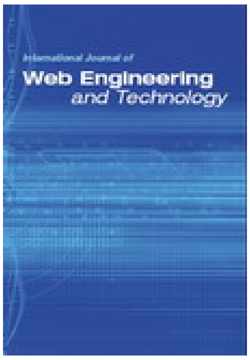
INDERSCIENCE PUBLISHERS
Linking academia, business and industry through research



International Journal of Web Engineering and Technology

ISSN online: 1741-9212 - ISSN print: 1476-1289
https://www.inderscience.com/ijwet

A novel machine extraction algorithm for implicit and explicit keywords based on dynamic web metadata of scientific scholars' corpus

Mawloud Mosbah

DOI: 10.1504/IJWET.2023.10056508

Article History:

Received:	17 March 2022
Last revised:	22 August 2022
Accepted:	08 January 2023
Published online:	31 May 2023

A novel machine extraction algorithm for implicit and explicit keywords based on dynamic web metadata of scientific scholars' corpus

Mawloud Mosbah

LRES Laboratory,
Informatics Department,
Faculty of Sciences,
University 20 Août 1955,
Skikda, Algeria
Email: mos_nasa@hotmail.fr

Abstract: Keywords extraction, as an operation to construct metadata, is an important pre-processing task considered by many natural language processing applications such as text summarisation, information retrieval, and clustering of documents. In this paper, we introduce a novel machine extraction algorithm for implicit and explicit keywords. The algorithm relies on a dynamic corpus of similar documents built by information retrieval engines. In addition to the direct utilisation of the keywords for similar documents, our algorithm combines some basic techniques. The given results, compared with some basic methods of the literature, seem to be very promising and we claim also the efficiency of our solution.

Keywords: natural language processing; keywords extraction; automatic construction of metadata; implicit keywords; explicit keywords.

Reference to this paper should be made as follows: Mosbah, M. (2023) 'A novel machine extraction algorithm for implicit and explicit keywords based on dynamic web metadata of scientific scholars' corpus', *Int. J. Web Engineering and Technology*, Vol. 18, No. 1, pp.29–44.

Biographical notes: Mawloud Mosbah is an Associate Professor at Informatics Department, University 20 Août 1955 of Skikda and member of LRES research laboratory within the same university. He has respectively received his Doctor of Sciences in informatics and his habilitation from University 20 Août 1955 in 2017 and 2022. His main areas of interest and expertise include information retrieval, image retrieval, and automatic natural language processing.

1 Introduction

Automatic processing of natural language (APNL) is becoming, increasingly, important regarding its huge number of applications in several sectors of our daily life. Automatic understanding of natural language, with its complex and unstructured character, is becoming then a necessity and a challenge for building intelligent machines and programs with strict control for a human language.

For a document, keywords are considered as a condensed version of its essential content. Moreover, they constitute a short and a concise form of its summary .They help then the simple user to decide, with a quick overview, whether or not a document is relevant. There are two kinds of keywords:

1 implicit keywords, with the involvement of an additional external knowledge

2 explicit keywords, extracted directly from a document, news, or a corpora.

Consulting of literature reveals two ways for identifying keywords from a document:

1 *keywords extraction*, with its discovering character

2 *keywords assignment*, with its matching aspect.

Keywords/phrases extraction is a task to identify some words, phrases and concepts representing the major topics included into a document, news, or a collection of documents. On the other hand, *keywords assignment* proceeds to choose, among a set of fixed controlled vocabulary and predefined taxonomy, some terms that match the meaning of a document or a corpus. Because of the hardness and the time-consuming to accomplish manually both operations, it is desirable and advisable to perform them automatically through statistics, linguistics, and machine learning. It is worthy to note that we are interested, here, in machine keywords extraction and not in keywords assignment.

For keywords extraction, the majority of publications use documents with its associated keywords, as training samples, to guess the desired keywords either using statistical features or through adopting advanced machine learning tools. Consulting of the literature reveals that there is a binary categorisation of keywords:

1 specialised keywords, dedicated specially for a document

2 global keywords, may be shared with similar documents.

Our contribution comes from the idea to consider firstly the global keywords may be shared explicitly with similar documents. Secondly, the considered global keywords are used to find other specialised keywords from the document in question with its both granularities: title and abstract, using statistical features and machine learning. The quality of our scheme is tied strictly then to the effectiveness of information retrieval systems may inject us with the appropriate similar documents.

