jointly learn to generate keyphrases as well as the titles of the articles.

All of the works presented here consider the universe of terms that appear in the text. We identified that there are keyphrases in the corpus that are not textual, that is, they are not articulated in the texts. Therefore, keyphrase extraction methods that do not consider not-in-text keyphrases are not comparable with those assigned by the best performing human taggers.

Recently, Silveira *et al.* (2015) proposed a feature called inference semantic relatedness (ISR) for terms that do not appear in the text, with the objective of capturing if the semantic relatedness of a not-in-text term (in relation to the other terms in the text) is higher than the semantic relatedness of the textual term from which it was inferred. Meng *et al.* (2017) proposed to apply a generative model based in RNN for keyphrase prediction. This model can generate keyphrases based on an understanding of the text, regardless of the presence or absence of keyphrases in the text. The basic idea of the proposed keyphrase generation model is to use an RNN Encoder–Decoder model to convert the content of the original text with an RNN encoder and generate corresponding keyphrases by producing the probability of all words in the vocabulary with an RNN decoder. Experiments with five benchmark datasets were performed, and the results show

https://doi.org/10.1017/S1351324919000342 Published online by Cambridge University Press

that this proposed model significantly outperforms existing supervised and unsupervised extraction methods. Chen *et al.* (2018) proposed a new sequence-to-sequence architecture for keyphrase generation, CorrRNN, which captures correlation among multiple keyphrases. A coverage mechanism (Tu *et al.* 2016) is employed to indicate whether the word has been summarized by previous phrases. Furthermore, preceding phrases are taken into account to avoid the repetition of phrases and to improve result coherence.

Word embeddings have also been used in keyphrase extraction. Wang *et al.* (2015) proposed an approach that uses word embeddings as an external knowledge base for both keyphrase extraction and generation. They proposed an unsupervised graph ranking approach using word embeddings as an external knowledge base for extracting keyphrases that appear in a document and then introduced a keyphrase generation approach built on top of the extraction model using the power of word embeddings to suggest abstractive keyphrases not appearing in a document. Debanjan *et al.* (2018a) and Debanjan *et al.* (2018b) proposed a method that represents candidate keyphrases extracted from a scientific article by domain-specific phrase embeddings and ranking them using a theme-weighted PageRank algorithm (Langville and Meyer 2004), such that the thematic weight of a candidate keyphrase indicates how similar it is to the thematic representation or the main theme of the article, which is also constructed using the same embeddings. Papagiannopoulou and Tsoumakas (2018) presented an unsupervised method for keyphrase extraction that uses local word embeddings (more specifically, GloVe vectors (Pennington, Socher, and Manning 2014)) to detect the candidate keyphrases and to provide a complete set of representative keyphrases for a particular title and abstract.

3. Learning not-in-text keyphrases

When writing texts, people often assign as keyphrases terms that are not necessarily present in the body of the text, but which represent the meaning of the text in some way. In this way, automatic keyphrase extraction systems have the challenge of extracting present and missing keyphrases from the text. Some important aspects of the extraction of missing keyphrases in the text and/or corpus should be considered: (i) identification of candidate terms that are not in the text and (ii) definition of heuristics that allow the algorithm to create a template for deciding which not-in-text candidate terms are keyphrases. This section defines the proposed method for extracting keyphrases, dealing with the limitation of traditional systems of extracting missing keyphrases from the text and/or the corpus. In addition, the inference process of the not-in-text candidate terms and the new heuristics to characterize the relevance of these terms is described.

3.1 Architecture of inference method

The task of identifying keyphrases can be modeled as the problem of selecting from a list of candidate terms $C = \{c_1, c_2, ..., c_n\}$ the subset $K = \{k_1, k_2, ..., k_z\}$ representing the terms that summarize the contents of the document $d \in D$, where D is a collection of documents. We treat keyphrase extraction as a task of binary classification. Each candidate term $c \in C$ of the training set is marked as a positive or negative example, depending on whether the authors associated the term as a keyphrase or not in the corresponding document. The classification algorithm learns the model based on the training set, analyzing the values of the heuristics of each candidate example.

The authors assign keyphrases to a document based on their semantic meaning, instead of following only the written content of the text. In this work, an inference method[a] is proposed to generate part of the set of candidates, more specifically, a complementary form to add to the set of candidates terms that are absent from the text. Figure 1 illustrates the steps that our method follows.

[a]Source code is available at https://github.com/raquelsilveira/KeyphraseExtraction.git

https://doi.org/10.1017/S1351324919000342 Published online by Cambridge University Press

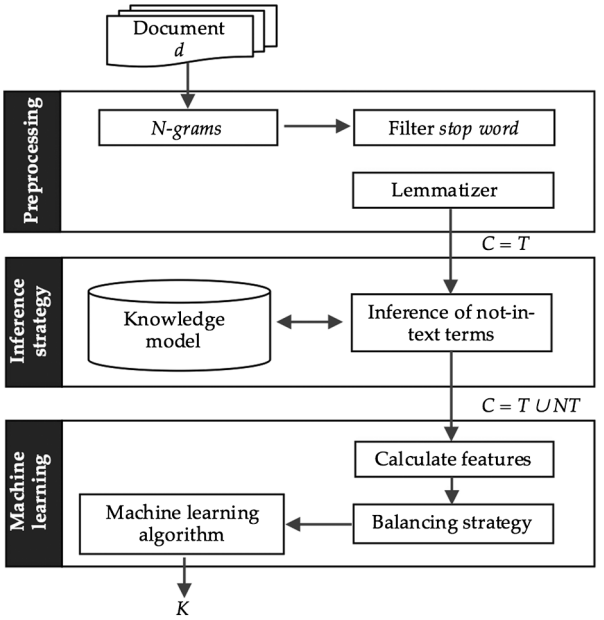


Figure 1. Inference of not-in-text candidates from the knowledge model for learning keyphrases of a document *d*.

3.1.1 Preprocessing

Initially, the text of each document d in D is preprocessed. The text words are tokenized[b] and all the possible n-grams with a maximum of n ($n = 3$) words and that appear in the text more than x ($x = 2$) times form the set of candidate terms. A stop word filter is applied to remove from the candidate set all the n-grams which have initiated or terminated with a stop word.[c] The n-grams of the candidates' set are lemmatized[d] and selected as text candidates, forming the set of textual candidate terms, $T = \{t_1, t_2, ..., t_u\}$. Therefore, at first the set of candidates C is formed by terms present in the text of document d, that is, $C = T$.

3.1.2 Inference strategy

We propose to add to the set of candidates C, terms that do not appear in the text and/or in the corpus, obtained from an inference strategy that infers terms (which do not appear in the text) according to the knowledge model that has a semantic relationship with the terms of the text.

It is assumed that when people assign keyphrases that are not present in the text, they somehow express the existence of a semantic relationship between those keyphrases and the meaning of the text, either because it represents the main subject of the document, a generalization of specific terms present in the text or a synonym for terms that represent the content of the document. Thus, it is argued that there must be an association between the missing keyphrases of the text and the terms present in the text. In this sense, the inference process attempts to obtain not-in-text terms, which are related to the terms in the text.

The inference strategy generates a set $NT = \{nt_1, nt_2, ..., nt_j\}$ of not-in-text candidate terms ($NT \not\subset T$), or in other words, candidate terms that are not in the text. For each textual term from

[b]The tokenization process separates text into minimal units (tokens) using characters to delimit units, such as spaces or punctuation markers. For example, given the text *Extraction of keyphrases*, the following tokens are separated: [*extraction*, *of*, *keyphrases*].

[c]Generally, elements like conjunctions, prepositions, pronouns, and articles are considered as stop words.

[d]The process of lemmatization reduces the words inflected or derived from the lemma. For example, the words *boy* and *boys* form the same lemma, *boy*.

https://doi.org/10.1017/S1351324919000342 Published online by Cambridge University Press

Barcos is one of the most important players of **Palmeiras** and his departure from the team would be something very harmful. Therefore, the top scorer of the team in the **Brazilian Championship** with 11 goals and the season, with 25, admitted he has taken care not to take a new yellow card and thus be suspended. "I'm taking care for several games not to take more cards. I will keep calm and avoid any trouble and complaint," said the striker, who won the last card on 22 September, in the victory by 3 to 1 over Figueirense.

Keyphrases: Football, Brazilian Championship, Palmeiras, Barcos.

Figure 2. Sports Domain News from corpus News corpus Silveira *et al.* 2015.

$ti \in T$, one searches the semantic relations between two terms in a knowledge model by means of a triplet $\{t_i, \phi, nt_j\}$, where t_i is a textual term, nt_j is a term absent of the text, and ϕ expresses a semantic relation. We assume that it is possible to find for a textual term t_i in d a semantic relatedness with one or more not-in-text terms nt_j, thus Tnt_j represents the set of all the textual terms that have a semantic relation with nt_j. If $nt_j \notin T$ and $|Tnt_j| \geq z$, where $|Tnt_j|$ represents the number of terms of the text that inferred the term not-in-text nt_j and z the minimum amount of inference by terms of the text, then nt_j is a not-in-text candidate term and is included in the set NT. The complete candidate set $C = T \cup NT$ contains the terms that appear in the document texts and those not-in-text terms ($nt_j \notin T$) that do not appear in the text and/or corpus, but that possess a semantic relatedness with d.

For example, Figure 2 displays a sports text with the corresponding keyphrases. The set of terms in the text T is formed by the terms in the text, such as terms referring to names of clubs and players (Barcos, Palmeiras, and Brazilian Championship). In this example, the keyphrase that is not explicitly expressed in the text refers to the type of sport (in this case, football). A semantic relationship between the name of the clubs and/or players can retrieve, from a knowledge model, this not-in-text term as a candidate for the keyphrase, composing the set of not-in-text NT candidate terms.