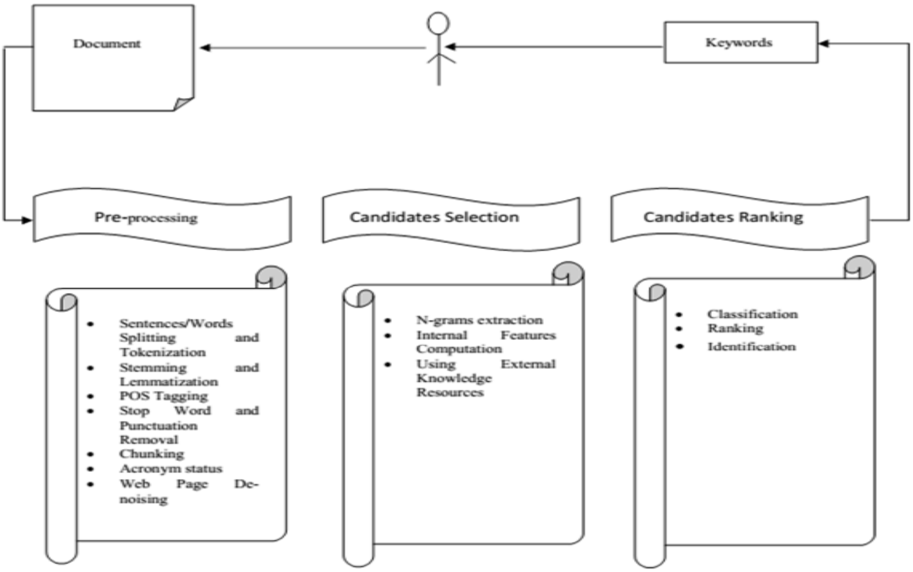
The major of works, for keywords extraction, addresses scientific documents and publications (Nguyen and Kan 2007; Wang et al., 2016; Beliga et al., 2017; Kim et al., 2013; Kovacevic et al., 2011; Anju et al., 2018). Even us, in this paper, we address scientific documents through a simple algorithm based on a corpus constructed dynamically from the web via some information retrieval engines.

The rest of the paper is structured as follows: in Section 2, we give related works dealing with keywords extraction. Section 3 presents our introduced algorithm. Some preliminary results are given in Section 4. Section 5 provides a conclusion and some perspectives.

2 Related works

Machine keywords/phrases extraction (Merrouni et al., 2020) consists of identifying, from a document, news, or a collection of documents, a subset of relevant words. It is then an attempt to understand text semantic, to characterise a document through constructing its semantic metadata, and to provide a rich set of information and concepts. Moreover, keywords extraction represents a first step around it many other natural language processing applications, such as: automatic text summarisation, information retrieval, query or topic generation, near-duplicate articles detection (Do and Ho, 2015), question-answering systems, and documents clustering and classification, are built. Extracted keywords, as a sequence of words known as n-grams or chunks, constitute an alternative or a complement for a document itself. Indeed, based on keywords, it is easy that a simple reader decide, with a quick overview, whether or not the document is relevant. Relevant keywords extracted should have then a close relation with the main topics, themes, or key ideas of the document. They should well describe, cover, and summarise concisely the entire document content or at least the main document essential content.

Figure 1 Keywords extraction stages



Owing to the increasable applications of natural language processing using keywords as an input, the review of literature reveals a large spectrum of works that deals with keywords extraction. The majority of works relies on noun-phrase chunks with four consecutive words known as 4-*grams*. As depicted in Figure 1, keywords extraction process is composed then of three different stages, namely: *pre-processing*, *candidates*' *selection*, and *candidates' ranking*. Pre-processing step is to prepare the document, the report, or the news to the steps coming later. Pre-processing tasks are distinguished then regarding the type of the resource being treated. Globally, these tasks are: *sentences and words splitting and tokenisation*, in order to extract the different tokens, *stemming and* *lemmatisation*, for designating stems and lems, POS *tagging*, *stop words and punctuation removal, chunking, acronym status, and web page de-noising* to discard presentation and non-informational tags. For *candidates' selection* step, there are commonly three tasks, namely: *N-grams extraction*, *internal features computation*, and the *use of external knowledge resources*. The ranking of the candidates, as a last step of the process, may be viewed as:

1 a classification problem

2 ranking issue

3 implemented using an identification operation.