3.1.3 Machine learning

However, although there is a semantic relationship with the terms of the text, it does not mean that all inferred not-in-text terms are good candidates for the keyphrase. In this example, football proves to be a good candidate for the keyphrase because it represents a generalization of the type of sport; however, referee is probably not a good candidate for the keyphrase because it is not semantically related to many terms of the text, this does not represent the content of the text. Therefore, with the candidate set, $C = T \cup NT$, it is possible to compute several features for each candidate c_n, which are used in a machine learning algorithm to obtain the probability that the candidate is indeed a keyphrase in K.

Allied with this, the task of extracting keyphrases has notably an unbalanced data problem, since the number of terms considered keyphrases is significantly lower than the terms that are not keyphrases. Therefore, the improvement of the training from inferences made in knowledge bases makes this problem even more evident. To address this problem, our method also involves the application of class balancing strategies prior to the application of the machine learning algorithm. In the following sections, we detail (i) the process of inference of not-in-text candidate terms and (ii) heuristic methods to select the candidate not-in-text terms.

3.2 Inference of not-in-text candidate terms

As defined above, the set of candidate keyphrases C contains terms that appear in a document d, and not-in-text terms that have a semantic relationship in a knowledge model with terms that are in d. Then two traditional strategies can be used in the inference of not-in-text candidates terms: (1) distributional semantics and (2) symbolic models of knowledge bases.

https://doi.org/10.1017/S1351324919000342 Published online by Cambridge University Press

The knowledge model of distributional semantics when executed in the corpus itself allows us to capture more specific terms in the corpus domain. A term that does not appear in the text of a particular document but which is present in the text of other documents of the same corpus of d_i and which has a semantic relation to the terms of the text of d_i possibly represents a specific term in the domain of the corpus. On the other hand, symbolic models of the knowledge base allow us to capture more general terms from external resources (knowledge bases) and, not necessarily, terms present in the other documents of the corpus. The inference process of not-in-text terms can be accomplished by combining knowledge models to increase the number of not-in-text candidate terms that are keyphrases, while not restricting the inference only to the terms present in the corpus.

3.2.1 Inference of not-in-text terms from distributional semantic models

Distributional semantics models, such as LSA (Deerwester *et al.* 1990), can be used to describe the distributional vector of words from a large corpus according to their occurrence patterns, obtaining the similarity between words using these vectors.

Formally, M is an $n \times m$ matrix of document term from a collection of documents D. Each column of M corresponds to a document ($d \in D$), and each line corresponds to a text term ($t_i \in T$). The dimensions of M, n, and m correspond to the number of terms in the text $|T|$ and documents $|D|$, respectively. Therefore, each candidate term of the text $t_i \in T$ and each document $d \in D$ correspond to a row and a column, respectively, of the matrix M. Each cell of matrix M corresponds to the number of times that the term t_i occurs in each document $d \in D$. If the term t_i occurs x times in document d, then $M\,[t_i, d] = x$. The occurrence of terms can be modified so that the terms are weighted according to the specificity for the documents, that is, rare terms are given more weight than common terms. The value of the TFxIDF property is used as the value of the occurrence of the term t_i in document d. Then, the matrix M is subjected to a singular value decomposition (SVD) process, generating M_k, in order to find a reduced dimensional representation of M that emphasizes the strongest relationships and removes the existing noises.

In this approach, a sorted list $TS = \{t_1, t_2, ..., t_n\}$ of top n similar words, where n is the quantity of terms in M, is computed for each textual term from $t_i \in T$, according to the cosine similarity of the distributional vector of words of the matrix M_k, creating a set P with n triplet, in the form, $\{t_i, \phi, nt_n\}$, that is, $P = \{\{t_i, \phi, t_1\}, \{t_i, \phi, t_2\}, ..., \{t_i, \phi, t_n\}\}$. The cosine similarity between the two-term vectors (t_i and t_j) is calculated as follows:

$$similarityCos(t_i, t_j) = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2}\sqrt{\sum_{i=1}^{n} B_i^2}}, \tag{1}$$

where A and B correspond to the distributional vector in M_k of the terms t_i and t_j, respectively. Finally, given a term of the text t_i, for each triple $\{t_i, \phi, t_n\} \in P$, where t_n represents the nth most similar term to t_i, if $t_n \notin T$ and $|NT_{ti}| < k$, where NT_{ti} represents the set of not-in-text terms inferred from t_i and k the amount of not-in-text terms for each term of the text, then t_n is a not-in-text term and is included in the set NT.

3.2.2 Inference of not-in-text terms from symbolic models of knowledge bases

Traditionally, symbolic knowledge bases express the semantic relations between two terms by means of a triplet $\{t_i, \overbrace{\phi}^{n}, nt_j\}$, where t_i and nt_j are terms in a natural language and $\overbrace{\phi}^{n}$ represents the path of size n that links the two terms by means of relationships between them, for example, *football*, *isA*, *sport*. The relationships can be of several types, such as capableOf, propertyOf, effectOf, isA, and partOf.

https://doi.org/10.1017/S1351324919000342 Published online by Cambridge University Press

We assume the existence of a knowledge base κ as input and for each term of the text of $t_i \in T$, we search the knowledge base κ by triples of size $n = 1$, with the form $\{t_i, \overbrace{\phi}^{1}, nt_j\}$ or $\{nt_j, \overbrace{\phi}^{1}, t_i\}$. If $nt_j \notin T$, then nt_n is a not-in-text candidate term and is included in the NT set. Formally, this can be written logically through the following rules of inference:

$$\frac{(nt_j, \overbrace{\phi}^{1}, t_i), (t_i \in T), (nt_j \notin T)}{nt_j \in NT} \tag{2}$$

$$\frac{(t_i, \overbrace{\phi}^{1}, nt_j), (t_i \in T), (nt_j \notin T)}{nt_j \in NT} \tag{3}$$

For example, given a term of the text *sport*, it is possible to deduce from a knowledge base the following triples with *n = 1: {football, IsA, sport}, {sport, isA, game}*, forming *NT = {football, game}*. The use of $n > 1$ searches for matches related to *football* and *game*, obtain, for example, *{football, RelatedTo, ball}*, *{attacker, HasContext, soccer}*, *{game, IsAn, activity},* and *{domino, IsA, game}*, obtaining *NT = {football, sport, ball, striker, activity, domino}*. The intuition is that not-in-text terms that are semantically related to text terms through a direct relationship ($n = 1$) are potential candidates for keyphrases.

3.3 Heuristic methods for selecting not-in-text candidates

The deduction process obtains all the existing terms that have a similarity relationship with the textual terms. For example, in a text about sport, it is possible to deduce the soccer team names and the players as well as other terms used in soccer, such as goal, referee, and stadium. However, not all the deduced not-in-text terms are considered good candidates for keyphrases. Therefore, we need to measure the relevance of the not-in-text terms, allowing the algorithms to learn and characterize a significant example of not-in-text keyphrases. Thus, we use the feature proposed by Silveira *et al.* (2015) and we define other two new heuristics that measure the relevance of a not-in-text term to be considered a good candidate of a keyphrase.

3.3.1 Inference semantic relatedness

The first heuristic considers that the semantic relationship of a term is a good indication of its relevance to be a keyphrase. To characterize the quality of not-in-text terms, Silveira *et al.* (2015) have created a Boolean heuristic called ISR, which captures whether the semantic relatedness of the not-in-text term nt_j (in respect to other terms of the text) is higher than the semantic relatedness of the term t_i which inferred it. It is argued that terms that have a greater semantic relationship with the terms of the document are more likely to represent the text. More specifically, not-in-text terms that are semantically more related to the text are potential candidates for keyphrases.

Formally, given $\{nt_i, \phi, t_i\}, \{nt_i, \phi, t_j\}, ..., \{nt_i, \phi, t_n\}$, where $nt_i \in NT$ is a not-in-text term, $T_t = \{t_i, t_j, ..., t_n\} \subset T$ are textual terms with semantic relationship to nt_i and ϕ represents the relationship in a knowledge base between nt_i and each term of T_t. Thus, the $ISR(nt_i)$ to the not-in-text terms can be written as follows:

$$ISR(nt_i) = \begin{cases} 1, & if(SemRelat(nt_i) \geq SemRelat(t_c)) \\ 0, & otherwise \end{cases}, \tag{4}$$

https://doi.org/10.1017/S1351324919000342 Published online by Cambridge University Press

where $nt_i \in NT$ represents the not-in-text term, $t_c \in T_t$ corresponding to the term of higher value of TFxIDF in T_t and $SemRelat(x)$ is the value of semantic relatedness (according to Section 2.1) of the term x, whose value quantifies the semantic similarity of the term x with the other candidates.

For example, considering a crime text that does not mention the type of crime, but contains the terms *revolver* and *death* in the text. These terms (*revolver* and *death*) can infer, through searching for the semantic relationship in a knowledge model, the not-in-text term *murder*. The basic ISR hypothesis is that if *murder* has a bigger semantic relationship measurement ($SemRelat(x)$) with the text than the terms *revolver* and *death*, then it is more relevant as a keyphrase.