As quoted in Hammouda et al. (2005), a key phrase is a sequence of one or more words that is considered highly relevant while a keyword is a single word that is potentially relevant. An arbitrary combination of keywords does not constitute a key phrase neither the constituents of a key phrase necessarily represent individual keywords.

According to what exists in the literature, many criteria may be adopted to categorise the published works for keywords extraction:

1 Selection of key phrases from candidates: where there are:

   a classification, as a supervised approach (Kovacevic et al., 2011; Ali and Omar, 2014; Zhang et al., 2020; Wang et al., 2006; Zhang et al., 2006; Li et al., 2008; Ecran and Cicekli, 2007; Wu et al., 2007)

   b ranking approach, based on certain attributed scores (Nguyen and Kan, 2007; Wang et al., 2016; Beliga et al., 2017; Kim et al., 2013; Anju et al., 2018; Hammouda et al., 2005; Li et al., 2007; Poulimenou et al., 2014; HaCohen-Kerner, 2003; Matsuo and Ishizuka, 2004; Campos et al., 2020; Palshikar, 2007; Florescu and Caragea, 2017; Wu et al., 2005; Witten et al., 2005; Kumar et al., 2016; Ventura and Silva, 2013; Beliga et al., 2016; Bracewell et al., 2005; Qin, 2012; Rousseau and Vazirgiannis, 2015; Chen et al., 2019; Pasquier, 2010; Liu et al., 2008; Li and Li, 2011; Sun et al., 2017; Xie and Hu, 2010; Li and Zhao, 2016; Pan et al., 2019; He et al., 2014; Shi et al., 2008; Wang et al., 2012; Vidal et al., 2012; Wang et al., 2013).

   Unfortunately, although its performance, supervised approach requires the availability of a training dataset.

2 The range applicability of the approach: There are: *single-language approach*, or linguistic approach, where the generalisation for other languages is not possible, and *N-languages approach* applied for no matter what the natural language. *Single-language approach* uses semantic relations, language specific syntactic tools such as *part of speech* (POS) taggers, *syntactic patterns* and s*temmers*, *linguistic properties*, *semantic rules* such as *synonyms* and *hyponyms* (Anju et al., 2018), and lexical chain (Li et al., 2008; Ecran and Cicekli, 2007). N-languages approach adopts data analysis and text mining through respectively simple basic statistics (Nguyen and Kan, 2007; Wang et al., 2016; Beliga et al., 2017; Kim et al., 2013; Anju et al., 2018; Hammouda et al., 2005; Li et al., 2007; Poulimenou et al., 2014; HaCohen-Kerner, 2003; Matsuo and Ishizuka, 2004; Campos et al., 2020; Palishkar, 2007; Beliga et al., 2016; Bracewell et al., 2005; Qin, 2012; Rousseau and Vazirgiannis, 2015; Chen et al., 2019; Pasquier, 2010) such as *term frequency* (TF),

TF/IDF, word *co-occurrence*, and unsupervised/supervised machine learning tools (Kovacevic et al., 2011; Ali and Omar, 2014; Turney, 2000; Florescu and Caragea, 2017; Kaur and Jain, 2017) such as maximum entropy (ME) method (Kim et al., 2013), conditional random fields (Anju et al., 2018), linear logistic regression (LLR) and linear discriminant analysis (LDA) (Ali and Omar, 2014), conditional probability (Wu et al., 2005; Witten et al. 2005), graph-based unsupervised n-gram filtration technique (Kumar et al., 2016; Garg, 2021), neural network (Zhang et al., 2020; Wang et al., 2006), support vector machine (Ali and Omar, 2014; Zhang et al., 2006), and least-square support vector machine (Wu et al., 2007).

3 The granularity level: The four granularity levels of the fragments considered for keywords extraction from a document are: title (Poulimenou, 2014; HaCohen-Kerner, 2003; Li and Li, 2011) or a sentence in the case of question-answering systems (Zhang et al. 2020), abstract (Beliga et al., 2017; HaCohen-Kerner, 2003), single document (Matsuo and Ishizuka, 2004; Campos et al., 2020; Palshikar, 2007; Beliga et al., 2016; Bracewell, 2005; Qin, 2012; Rousseau and Vasirgiannis, 2015; Chen et al., 2019; Pasquier, 2010; Liu et al., 2008; Xie and Hu, 2010; Wang et al., 2013; Turney, 2000), and corpus (Hammouda et al., 2005; Wu et al., 2007; Li et al., 2007; Wu et al., 2005; Witten et al., 2005; Beliga et al., 2016; Sun et al., 2017; Li and Zhao, 2016; He et al., 2014). The considered granularity level affects surely the effectiveness and the efficiency of the keywords extraction solution. Indeed, there is a trade on between the granularity level and the effectiveness. On the other hand, the more the granularity level is, the more the efficiency is not good.