Finally, it is important to note that a not-in-text nt_i term can be inferred from n terms of the text, which form the set $T_t = \{t_i, t_j, ..., t_n\}$. In the ISR property, the comparison of the semantic value of the not-in-text nt_i is performed with the text term $t_k \in T_t$ with the highest value of TFxIDF. It is considered that the term t_k that best represents the set T_t is the most significant for the document, that is, the term with the highest value of TFxIDF. For example, the terms *revolver* and *death* infer the term *murder*. Thus, considering that murder = *revolver*, *death* and that $TFxIDF(death) \geq TFxIDF(revolver)$, where $TFxIDF(x)$ is the value of the property TFxIDF of the term x, the term *death* is used in the ISR in the comparison of the semantic similarity with the not-in-text *murder*, in other words $ISR(murder) = 1$, if $SemRelat(murder) \geq SemRelat(death)$ and 0, otherwise.

3.3.2 Semantic power inference

The second heuristic, called semantic power inference (SPI), captures the descriptive strength of a not-in-text candidate nt_i. This strength is based on the intuition that not-in-text keyphrases must have a strong semantic relationship with the text. The descriptive power is based on the intuition that not-in-text keyphrases are deduced from terms that have a high semantic value with the text, that is, not-in-text terms that are semantically related to text terms that have high semantic value with the document are potentially more suitable candidates for keyphrases.

For example in a text about football, terms such as the name of the club and the name of the players have high semantic relationship. Subsequently, all the not-in-text terms inferred from these probably have a strong semantic relationship with the text. Formally, given a not-in-text term nt_i deduced from a document d where a subset of terms that are in the text $T_t = \{t_i, t_j, ..., t_n\}$, $SPI(nt_i)$ is computed as:

$$SPI(nt_i) = \frac{\sum_{t_i \in T_t} SemRelat(t_i)}{\sum_{t_k \in T} SemRelat(t_k)}, \tag{5}$$

where $SemRelat(x)$ is the value of the semantic relatedness feature (according to Section 2.1) of the term x in the document d, whose value quantifies the semantic similarity of the term x with the other candidates, T_t is the set of terms of the text that inferred the not-in-text term nt_i, and T the set of all terms of the text.

3.3.3 Inference frequency versus inverse document frequency

The third feature is based on the intuition that not-in-text terms that are good candidates of a keyphrase are those that are inferred from various terms belonging to a specific document and that, in turn, are not often inferred in other documents. Therefore, based on TFxIDF, we propose a heuristic called inference frequency x inverse document frequency (IFxIDF) for not-in-text terms. This heuristic captures the frequency of a not-in-text term to be inferred from a specific document multiplied by the inverse frequency of its occurrence in the corpus. Given the not-in-text term nt_i inferred in a document d, $IF \times IDF(nt_i)$ is calculated as:

https://doi.org/10.1017/S1351324919000342 Published online by Cambridge University Press

$$IF \times IDF(nt_i) = \frac{freq(nt_i, d)}{\sum_{nt_j \in NT} freq(nt_j, d)} \times -log_2 \frac{d_{nt}}{|D|}, \quad (6)$$

where $freq(nt_i, d)$ is the number of times that nt_i was inferred from any textual term of the document d, NT is a not-in-text terms set of document d, d_{nt} is the quantity of documents of the corpus in which nt_i is inferred, and $|D|$ is the number of documents of the corpus. The first component in this expression refers to the normalized frequency of nt_i in the document d. The second component is the negative logarithm of the inverse frequency in the documents, in which high values indicate rare terms.

It is assumed that not-in-text keyphrases are represented by terms that are significant for a particular document. Therefore, not-in-text terms inferred from multiple documents are not relevant to a particular document and thus are not good candidates for the keyphrase. The rationale behind this is that terms that are infrequent in the corpus and frequent in a document are good candidates of keyphrases because they discriminate the document.

4. Experimental evaluation

The proposed method was applied to verify if the quality of the automatic extraction of keyphrases is better with the set of candidates extended by the process of inference, as well as, in addition to the terms of the text, the terms absent from the text. More specifically, the application of knowledge models such as WordNet (Miller 1995; Fellbaum 1998) and ConceptNet (Liu and Singh 2004), as well as distributional semantics, allows to improve the set of candidates of the learning algorithms. In addition, an experimental evaluation was performed using heuristics, such as ISR, SPI, and IFxIDF, for not-in-text terms, in order to verify if the learning algorithm also learns not-in-text keyphrases.

4.1 Training and test sets

For evaluating the proposed approach, five widely adopted scientific publication datasets were used. The first test scenario was executed in CiteULike (Medelyan 2009), a corpus of scientific articles that contains 180 academic documents from two journals: HighWire and Nature. In total, 332 taggers were associated with 4638 keyphrases. However, we are going to use only 935 keyphrases because they were identified by at least 2 taggers. As stated before, 21% of the keyphrases were not present in the texts of the documents.

The second is a corpus of paper abstracts, Inspec (Hulth 2003). This corpus contains 2000 journal abstracts in English from the areas of Computer and Control, and Information Technology. Each abstract has keyphrases assigned by a professional indexer from a set of controlled terms and from a set of uncontrolled terms that can mean any suitable term. In this corpus, nearly 27% of the keyphrases are absent from the texts.

The third dataset was built by Kim *et al.* (2010) for the keyphrase extraction task of the SemEval-2010. This corpus has 284 scientific articles from conferences and workshops of the ACM Digital Libraries. All collected papers contain author-assigned and reader-assigned keyphrases. In this corpus, on average 15% of the reader-assigned keyphrases and 19% of the author-assigned keyphrases do not appear in the text.

The fourth and fifth datasets contain scientific publication and were built by Nguyen and Kan (2007) (NUS corpus) and Krapivin *et al.* (2008) (Krapivin corpus), respectively. The NUS corpus contains 211 computer science articles, and Krapivin corpus has 2304 documents in the computer science domain.

All analyzed corpora have keyphrases associated by authors or readers that are absent from the text of the documents and the corpus. Table 2 shows the statistics for the number of documents, the percentage of keyphrases that are not present in the text of the referred to document and the percentage of keyphrases that are not included in any document of the corpora.

https://doi.org/10.1017/S1351324919000342 Published online by Cambridge University Press

Table 2. Statistics of documents and not-in-text keyphrases in the corpora evaluated

Corpus	Docs	% keyphrases absent from the text	% keyphrases absent from the corpus
CiteULike (Medelyan 2009)	180	21%	14%
Inspec (Hulth 2003)	2,000	27%	21%
SemEval-2010 (Kim *et al.* 2010)[a]	284	19%	11%
NUS Corpus (Nguyen and Kan 2007)[b]	211	32%	24%
Krapivin (Krapivin, Autayeu, and Marchese 2008)[b]	2,304	52%	35%
Average	996	32%	22%

[a] Percentual of not-in-text keyphrases, considering the author-assigned keyphrases.
[b] Percentual of absent keyphrases from the document abstract.

4.2 Models of knowledge

Different scenarios have been evaluated that extend the set of candidates by means of the inference process with the application of different knowledge models, such as knowledge base symbolic model, distributional semantics with the LSA, as well as a hybrid model that combines the symbolic model and distributional semantics.

- M1 – symbolic model of knowledge base using ConceptNet;
- M2 – symbolic model of knowledge base using WordNet;
- M3 – distributional semantics, with LSA executed in the corpus itself;
- M4 – distributional semantics, with LSA executed in TakeLab[e];
- M5 – hybrid model that combines the symbolic model with WordNet and the distributional semantics executed in the corpus.

The expansion of the candidate set by adding the non-text terms using a symbolic model of knowledge base (M1, M2, and M5) was performed using the inference strategy described in Section 3.2.2. Thus, the not-in-text terms using distributional semantics (M3, M4, and M5) were obtained according to the inference strategy described in Section 3.2.1, showing that the M3 and M5 models used their own corpus to construct the LSA matrix, while in the M4 model the LSA matrix was constructed through training the New York Times Annotated Corpus and Wikipedia.

4.3 Heuristics to characterize the relevance of candidate terms

To evaluate the use of heuristics that capture the relevance of not-in-text candidate terms with the keyphrase, we have varied the set of properties that characterize an example. Six sets of properties were used:

- A1 – has the same properties as KEA: (i) TFXIDF, (ii) first occurrence, and (iii) keyphraseness;
- A2 – has the same properties as the MAUI, that is, the same properties as KEA, plus six other properties: (i) amplitude of occurrence, (ii) semantic relationship, (iii) inverse linking on Wikipedia, (iv) term, (v) Wikipedia's keyphraseness, and (vi) degree of the node;
- A3 – has the same properties as MAUI, plus the ISR property.
- A4 – has the same properties as MAUI, plus the property of Silveira *et al.* (2015) – IFxIDF.
- A5 – has the same properties as MAUI, plus the SPI property.
- A6 – has the same properties as MAUI, plus the ISR, IFxIDF, and SPI properties.

[e]TakeLab provides two distributional word sets, where LSA was performed on the New York Times Annotated Corpus and Wikipedia.

https://doi.org/10.1017/S1351324919000342 Published online by Cambridge University Press

4.4 Data balancing

As the number of positive examples (terms that are keyphrases) is significantly smaller than the number of negative examples (terms that are not keyphrases), different existing data balancing strategies were applied in the training set: (i) random oversampling and (ii) synthetic minority oversampling technique (SMOTE) (Chawla *et al.* 2002), increasing by 200% examples in the minority class (in this case, positive examples).

In general, SMOTE and random oversampling have similar results; however, when the ISR heuristic is used, the random oversampling strategy outperforms the SMOTE. The interpretation is that the introduction of a Boolean property, such as ISR, can undermine the SMOTE process of generating synthetic examples. Thus, in the experiments with balanced data, the SMOTE balancing strategy was applied using the set of properties KEA (A1) and MAUI (A2) and the balancing random oversampling strategy was applied when using the other sets of properties (A3–A6).

4.5 Experimental setups

We configured the experiments with the following resources and tools:

- Encyclopedic knowledge base Wikipedia (snapshot on February 4th, 2014 with 12,853,780 articles);
- Knowledge base ConceptNet 5.0 (Liu and Singh 2004) and WordNet 3.1 (Miller 1995; Fellbaum 1998);
- Distributional Semantic Model TakeLab (Saric *et al.* 2012) and LSA (Deerwester *et al.* 1990);
- Lemmatizer Stanford CoreNLP Toolkit (Manning *et al.* 2014);
- Weka Framework (Hall *et al.* 2009) and the random forest algorithm (Breiman 2001).

The experiments were run on a machine with two 6-core Intel (R) Xeon (R) CPUs E5-2630L v2 @ 2.40GHz, with 40GB of RAM. Each corpus goes through all three steps described in Figure 1. For example, if we consider the corpus CiteULike, step (1) preprocessing is performed in approximately 3 h, step (2) inference strategy, using the inference of terms by means of WordNet, is performed in approximately 36 h, while step (3) machine learning is accomplished in approximately 156 h.

4.6 Results and discussion

To evaluate the inference process of not-in-text terms from different knowledge models, experiments were carried out by extending the set of candidates, adding, in addition to the terms of the text, not-in-text terms obtained from the models M1, M2, M3, M4, and M5 (described in Section 4.2). In each experiment, the f-measure was calculated separately for each set of candidate terms: (i) only examples representing the terms of the text T, (ii) only examples representing the not-in-text terms NT, and (iii) all examples, that is, terms of the text and not-in-text, $C = T \cup NT$, performing an exact match of the term classified as a keyphrase by the learning algorithm in relation to the keyphrases assigned by the author to the corresponding document.

For each set of candidates (T, NT, and C), the keyphrases assigned by the author according to the corresponding set are considered as gold standard, that is, when calculating the f-measure (F1-measure) for the set of candidates in the text (T), the gold standard is regarded as only the keyphrases assigned by the author that are present in the text of the document, and so consider a set of keyphrases smaller than the universe of all the terms assigned as keyphrases by the author. Thus, the f-measure is calculated using the exact match evaluation framework between the extracted keyphrases by machine learning algorithm and the gold standard keyphrases of the document corresponding.

https://doi.org/10.1017/S1351324919000342 Published online by Cambridge University Press

Table 3. Results in terms of f-measure of the learning of textual and not-in-text keyphrases for the CiteULike corpus

			With not-in-text candidate terms				
	K	Only text	M1	M2	M3	M4	M5
A1	T	0.489	0.487	0.496	0.495	0.492	**0.499**
	NT	0.000	0.000	0.000	0.000	0.000	**0.039**
	C	0.434	0.435	0.441	0.444	0.437	**0.451**
A2	T	0.561	0.575	0.573	0.569	0.567	**0.577**
	NT	0.000	0.010	0.024	0.018	0.000	**0.059**
	C	0.496	0.508	0.513	0.511	0.502	**0.531**
A3	T	0.561	0.574	0.576	0.565	0.564	**0.578**
	NT	0.000	0.017	0.033	0.029	0.009	**0.077**
	C	0.496	0.513	0.529	0.515	0.504	**0.533**
A4	T	0.561	0.572	0.577	0.566	0.563	**0.579**
	NT	0.000	0.018	0.035	0.031	0.007	**0.078**
	C	0.496	0.514	0.531	0.517	0.502	**0.535**
A5	T	0.561	0.573	0.575	0.564	0.566	**0.576**
	NT	0.000	0.017	0.036	0.027	0.012	**0.076**
	C	0.496	0.512	0.530	0.512	0.508	**0.531**
A6	T	0.561	0.573	0.575	0.570	0.560	**0.579**
	NT	0.000	0.019	0.059	0.034	0.017	**0.086**
	C	0.496	0.517	0.535	0.522	0.518	**0.538**

Tables 3–7 show the results (in respect to the f-measure, considering all the keyphrases classified by the learning algorithm) between the assessment scenarios for the CiteULike, Inspec, SemEval-2010, Krapivin, and NUS corpora, respectively, considering the balanced dataset with the SMOTE (Chawla *et al.* 2002) (A1 and A2) and random oversampling (A3–A6) methods. In the tables, the column *K* represents the keyphrases, where the value of *T* represents the f-measure calculated only for the candidate terms of the text, *NT* where the f-measure is calculated only for the not-in-text candidate terms, and *C* is calculated for the complete candidate set ($T \cup NT$). The column *Only text* shows the f-measure value, considering candidates only as the candidate terms of the text, whereas the columns *With not-in-text candidate terms* consider that the candidate set contains textual and not-in-text terms, according to the process of inference of not-in-text candidate terms, described in Section 3.2, from different knowledge models, symbolic model (M1 and M2), distributional semantics model (M3 and M4), and hybrid model (M5). The values highlighted in bold represent the best f-measure results in each experiment.

The data shown in Tables 3–7 indicate an increase in the f-measure value of the experiments whenever the set of candidates is improved with the addition of not-in-text terms inferred from a knowledge model. For example, for the CiteULike corpus (Table 3), using the MAUI property set (A2), the f-measure value changes from 49.6% (with the set of candidates consisting only of terms from the text) to 53.1% (with the set of candidates formed according to M5), representing an increase of 7.1% in the f-measure value. The same pattern in the results is verified for the other four corpora, considering the set of properties of the MAUI (A2) and two scenarios of variation of the set of candidates: (i) set of candidates formed by only terms from the text and (ii) set of candidates conforming to the M5 model. In the Inspec corpus (Table 4), the f-measure increases from 32.4% to 33.2%, representing an increase of 2.5%; for the SemEval-2010 corpus (Table 5), the f-measure rose from 22.7% to 24.3%, representing an increase of 7.0%; for the Krapivin corpus

https://doi.org/10.1017/S1351324919000342 Published online by Cambridge University Press

Table 4. Results in terms of f-measure of the learning of textual and not-in-text keyphrases for the Inspec corpus

			With not-in-text candidate terms				
	K	Only text	M1	M2	M3	M4	M5
A1	T	0.192	0.195	0.196	0.198	0.200	**0.204**
	NT	0.000	0.016	0.021	0.027	0.031	**0.047**
	C	0.189	0.192	0.193	0.195	0.198	**0.202**
A2	T	0.331	0.334	0.336	0.335	0.336	**0.337**
	NT	0.000	0.046	0.073	0.051	0.065	**0.096**
	C	0.324	0.328	0.330	0.329	0.330	**0.332**
A3	T	0.331	0.335	0.348	0.338	0.341	**0.353**
	NT	0.000	0.053	0.107	0.067	0.074	**0.138**
	C	0.324	0.329	0.343	0.332	0.336	**0.346**
A4	T	0.331	0.334	0.349	0.339	0.340	**0.358**
	NT	0.000	0.050	0.110	0.069	0.073	**0.141**
	C	0.324	0.327	0.347	0.334	0.335	**0.347**
A5	T	0.331	0.337	0.347	0.337	0.339	**0.352**
	NT	0.000	0.059	0.109	0.065	0.072	**0.135**
	C	0.324	0.331	0.345	0.330	0.333	**0.343**
A6	T	0.331	0.342	0.364	0.346	0.360	**0.365**
	NT	0.000	0.093	0.125	0.100	0.109	**0.157**
	C	0.324	0.339	0.345	0.340	0.343	**0.347**

(Table 6), the f-measure increases from 21.6% to 23.7%, representing an increase of 9.7%; and for the NUS corpus (Table 7), the f-measure increases from 24.7% to 26.5%, representing an increase of 7.3%.

In the CiteULike corpus (Table 3), for the experiment applying the M5 model and the A6 property set, we obtained a f-measure result of 57.9% for the set of text candidates (T), 8.6% for the candidate set not-in-text (NT) and 53.8% for the total set of candidates ($C = T \cup NT$). It should be noted that the gold standard differs in each set of candidates. When calculating the f-measure of the set T, the gold standard is only considering the keyphrases assigned by the author that are in the text, whereas in calculating the f-measure of the set C the gold standard is considering all the keyphrases attributed by the author (whether present or not in the text).

As mentioned, it is assumed that the keyphrases missing from the text have a semantic relationship with the terms present in the text, either by expressing a generalization of specific terms of the text (hyperonymy) or being the synonym of the terms that represent the subject of the text (synonymy). In an analysis carried out in the corpora, it can be seen that most of the missing keyphrases in the text have a relation of hyperonymy or synonymy with terms of the text. For example, in a document from the CiteULike corpus dealing with the maintenance of Borneo's biodiversity, the threat of extinction of orangutans is explicitly described in the text. From WordNet, it is possible to infer the relation *{great ape, hypernym, orangutan}*, "great ape" being a non-text term assigned as a keyphrase by the authors, representing a generalization of orangutans. In another document of the same corpus that deals with the social interaction between people, the authors analyze in the text the use of the mobile phone in communication among people; however, they attribute the term cell as a keyphrase of the document. From WordNet, it is possible to infer the relation {*cell*, *synonym*, *mobilephone*}. This way cell, synonymous with mobile phone, is added to the set of not-in-text candidates and classified as not-in-text keyphrase. In the

https://doi.org/10.1017/S1351324919000342 Published online by Cambridge University Press

Table 5. Results in terms of f-measure of the learning of textual and not-in-text keyphrases for the SemEval-2010 corpus