4 Explicit vs. implicit keywords: Explicit keywords occur morphologically in a document whereas implicit keywords are attained using external knowledge resources (Ventura and Silva, 2013; Li and Li, 2011). In terms of difficulty, it is difficult to address implicit keywords where external semantic knowledge resources are required.

5 The features words dependency: there are some features that deal with a word regardless the others like *TF/IDF* whereas some features are calculated on the basis of the *words dependency* such as *co-occurrence statistical information* (Matsuo and Ishizuka, 2004; Xie and Hu, 2010), *linkage features* (Zhang et al., 2006), *semantic proximity* (Ventura and Silva, 2013), and *mutual information* (Liu et al., 2007; Li and Zhao, 2016; Pan et al., 2019). From semantic point of view, there is a dependency, between words in natural language, to be considered. Combining both kinds of features, with and without dependency, may, surely, give better results.

6 The considered resources: Some works consider just a documents in question (Matsuo and Ishizuka, 2004; Campos et al., 2020; Palshikar, 2007; Beliga et al., 2016; Bracewell, 2005; Qin, 2012; Rousseau and Vasirgiannis, 2015; Chen et al., 2019; Pasquier, 2010; Liu et al., 2008; Xie and Hu, 2010; Wang et al., 2013; Turney, 2000), some others use a corpus, some others consider thesaurus (He et al., 2014; Gazendam et al., 2010) such as HowNet (Li et al., 2008) and WordNet (Ecran and Cicekli, 2007), some ones use ontology (Do and Ho, 2015) whereas some others utilise Wikipedia (Li and Zhao, 2016; Shi et al. 2008; Wang et al., 2012; Vidal et al., 2012; Wang et al., 2013; Zhou et al., 2009). Although the efficiency aspect, the more considered resources are, the more the performance is good.

3 The introduced algorithm

The conception of our introduced algorithm, for keywords extraction, comes from the idea that the keywords, for a document, are of two kinds: global keywords that a document can share with other similar documents and specific or local keywords, related specially to some details of the document in question. For doing so, a corpus, including some similar documents, should firstly be gathered dynamically from the web via some information retrieval engines. In consequence, the accuracy of our keywords extraction solution is relative to the performance of the considered information retrieval engines. Moreover, every document, for which we desire extract keywords, has then its proper ad hoc corpus to build.

As depicted in Figure 2, the flow diagram of our proposed solution is as follows: after the user submits the document for which he/she desires to identify the keywords, the document is communicated by *user interaction module* to both modules: *pre-processing* and *web communication*. This latter proceeds to extract the title for sending it as a query to some scientific databases engines such as *Elsevier, Inderscience, Springer*, and *IEEE*. It is worthy to note that we avoid to use *Google Scholar* although its effectiveness for the simple reason that with the document title, as a query, *Google Scholar* answers only by the document itself. The asked engines answer by a set of documents considered then by the system, through *web communication module*, for building the related corpus. This corpus is communicated to the *pre-processing module* which begins to:

1 transform the format of documents from PDF to text

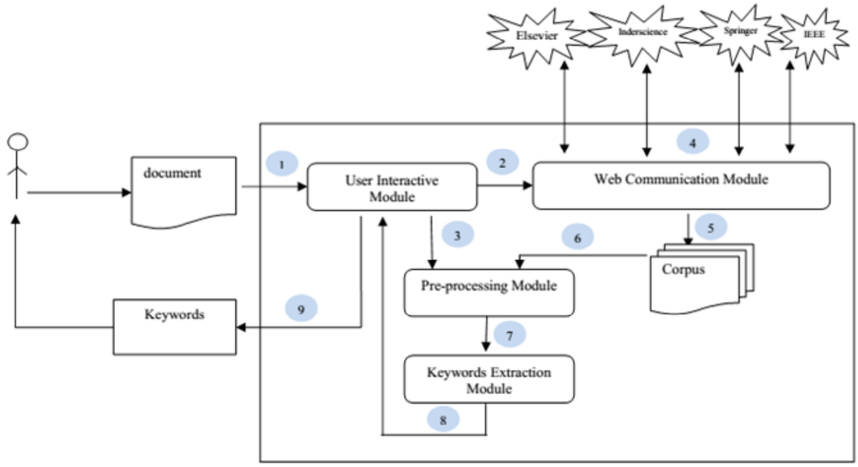
2 distinguish the keywords of all documents given by their authors