			With not-in-text candidate terms				
	K	Only text	M1	M2	M3	M4	M5
A1	T	0.207	0.216	0.233	0.229	0.220	**0.237**
	NT	0.000	0.006	0.008	0.007	0.006	**0.012**
	C	0.171	0.179	0.191	0.188	0.182	**0.195**
A2	T	0.283	0.286	0.300	0.287	0.296	**0.303**
	NT	0.000	0.012	0.028	0.021	0.023	**0.031**
	C	0.227	0.231	0.241	0.231	0.238	**0.243**
A3	T	0.283	0.287	0.311	0.291	0.303	**0.313**
	NT	0.000	0.017	0.055	0.042	0.045	**0.061**
	C	0.227	0.231	0.252	0.236	0.244	**0.255**
A4	T	0.283	0.289	0.314	0.294	0.302	**0.317**
	NT	0.000	0.018	0.059	0.045	0.048	**0.068**
	C	0.227	0.232	0.255	0.239	0.245	**0.260**
A5	T	0.283	0.290	0.312	0.290	0.303	**0.313**
	NT	0.000	0.020	0.056	0.043	0.047	**0.064**
	C	0.227	0.233	0.253	0.236	0.245	**0.258**
A6	T	0.283	0.295	0.324	0.299	0.308	**0.327**
	NT	0.000	0.021	0.081	0.069	0.072	**0.090**
	C	0.227	0.237	0.263	0.243	0.257	**0.264**

CiteULike corpus, in an analysis of the candidate set improved using WordNet, approximately 58% of the not-in-text keyphrases have a hyperonymic relationship with terms of the text, 37% have a synonymic relation, and 5% have a different type of semantic relation. In the corpus Inspec, approximately 52% of the not-in-text keyphrases have a hyperonymic relation with terms of the text, 33% have a synonymic relation, and 15% have another type of semantic relation. In the SemEval-2010 corpus, about 40% of the not-in-text keyphrases have a hyperonymic relation with terms of the text, 47% have a synonymic relationship, and 13% have another type of semantic relation.

The symbolic models used in the experiments make use of external resources, more specifically, use the knowledge base, to obtain not-in-text terms that have different semantic relations with the terms of the term. Considering that most of the missing keyphrases of the text captured by the knowledge bases, for example WordNet, are obtained through synonymic or hyperonymic relations, it is assumed that the knowledge base symbolic models extract more generic not-in-text terms.

The distributional semantics models used in the experiments construct an LSA matrix by exploiting the co-occurrence of words in a corpus and make use of a measurement of similarity to obtain the terms present in the LSA matrix that are more similar to the terms of the text. When using the corpus itself in the distributional semantics model, it is assumed that not-in-text terms specific to the corpus domain are obtained, since they are present in other corpus documents. Some of the not-in-text keyphrases are contained in the corpus documents, for example, in the CiteULike corpus, about 35% of the not-in-text keyphrases are present in other corpus documents.

The hybrid model (M5) obtains the best results. Our interpretation is that the combination of a symbolic model with a distributional semantic model enables the capturing from external

https://doi.org/10.1017/S1351324919000342 Published online by Cambridge University Press

Table 6. Results in terms of f-measure of the learning of textual and not-in-text keyphrases for the Krapivin corpus

			With not-in-text candidate terms				
	K	Only text	M1	M2	M3	M4	M5
A1	T	0.279	0.283	0.288	0.283	0.285	**0.292**
	NT	0.000	0.162	0.178	0.169	0.174	**0.183**
	C	0.154	0.206	0.213	0.209	0.211	**0.218**
A2	T	0.294	0.302	0.310	0.306	0.308	**0.319**
	NT	0.000	0.192	0.198	0.192	0.195	**0.207**
	C	0.216	0.225	0.230	0.226	0.229	**0.237**
A3	T	0.294	0.314	0.323	0.319	0.320	**0.327**
	NT	0.000	0.201	0.213	0.207	0.209	**0.219**
	C	0.216	0.233	0.241	0.237	0.239	**0.245**
A4	T	0.294	0.312	0.329	0.317	0.322	**0.330**
	NT	0.000	0.200	0.217	0.205	0.215	**0.223**
	C	0.216	0.231	0.246	0.233	0.244	**0.248**
A5	T	0.294	0.315	0.320	0.314	0.317	**0.325**
	NT	0.000	0.205	0.217	0.203	0.211	**0.221**
	C	0.216	0.236	0.245	0.230	0.239	**0.245**
A6	T	0.294	0.323	0.332	0.326	0.330	**0.338**
	NT	0.000	0.215	0.224	0.219	0.221	**0.230**
	C	0.216	0.242	0.249	0.244	0.247	**0.253**

resources the more generic not-in-text terms inferred from the symbolic model of the WordNet, as well as capturing the not-in-text terms more specifically related to the corpus domain, augmenting the number of not-in-text candidate terms that are in fact keyphrases with a semantic relationship with the text and is not restricted to deducing only terms presented in the corpus. More specifically, in a text about health care, the symbolic model of the WordNet enables the capturing of the not-in-text term disease, while the distributional semantic model LSA enables the capturing of the not-in-text terms that are more specific with the corpus domain, such as breast cancer.

The extension of the set of keyphrase candidates with the not-in-text terms obtained from a knowledge model enables an improvement in the accuracy of the learning algorithms and overcomes the limitation of the state-of-the-art systems that consider as candidates only the terms of the text and/or training corpus. However, it is assumed that it is not enough to only increase the pool of candidates with not-in-text terms. To increase the accuracy of learning algorithms, it is also necessary to have heuristics that characterize the relevance of not-in-text candidate terms to keyphrases.

We performed experiments using the set of candidates formed by text terms and not-in-text terms obtained by the hybrid model (M5), varying the properties of the candidate terms: A1, A2, A3, A4, A5, and A6 (as described in Section 4.3). It should be noted that in scenarios A1 to A6, the properties (i) TFxIDF, (ii) first occurrence, and (iii) range of occurrence are not applied to characterize not-in-text terms, since they make use of information in the text of the document, for example, the TFxIDF property uses the frequency of the term occurrence in the text of the document. Therefore, these properties have no value for not-in-text terms. In the scenarios A3 to A6, the properties (i) ISR, (ii) SPI, and (iii) IFxIDF are used to characterize the relevance of not-in-text terms, not having a value for the terms of the text.

https://doi.org/10.1017/S1351324919000342 Published online by Cambridge University Press

Table 7. Results in terms of f-measure of the learning of textual and not-in-text keyphrases for the NUS corpus

			With not-in-text candidate terms				
	K	Only text	M1	M2	M3	M4	M5
A1	T	0.303	0.308	0.313	0.311	0.310	**0.316**
	NT	0.000	0.055	0.057	0.056	0.055	**0.063**
	C	0.232	0.239	0.243	0.242	0.241	**0.246**
A2	T	0.325	0.329	0.333	0.329	0.331	**0.337**
	NT	0.000	0.105	0.108	0.107	0.107	**0.110**
	C	0.247	0.258	0.262	0.248	0260	**0.265**
A3	T	0.325	0.333	0.345	0.337	0.343	**0.347**
	NT	0.000	0.125	0.131	0.127	0.130	**0.133**
	C	0.247	0.262	0.271	0.266	0.270	**0.273**
A4	T	0.325	0.333	0.343	0.339	0.341	**0.348**
	NT	0.000	0.126	0.130	0.128	0.129	**0.134**
	C	0.247	0.263	0.269	0.267	0.268	**0.274**
A5	T	0.325	0.332	0.343	0.338	0.344	**0.347**
	NT	0.000	0.125	0.130	0.127	0.129	**0.135**
	C	0.247	0.262	0.269	0.266	0.270	**0.274**
A6	T	0.325	0.335	0.346	0.337	0.342	**0.350**
	NT	0.000	0.127	0.133	0.129	0.131	**0.139**
	C	0.247	0.264	0.270	0.266	0.269	**0.276**

The addition of heuristics to characterize the relevance of not-in-text terms improves the accuracy of learning algorithms. For the CiteULike corpus, the f-measure changes from 53.3% in scenario A2 (set of MAUI properties) to 53.8% in scenario A6 (set of properties of MAUI + ISR + IFxIDF + SPI), representing an increase of about 1%. The same pattern of results is verified for the other corpora. In the Inspec corpus, the f-measure increases from 33.2% (A2) to 34.7% (A6), representing an increase of approximately 5%; in the SemEval-2010 corpus, the f-measure increases from 24.3% (A2) to 26.4% (A6), representing an increase of approximately 9%; in the Krapivin corpus, the f-measure increases from 23.7% (A2) to 25.3% (A6), representing an increase of approximately 7%; and in the NUS corpus, the f-measure increases from 26.5% (A2) to 27.6% (A6), representing an increase of approximately 4%.

The results presented in Tables 3–7 show the importance of the new heuristics to aid the algorithm in dealing with the learning of not-in-text candidates obtained by the inference process. In addition to improving the set of candidates that includes not-in-text terms inferred from a knowledge model, it is necessary to add new heuristics to characterize the relevance of not-in-text terms to increase the accuracy of learning algorithms in the classification of not-in-text keyphrases.

Considering the results obtained in the experiments with the use of the ISR, IFxIDF, and SPI properties (Tables 3–7), it is noticed that by adding one of these properties to the MAUI properties set (A3–A5), the learning algorithms can distinguish the terms that are not-in-text candidates, improving the accuracy of the algorithms. By combining all these properties (A6), the accuracy of learning algorithms is even greater, improving the extraction quality of not-in-text keyphrases.

The heuristics defined in this work for the not-in-text terms capture the descriptive and discriminative capacity of a specific document. Considering the results of the experiments with the addition of the ISR, IFxIDF, and SPI properties, it is understood that the keyphrases missing from the text: (i) have a semantic relation with the terms of the text, allowing the inference of these

https://doi.org/10.1017/S1351324919000342 Published online by Cambridge University Press

Title: Reconstructing surfaces by volumetric regularization using radial basis functions

We present a new method of surface reconstruction that generates smooth and seamless models from sparse, noisy, nonuniform, and low resolution range data. Data acquisition techniques from computer vision, such as stereo range images and space carving, produce 3D point sets that are imprecise and nonuniform when compared to laser or optical range scanners. Traditional reconstruction algorithms designed for dense and precise data do not produce smooth reconstructions when applied to vision-based data sets. Our method constructs a 3D implicit surface, formulated as a sum of weighted radial basis functions. We achieve three primary advantages over existing algorithms: (1) the implicit functions we construct estimate the surface well in regions where there is little data, (2) the reconstructed surface is insensitive to noise in data acquisition because we can allow the surface to approximate, rather than exactly interpolate, the data, and (3) the reconstructed surface is locally detailed, yet globally smooth, because we use radial basis functions that achieve multiple orders of smoothness.

Keyphrases: **computer vision**, image reconstruction, image texture, **interpolation**, surface fitting, surfaces reconstruction, **volumetric regularization**, **radial basis functions**, sparse range data, noisy data, nonuniform data, low-resolution range data, data acquisition techniques, stereo range images, space carving, 3D point sets, vision-based data sets, 3D implicit surface, weighted radial basis functions

Terms classified as keyphrases by the learning algorithm: **computer vision**, **volumetric regularization**, **radial basis functions**, technical, **interpolation**, render

Figure 3. An example of the keyphrases extracted with the application of the proposed method for a document from corpus Inspec.

terms from models of knowledge, (ii) have a greater semantic relationship with the text if compared with the terms of the text that inferred them (according to the ISR heuristic), (iii) have a descriptive power, that is, they are deduced from terms with high semantic value for the document (according to the SPI heuristic), and (iv) have a discriminatory power, that is, they are deduced from several terms of a specific document and are not often deduced from other documents.

Figure 3 shows a document of the Inspec corpus with the keyphrases extracted by the proposed method with the inference of the not-in-text terms from the model M5 and with the set of properties A6. In the figure, the underlined terms correspond to the keyphrases present in the text, the terms shaded in gray correspond to the candidate terms and/or keyphrases that are not in the text and that were obtained through the inference process, as well as the terms that are highlighted in bold corresponding to the keyphrases correctly extracted by the learning algorithm.

The improvement of the set of candidates, in addition to the terms of the text, with the not-in-text terms obtained from a knowledge model, and the use of heuristics that characterize the relevance of the not-in-text terms to the keyphrases allow the algorithm to learn not-in-text terms, increasing accuracy and overcoming the limitation of traditional keyphrase extraction systems that extract only keyphrases present in the text and/or in the corpus. In the example shown in Figure 3, it is noted that the learning algorithm correctly extracted one of the absent keyphrases from the text attributed by people.

It is important to emphasize that, independently of the knowledge representation model used, our method also improves the learning of keyphrases derived from terms that are in the text, that is, besides improving the accuracy in the identification of not-in-text keyphrases, it is possible to verify an improvement in the f-measure for the terms of the text. For instance, comparing the results obtained in the CiteULike corpus with the terms of the text in experiment A2 with the training set only with the terms of the text and with the addition of not-in-text terms from the hybrid model (WordNet and LSA executed in the corpus) of the CiteULike corpus, we obtained 420 and 404 true positives for the examples that are keyphrases, respectively. Our hypothesis is that the improvement of the training set with the not-in-text terms ends up giving the classification algorithm greater power of discrimination and thus also impacting the learning of keyphrases that are in the texts of the corpus.

Table 8 shows the results (relative to f-measure) for the top 5, top 10, top 15, as well as for all terms classified as keyphrases by the learning algorithm for the corpora CiteULike, Inspec, SemEval-2010, Krapivin, and NUS Corpus. The scores given by each tree of the random forest

https://doi.org/10.1017/S1351324919000342 Published online by Cambridge University Press

Table 8. Results in terms of f-measure for keyphrase extraction using knowledge models to top 5, 10, and 15 keyphrases

	K	Top 5	Top 10	Top 15	All keyphrases
CiteULike (Medelyan 2009)	T	0.501	0.513	0.565	**0.579**
	NT	0.017	0.036	0.059	**0.086**
	C	0.505	0.515	0.521	**0.538**
Inspec (Hulth 2003)	T	0.325	0.343	0.352	**0.365**
	NT	0.050	0.114	0.123	**0.157**
	C	0.319	0.337	0.340	**0.347**
SemEval-2010 (Kim *et al.* 2010)	T	0.292	0.305	0.318	**0.327**
	NT	0.035	0.069	0.075	**0.090**
	C	0.220	0.239	0.251	**0.264**
Krapivin (Krapivin, Autayeu, and Marchese 2008)	T	0.298	0.313	0.324	**0.338**
	NT	0.199	0.205	0.218	**0.230**
	C	0.223	0.231	0.244	**0.253**
NUS Corpus (Nguyen and Kan 2007)	T	0.321	0.335	0.342	**0.350**
	NT	0.098	0.118	0.127	**0.139**
	C	0.252	0.259	0.263	**0.276**

Table 9. Results in terms of f-measure for extraction of keyphrases absent from the text and the corpus using knowledge models

K	CiteULike (Medelyan 2009)	Inspec (Hulth 2003)	SemEval-2010 (Kim *et al.* 2010)	Krapivin (Krapivin, Autayeu, and Marchese 2008)	NUS Corpus (Nguyen and Kan 2007)
T	0.579	0.365	0.327	0.338	0.350
NT_c	0.042	0.113	0.037	0.153	0.098
NT	0.086	0.157	0.090	0.230	0.139
C	0.538	0.347	0.264	0.253	0.276

algorithm to each candidate term are averaged and used to produce the final ranking of terms to keyphrases, obtaining the top 5, top 10, and top 15 keyphrases. For these experiments, we used (i) the inference of the not-in-text terms from the hybrid model that combines WordNet and LSA executed in the corpus itself (M5), and (ii) the set of properties A6 (according to Section 4.3). It is verified that for all corpora, the results are better when considering all the terms classified as keyphrases by the classification algorithm. For example, for the CiteULike corpus, when considering only the top 5 keyphrases, we have a result of f-measure for the set C of 50.5%, whereas when considering all keyphrases, we have a f-measure result of 53.8%, that is, there is an increase of 6.5%.

Table 9 presents the results (in relation to f-measure) for the set of text candidates (T), set of not-in-text candidates absent from all corpus documents (NT_c), set of non-text candidates (NT), and the set of all candidate terms ($C = T \cup NT$) for the corpora CiteULike, Inspec, SemEval-2010, Krapivin, and NUS Corpus. For these experiments, we used (i) the inference of the not-in-text terms from the hybrid model that combines WordNet and LSA executed in the corpus itself (M5) and (ii) the set of properties A6 (according to Section 4.3). It can be seen that the use of the M5 model (which combines the symbolic model of knowledge base with the distributional semantic model) allows us to infer as candidates not-in-text terms absent from the text as well as not-in-text terms absent from the corpus.

https://doi.org/10.1017/S1351324919000342 Published online by Cambridge University Press

Table 10. Results in terms of f-measure top 10 to the text terms and in terms of recall top 50 to the not-in-text terms for the Inspec, SemEval-2010, Krapivin, and NUS corpora comparing with the state-of-the-art systems

	K	Inspec	SemEval-2010	Krapivin	NUS
TopicCoRank (Bougouin, Boudin, and Daille 2016)	T	0.323	0.258	0.272	0.278
	NT	0.041	0.023	0.142	0.050
Meng *et al.* (2017)	T	0.342	0.304	0.266	0.326
	NT	0.100	0.067	0.202	0.116
Silveira *et al.* (2015) (A3)	T	0.327	0.283	0.312	0.335
	NT	0.091	0.043	0.201	0.119
MAUI + ISR + TFxIDF + SPI (A6)	T	**0.343**	**0.305**	**0.313**	**0.335**
	NT	**0.132**	**0.077**	**0.215**	**0.120**

4.6.1 Evaluation of the proposed method with state-of-the-art systems

One limitation of traditional keyphrase extraction systems is that the set of keyphrase candidates is formed only by terms present in the text. An improvement of this limitation is made by Bougouin *et al.* (2016) using TopicCoRank, a method for identifying keyphrases based on a unified graph that connects two graphs representing the document topics, the controlled keyphrases, and the relations between them. Meng *et al.* (2017) propose a model to extract keyphrases missing from the text, but which are present in the training corpus and Silveira *et al.* (2015) define a heuristic for not-in-text candidate terms. Table 10 presents the results in terms of f-measure top 10 to the text terms and in terms of recall top 50 to the not-in-text terms of the method proposed in this work with Inspec, SemEval-2010, Krapivin, and NUS corpora, and those obtained by state-of-the-art systems by Meng *et al.* (2017) and Silveira *et al.* (2015). The values highlighted in bold represent the best f-measure results in each corpus.