3 tokenise the documents abstracts into phrases and words as singular n-grams

4 consider only nouns as a part of speech.

The different author keywords, for the similar documents, and the extracted n-grams are sent to *keywords extraction module* that proceeds to calculate the various features of the different n-grams for extracting the keywords.

Figure 2 The flow diagram of our proposed solution (see online version for colours)



Our introduced algorithm, for keywords extraction, considers then some assumptions that are in convenience with some techniques from the literature. These assumptions are as follows:

1 As the documents of the corpus are similar to the document for which we desire to identify keywords (at least from the point of view of the engines), the common keywords shared by the documents, constituting the corpus, are considered then as global keywords for the document in question. The documents of the corpus that do not share the common keywords should be discarded from the corpus and they will not be taken into consideration in the future assumptions. The works that are in convenience with this assumption are: the clustering of sentences introduced in (Pasquier, 2010) and the probabilistic model given in (Coursey et al., 2008) whose the formula is:

$$p(keyword \setminus W) = \frac{count(D_{key})}{count(D_w)} \tag{1}$$

where

$p(keyword \setminus W)$ is the probability of a term W to be selected as a keyword in a new document.

$count(D_{key})$ is the number of documents where the term was already selected as a keyword.

$count(D_w)$ is the total number of documents where the term appeared.

2 The second level of global keywords, to be identified, is to consider the n-grams that appear in the kept documents of the corpus and co-occur with the pre-extracted keywords (extracted in the previous assumption). This assumption is in convenience with the *mutual information* reported in Siddiqi and Sharan (2015) and in accordance too with *inter-document correlation* (Ventura and Silva, 2013) whose the formulas are respectively:

$$M1(x, y) = log \frac{p(x, y)}{p(x)p(y)} \tag{2}$$

$$corr(A, B) = \frac{cov(A, B)}{\sqrt{cov(A, A)}\sqrt{cov(B, B)}} \tag{3}$$

where

$p(x, y)$ is the probability that the terms x and y appear together in a text.

$p(x)$ is the probability that the term x appears in a text.

$p(y)$ is the probability that the term y appears in a text.

$corr(A, B)$ is the correlation between terms A and B.

$cov(A, B)$ is the covariance between terms A and B.

3 The third level of keywords is to select the n-grams that appear in the document abstract and co-occur with the pre-designed keywords. This supposition is close to

the *co-occurrence statistical information* with x^2 with adaptation introduced in (Matsuo and Ishizuka 2004):

$$x^2(w) = \sum_{g \in G} \frac{\left(freq(w,g) - n_w P_g\right)^2}{n_w P_g} \tag{4}$$

where

freq(*w*, *g*) is the frequency of co-occurrence of term w and term g.

n_w is the total number of co-occurrences of term w and frequent terms G.

P_g is the unconditional probability of a frequent term g.

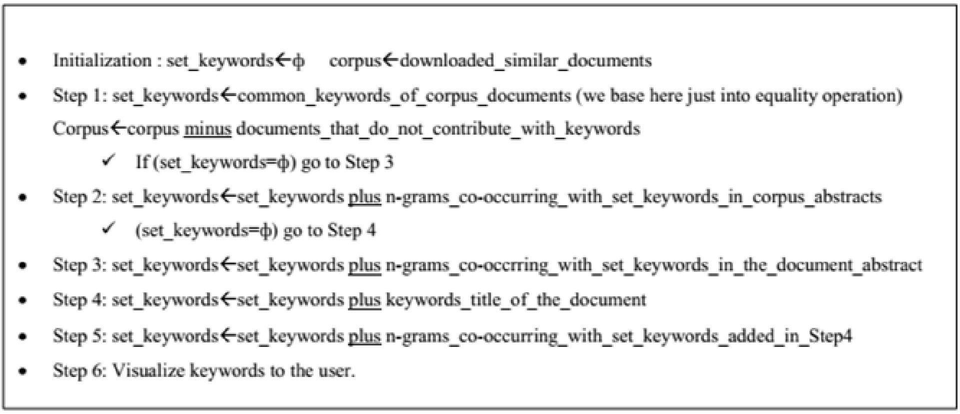
4 The fourth level of keywords, to be identified, as local or specific keywords, is to consider the n-grams that appear in the document title but do not occur as keywords for the documents of the corpus.