With the Inspec corpus, TopicCoRank obtained f-measures top 10 of 32.3% and recall top 50 of 4.1% for keyphrases present in the text and for keyphrases absent from the text, respectively. Meng *et al.* (2017) obtained f-measure top 10 of 34.2% for keyphrases in the text and recall top 50 of 10.0% for keyphrases absent from the text. Silveira *et al.* (2015) obtained 32.7% for keyphrases in the text and 9.1% for keyphrases absent from the text. In our approach with the Inspec corpus, we obtained 34.3% for keyphrases in the text and 13.2% for keyphrases absent from the text. With the SemEval-2010 corpus, TopicCoRank obtained f-measures top 10 of 25.8% and recall top 50 of 2.3% for keyphrases that are present in the text and absent from the text, respectively. Meng *et al.* (2017) obtained an f-measure top 10 of 30.4% for keyphrases in the text and recall top 50 of 6.7% for keyphrases absent from the text, Silveira *et al.* (2015) obtained 28.3% for keyphrases in the text and 4.3% for keyphrases absent from the text, whereas our approach obtained 30.5% for keyphrases in the text and 7.7% for keyphrases absent from the text. In the Krapivin corpus, we obtained f-measure top 10 of 31.3% for keyphrases in the text and recall top 50 of 21.5% for keyphrases not-in-text. Silveira *et al.* (2015) obtained 31.2% for keyphrases in the text and 20.1% for keyphrases not-in-text. Meng *et al.* (2017) obtained 26.6% for keyphrases in the text and 20.2% for keyphrases not-in-text, and TopicCoRank obtained f-measures top 10 of 27.2% and recall top 50 of 14.2% for keyphrases present in the text and for keyphrases absent from the text, respectively. For the NUS corpus, TopicCoRank obtained f-measures top 10 of 27.8% and recall top 50 of 5.0% for keyphrases present in the text and for keyphrases absent from the text, respectively. Meng *et al.* (2017) obtained an f-measure top 10 of 32.6% for keyphrases in the text and recall top 50 of 11.6% for keyphrases absent from the text. Silveira *et al.* (2015) obtained 33.5% for keyphrases in the text and 11.9% for keyphrases absent from the text. In our approach with the Inspec corpus, we obtained 33.5% for keyphrases in the text and 12.0% for keyphrases absent from the text.

https://doi.org/10.1017/S1351324919000342 Published online by Cambridge University Press

A way to deal with the limitation of the keyphrase extraction methods of classifying only textual terms is to improve the training set with examples of not-in-text terms. The approach of TopicCoRank (Bougouin, Boudin, and Daille 2016) and Meng *et al.* (2017) only considers as not-in-text candidates the terms presented in documents of the corpus. Consequently, this approach is not able to recall any not-in-text keyphrases that are not contained in the corpus. Not all the not-in-text keyphrases are contained in the documents of the corpus, for instance, in the CiteULike corpus, 65% of the not-in-text keyphrases are not in the corpus. Silveira *et al.* (2015) only deduct the not-in-text candidates from a symbolic model of knowledge base and define a new feature to characterize these terms. Our approach is more generic, allowing us to extract the not-in-text candidate terms from external and internal resources, considering the symbolic model which uses an external knowledge base and the distributional semantic model which uses an LSA matrix performed in its own corpus. Thus, we can extract as candidates not-in-text keyphrases that are present as well as absent from the corpus, augmenting the training set with examples of not-in-text terms that are in fact keyphrases. Not only this, but there are features defined in this paper that capture the relevance of not-in-text candidates, enabling the classification algorithm to learn the not-in-text keyphrases.

Our method outperforms the results of TopicCoRank (Bougouin, Boudin, and Daille 2016), Meng *et al.* (2017), and Silveira *et al.* (2015). Our claim is that this improvement is due to two aspects: (i) the knowledge representation models used by our approach allow us to identify a set of not-in-text terms that have a semantic relationship with the text, which are potential candidates for keyphrases and that can be present or absent from the corpus; (ii) although the not-in-text terms have a semantic relationship with the textual terms, not all terms are relevant for keyphrases. Therefore, the features created and used to characterize the not-in-text terms allow the classification algorithm to learn with more quality the not-in-text terms that are relevant to keyphrases.

5. Conclusion

In this paper, we show that the keyphrase extraction task can be improved by extending the set of candidates of keyphrases with not-in-text terms inferred from textual terms, applying the process to infer not-in-text terms from different knowledge models as well as applying heuristics that characterize the relevance of these terms to the keyphrase. The intuition of this method is that the missing keyphrases of the text have a semantic relationship with the content of the text, either by indicating a synonym or a generalization of the terms of the text. This process improves the pool of candidates and overcomes the limitation of traditional keyphrase extraction systems that use only text terms as candidates. An empirical evaluation has shown that this strategy has increased the accuracy of keyphrase extraction systems.

It should be noted that the task of keyphrase extraction is characterized by an unbalanced data problem, where the number of positive examples (terms that are keyphrases) is significantly less than the number of negative examples (terms that are not key words). The improvement of the set of candidates from the inference process accentuates the imbalance of the data. In this way, the method proposes the use of existing data balancing strategies in the training set to lessen the imbalance between the classes.

Not all deduced not-in-text terms in the knowledge base are good candidates for keyphrases. Beginning with the empirical analysis, we defined heuristics that measure the relevance of the inferred terms, which are provided as input, so that the machine learning algorithms also learn the not-in-text terms.

We discovered, by empirical analysis in five datasets, that the accuracy of keyphrase extraction can be improved substantially, overcoming the limitation of extracting from not-in-text keyphrases, as well as increasing the accuracy of the text terms. The hypothesis is that the addition of not-in-text terms, with their respective heuristic values, allows for the classification algorithm to also learn the text terms.

https://doi.org/10.1017/S1351324919000342 Published online by Cambridge University Press

This line of research opens many possibilities for further study and development. We need to apply other strategies that exist in the literature to handle the unbalanced data inherent in the task of automatic extraction of keyphrases, such as that of Sterckx *et al.* (2016). We still need to analyze the combination of strategies to infer not-in-text terms, such as Word2Vec (Mikolov *et al.* 2013), GloVe (Pennington, Socher, and Manning 2014), FastText (Bojanowski *et al.* 2017), Doc2Vec (Le and Mikolov 2014), Sent2Vec (Pagliardini, Gupta, and Jaggi 2017), and Key2Vec (Debanjan *et al.* 2018a) to capture more generic not-in-text terms, as well as specific terms to the domain of the document. We intend to investigate whether there is a taxonomic relation between textual and not-in-text keyphrases. Thus, following the presented approach here, we plan to explore the taxonomic nomination of keyphrases, using lexical knowledge bases, such as WordNet. It should be pointed out that methods based on multilayered neural networks (deep learning) have been used with great success in various NLP tasks (Zhang, Zhao, and Lecun 2015; Zhang *et al.* 2016; Meng *et al.* 2017). A future line of research to be investigated should be the adaptation of the proposed method to be used in algorithms of this type, as well as to implement and adapt the approach proposed by Meng *et al.* (2017) according to the inference strategy proposed in this work. We also suggest the adaptation of the method proposed in this work with methods of automatic extraction of keyphrases using thesaurus of specific domain, such as KEA++ (Medelyan and Witten 2006).

References

Ammar W., Peters M.E., Bhagavatula C. and Power R. (2017). The AI2 system at SemEval-2017 task 10 (ScienceIE): semi-supervised end-to-end entity and relation extraction. In *Proceedings of the 11th International Workshop on Semantic Evaluations (SemEval-2017)*. Vancouver: Association for Computational Linguistics.

Arampatzis A.T., Tsoris T., Koster C.H.A. and van der Weide Th.P. (1998). Phrase-based information retrieval. In *Information Processing and Management*, pp. 693–707.

Berend G. (2011). Opinion expression mining by exploiting keyphrase extraction. In *Proceedings of the 5th International Joint Conference on Natural Language Processing*. Chiang Mai: Asian Federation of Natural Language Processing, pp. 1162–1170.

Bojanowski P., Grave E., Joulin A. and Mikolov T. (2017). Enriching word vectors with subword information. In arXiv:1607.04606.

Bougouin A., Boudin F. and Beatrice D. (2013). Topicrank: Graph-based topic ranking for keyphrase extraction. In *Proceedings of the 6th International Joint Conference on Natural Language Processing*. Nagoya: Asian Federation of Natural Language Processing, pp. 543–551.

Bougouin A., Boudin F. and Daille B. (2016). Keyphrase annotation with graph co-ranking. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*. Osaka: The COLING 2016 Organizing Committee, pp. 2945–2955.

Breiman L. (2001). Random forests. Machine Learning. Dordrecht: Kluwer Academic Publishers, pp. 5–32.

Cândido Júnior A., Magalhães C., Caseli H. and Zangirolami R. (2015). Topic modeling for keyword extraction: using natural language processing methods for keyword extraction in portal Min@s. *Revista de Estudos da Linguagem - RELIN* **23**(3), 695–726.

Chawla N.V., Bowyer K.W., Hall L.O. and Kegelmeyer W.P. (2002). SMOTE: synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research (JAIR)* **16**, 321–357.

Chen J., Zhang X., Wu Y., Yan Z. and Li Z. (2018) Keyphrase generation with correlation constraints. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Brussels: Association for Computational Linguistics, pp. 4057–4066.

Danesh S., Sumner T. and Martin J.H. (2015). SGRank: combining statistical and graphical methods to improve the state of the art in unsupervised keyphrase extraction. *Lexical and Computational Semantics*. Denver, CO: Association for Computational Linguistics.

Debanjan M., Kuriakose J., Shan R.R. and Zimmermann R. (2018a). Key2Vec: automatic ranked keyphrase extraction from scientific articles using phrase embeddings. In *Proceedings of Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT 2018)*. New Orleans, LA: Association for Computational Linguistics, pp. 634–639.

Debanjan M., Kuriakose J., Shan R.R., Zimmermann R. and Talburt J.R. (2018b). Theme-weighted ranking of keywords from text documents using phrase embeddings. In *arXiv:1807.05962v1*.

Deerwester S., Dumais S. T., Furnas G. W., Landauer T. K. and Harshman R. (1990). Indexing by latent semantic analysis. *Journal of the American Society for Information Science* **41**, 391–407.

https://doi.org/10.1017/S1351324919000342 Published online by Cambridge University Press

Domingos P. and Pazzani M. (1997). On the optimality of the simple bayesian classifier under zeroone loss. *Machine Learning* **29**, 103–130.

El-Beltagy S.R. and Rafea A. (2010). Kpminer: participation in Semeval-2. In *Proceedings of the 5th International Workshop on Semantic Evaluation*. Uppsala: Association for Computational Linguistics, pp. 190–193.

Fellbaum C. (1998). WordNet: An Electronic Lexical Database. Cambridge, *MA*: MIT Press.

Frank E., Paynter G.W., Witten I.H., Gutwin C. and Nevill-Manning C.G. (1999). Domain-specific keyphrase extraction. In: *Proceedings of the 16th International Joint Conference on Artificial Intelligence*, Stockholm. San Francisco, CA: Morgan Kaufmann Publishers, pp. 668–673.

Grineva M., Grinev M. and Lizorkin D. (2009). Extraction key terms from noisy and multi-theme documents. In *Proceedings of the 18th International Conference on World Wide Web*. Madrid: ACM, pp. 661–670.

Gutwin C., Paynter G., Witten I.H., Nevill-Manning C.G. and Frank E. (1998). Improving browsing in digital libraries with keyphrase indexes. Technical report, Department of Computer Science, University of Saskatchewan, Canada.

Hall M., Frank E., Holmes G., Pfahringer B., Reutermann P. and Witten I.H. (2009). The WEKA data mining software: an update. *SIGKDD Explorations* **11**(1), 10–18.

Han J., Kim T. and Choi J. (2007). Web document clustering by using automatic keyphrase extraction. In *Proceedings of the 2007 IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology*, Washington, DC, USA. IEEE Computer Society, pp. 56–59.

Hulth A. (2003). Improved automatic keyword extraction given more linguistic knowledge. In *Proceedings of the 2003 Conference on Empirical Methods in Natural Language Processing*. Stroudsburg, PA: ACM, pp. 216–223.

Hulth A. and Megyesi B.B. (2016). A study on automatically extracted keywords in text categorization. In: *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*. Association for Computational Linguistics. Sydney: ACM, pp. 537–544.

Kim S.N., Medelyan O., Kan M.Y. and Baldwin T. (2010). Automatic keyphrase extraction from scientific articles. *Language Resources and Evaluation* **47**, 723–742.

Krapivin M., Autayeu A. and Marchese M. (2008). Large dataset for keyphrases extraction. Technical report disi-09-055, DISI, Trento, Italy.

Langville A.N. and Meyer C.D. (2004). Deeper inside pagerank. *Internet Mathematics* **1**(3), 335–380.

Le Q. and Mikolov T. (2014). Distributed representations of sentences and documents. In: *Proceedings of the 31st International Conference on Machine Learning (ICML 2014)*. Beijing: JMLR, pp. 1188–1196.

Litvak M. and Last M. (2008). Graph-based keyword extraction for single-document summarization. In *Proceedings of the Workshop on Multi-Source Multilingual Information Extraction and Summarization*, Stroudsburg, PA, USA. Association for Computational Linguistics, pp. 17–24.

Liu H. and Singh P. (2004). ConceptNet: a practical commonsense reasoning tool-kit. *BT Technology Journal* **22**(4), 211–226.

Liu Z., Huang W., Zheng Y. and Sun M. (2010). Automatic keyphrase extraction via topic decomposition. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*.

Manning C.D., Surdeanum M., Bauer J., Finkel J., Bethard S.J. and McClosky D. (2014). The Stanford CoreNLP natural language processing toolkit. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*. Baltimore, MD: Association for Computational Linguistics.

Medelyan O. (2005). Automatic keyphrase indexing with a domain-specific thesaurus. Master thesis. Albert-Ludwigs-Universitaet Freiburg im Breisgau, Germany.

Medelyan O. (2009). Human-competitive automatic topic indexing. PhD thesis. Department of Computer Science, University of Waikato, New Zealand.

Medelyan O. and Witten I.H. (2006). Thesaurus based automatic keyphrase indexing. In *Joint Conference on Digital Libraries 2006 (JCDL06)*.

Medelyan O. and Witten I.H. (2008). Domain independent automatic keyphrase indexing with small training sets. *Journal of the American Society for Information Science and Technology* **59**, 1026–1040.

Medelyan O., Frank E. and Witten I.H. (2009). Human-competitive tagging using automatic keyphrase extraction. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*. EMNLP '09, vol. 3. Stroudsburg, PA, USA, pp. 1318–1327.

Meng R., Zhao S., Han S., He D., Brusilovsky P. and Chi Y. (2017). Deep keyphrase generation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (ACL 2017)*. Vancouver: Association for Computational Linguistics.

Mihalcea H. and Tarau P. (2004). TextRank: bringing order into texts. In *Association for Computational Linguistics*. Barcelona, pp. 404–411.

Mihalcea R. and Andra C. (2007). Wikify!: Linking documents to encyclopedic knowledge. In *Proceedings of the sixteenth ACM conference on Conference on information and knowledge management (CIKM07)*. Lisbon: ACM, pp. 233–242.

Mikolov T., Chen K., Corrado G. and Dean J. (2013). Efficient estimation of word representations in vector space. In *Proceedings of International Conference on Learning Representations - ICLR Workshop*.

https://doi.org/10.1017/S1351324919000342 Published online by Cambridge University Press

Miller G.A. (1995). WordNet: A Lexical database for english. *Communications of the ACM* **38**(11), 39–41.

Miwa M. and Bansal M. (2016). End-to-end relation extraction using lstms on sequences and tree structures. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*. Berlin: Association for Computational Linguistics, pp. 1105–1116.

Nguyen T. and Kan M. (2007). Keyphrase extraction in scientific publications. In *Proceedings of the 10th International Conference on Asian Digital Libraries*. Hanoi: Springer-Verlag Berlin, pp. 317–326.

Pagliardini M., Gupta P. and Jaggi M. (2017). Unsupervised Learning of Sentence Embeddings using Compositional n-Gram Features. In *arXiv:1703.02507v2*.

Papagiannopoulou E. and Tsoumakas G. (2018). Local Word Vectors Guiding Keyphrase Extraction. In *arXiv:1710.07503.*

Pennington J., Socher R. and Manning C.D. (2014). GloVe: global vectors for word representation. In *Proceedings of the Empirical Methods in Natural Language Processing (EMNLP)*. Brussels: Association for Computational Linguistics, pp. 1532–1543.

Pinheiro V., Furtado V., Pequeno T. and Nogueira D. (2010). Natural language processing based on semantiberc inferentialism for extracting crime information from text. In *IEEE international conference on intelligence and security informatics (ISI). IEEE (2010)*. Vancouver, BC, Canada, pp. 19–24.

Salton G. and McGill M. J. (1983) *Introduction to Modern Information Retrieval*. New York, NY: McGraw-Hill.

Saric F., Glavas G., Karan M., Snajder J. and Basic B.D. (2012). Takelab: systems for measuring semantic text similarity. In *First Joint Conference on Lexical and Computational Semantics*, Canada, pp. 441–448.

Silveira R., Furtado V. and Pinheiro V. (2015). Ranking keyphrases from semantic and syntactic features of textual terms. In *2015 Brazilian Conference on Intelligent Systems (BRACIS)*.

Sterckx L., Caragea C., Demeester T. and Develder C. (2016). Supervised keyphrase extraction as positive unlabeled learning. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. Austin, TX: Association for Computational Linguistics, pp. 1924–1929.

Tu Z., Lu Z., Liu Y., Liu X. and Li H. (2016). Modeling coverage for neural machine translation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*. Berlin: Association for Computational Linguistics, pp. 76–85.

Turney P.D. (2000) Learning algorithms for keyphrase extraction. *Information Retrieval* **2**(4), 303–336.

Wang L. and Li S. (2017). PKU-ICL at SemEval-2017 task 10: keyphrase extraction with model ensemble and external knowledge. In *Proceedings of the 11th International Workshop on Semantic Evaluations (SemEval-2017)*. Vancouver: Association for Computational Linguistics, pp. 934–937.

Wang R., Liu W. and McDonald C. (2015). Using Word Embeddings to Enhance Keyword Identification for Scientific Publications. In *Australasian Database Conference*, pp. 257–268.

Witten I.H., Paynter G.W., Frank E., Gutwin C. and Nevill-Manning C.G. (1999). KEA: a practical automatic keyphrase extraction. In *DL99 Proceedings of the fourth ACM Conference on Digital Libraries*.

Ye H. and Wang L. (2018). Semi-Supervised Learning for Neural Keyphrase Generation. In *arXiv:1808.06773*.

Zhang Q., Wang Y., Gong Y. and Huang X. (2016). Keyphrase extraction using deep recurrent neural networks on twitter. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, Austin, TX: Association for Computational Linguistics, pp. 836–845.

Zhang X., Zhao J. and Lecun Y. (2015). Character-level convolutional networks for text classification. In *Proceedings of the 28th International Conference on Neural Information Processing Systems (NIPS 2015)*. Montreal: MIT Press Cambridge, pp. 649–657.

Zhang Y., Li J., Song Y. and Zhang C. (2018). Enconding Conversation Contexto for Neural Keyphrase Extraction from Microblog Posts. In *Proceedings of Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT 2018)*. Minneapolis, MN: Association for Computational Linguistics, pp. 1676–1686.

Cite this article: Silveira R, Furtado V and Pinheiro V (2020). Learning keyphrases from corpora and knowledge models. *Natural Language Engineering* **26**, 293–318. https://doi.org/10.1017/S1351324919000342

https://doi.org/10.1017/S1351324919000342 Published online by Cambridge University Press