5 The fifth level of keywords is to choose the n-grams that well-correlate with the keywords, pre-designed in the assumption (4), into the abstract of the document. As in assumption (3), this supposition is close to the *co-occurrence statistical information* with x^2 with adaptation introduced in Matsuo and Ishizuka (2004).

The first two assumptions may generate implicit and global keywords that do not appear in the document in question. The last three assumptions produce explicit and specific keywords. In addition, although our proposal is based on a corpus, it does not require an important additional time as other techniques based-corpus does. Indeed, our solution uses firstly the available keywords of the similar documents without any deep processing. The processing is reserved only for identifying keywords from the title and the abstract of the document.

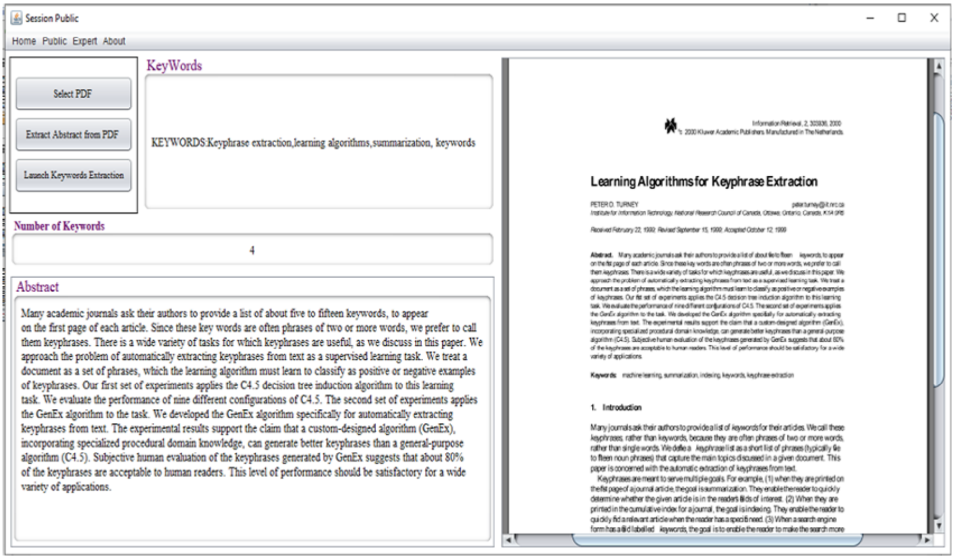
The pseudo code of our solution, according to the considered assumptions, is given in Figure 3.

Figure 3 The pseudo code of our proposed solution



```
• Initialization : set_keywords←ϕ     corpus←downloaded_similar_documents
• Step 1: set_keywords←common_keywords_of_corpus_documents (we base here just into equality operation)
  Corpus←corpus minus documents_that_do_not_contribute_with_keywords
      ✓ If (set_keywords=ϕ) go to Step 3
• Step 2: set_keywords←set_keywords plus n-grams_co-occurring_with_set_keywords_in_corpus_abstracts
      ✓ (set_keywords=ϕ) go to Step 4
• Step 3: set_keywords←set_keywords plus n-grams_co-occrring_with_set_keywords_in_the_document_abstract
• Step 4: set_keywords←set_keywords plus keywords_title_of_the_document
• Step 5: set_keywords←set_keywords plus n-grams_co-occurring_with_set_keywords_added_in_Step4
• Step 6: Visualize keywords to the user.
```

Figure 4 GUI of the implemented prototype



KEYWORDS:Keyphrase extraction,learning algorithms,summarization, keywords

4

Many academic journals ask their authors to provide a list of about five to fifteen keywords, to appear on the first page of each article. Since these key words are often phrases of two or more words, we prefer to call them keyphrases. There is a wide variety of tasks for which keyphrases are useful, as we discuss in this paper. We approach the problem of automatically extracting keyphrases from text as a supervised learning task. We treat a document as a set of phrases, which the learning algorithm must learn to classify as positive or negative examples of keyphrases. Our first set of experiments applies the C4.5 decision tree induction algorithm to this learning task. We evaluate the performance of nine different configurations of C4.5. The second set of experiments applies the GenEx algorithm to the task. We developed the GenEx algorithm specifically for automatically extracting keyphrases from text. The experimental results support the claim that a custom-designed algorithm (GenEx), incorporating specialized procedural domain knowledge, can generate better keyphrases than a general-purpose algorithm (C4.5). Subjective human evaluation of the keyphrases generated by GenEx suggests that about 80% of the keyphrases are acceptable to human readers. This level of performance should be satisfactory for a wide variety of applications.

Information Retrieval, 2, 303–336, 2000
© 2000 Kluwer Academic Publishers. Manufactured in The Netherlands.

Learning Algorithms for Keyphrase Extraction

PETER D. TURNEY peter.turney@iit.nrc.ca
Institute for Information Technology, National Research Council of Canada, Ottawa, Ontario, Canada, K1A 0R6

Received February 22, 1999; Revised September 15, 1999; Accepted October 12, 1999

Abstract. Many academic journals ask their authors to provide a list of about five to fifteen *keywords*, to appear on the first page of each article. Since these key words are often phrases of two or more words, we prefer to call them *keyphrases*. There is a wide variety of tasks for which keyphrases are useful, as we discuss in this paper. We approach the problem of automatically extracting keyphrases from text as a supervised learning task. We treat a document as a set of phrases, which the learning algorithm must learn to classify as positive or negative examples of keyphrases. Our first set of experiments applies the C4.5 decision tree induction algorithm to this learning task. We evaluate the performance of nine different configurations of C4.5. The second set of experiments applies the *GenEx* algorithm to the task. We developed the GenEx algorithm specifically for automatically extracting keyphrases from text. The experimental results support the claim that a custom-designed algorithm (GenEx), incorporating specialized procedural domain knowledge, can generate better keyphrases than a general-purpose algorithm (C4.5). Subjective human evaluation of the keyphrases generated by GenEx suggests that about 80% of the keyphrases are acceptable to human readers. This level of performance should be satisfactory for a wide variety of applications.

Keywords: machine learning, summarization, indexing, keywords, keyphrase extraction

1. Introduction

Many journals ask their authors to provide a list of *keywords* for their articles. We call these *keyphrases*, rather than keywords, because they are often phrases of two or more words, rather than single words. We define a *keyphrase list* as a short list of phrases (typically five to fifteen noun phrases) that capture the main topics discussed in a given document. This paper is concerned with the automatic extraction of keyphrases from text.

Keyphrases are meant to serve multiple goals. For example, (1) when they are printed on the first page of a journal article, the goal is summarization. They enable the reader to quickly determine whether the given article is in the reader's fields of interest. (2) When they are printed in the cumulative index for a journal, the goal is indexing. They enable the reader to quickly find a relevant article when the reader has a specific need. (3) When a search engine form has a field labelled *keywords*, the goal is to enable the reader to make the search more

4 Experimental results and discussion

As depicted in Figure 4, we have implemented a Java prototype to test our proposed solution. We have used *Netbeans* as an editor and two APIs s: PDFBox.jar to transform format from PDF to text and *Icepdf-viewer.jar* to visualise PDF document. Unfortunately, there is no a *Java* API to identify the POS of the words. For this purpose, we employ *TreeTagger* software whose trial version is available freely on the web (https://www.cis.uni-muenchen.de/~schmid/tools/TreeTagger/). As an evaluation task, we have experimented our solution, as well as some basic methods from the literature, on our local benchmark of 500 English papers (100 papers + 400 downloaded papers as a dynamic corpus) with 10 topics (10 papers + 40 downloaded papers for each topic) namely: *query expansion, information retrieval, object design, software maintenance, validation and verification, functional programming, remote object access, code performance optimisation, object modelling with UML*, and *business component resource*. As depicted in Table 1, we consider, for all the experimented techniques, 5 and 10 returned keywords. We use the average of the average for the three considered assessment metrics namely: *precision*, *recall*, and *f-measure* (Firoozeh et al., 2020) whose the equations are given respectively as follows:

$$precision = \frac{Retrieved \bigcap Relevant}{Retrieved} \tag{5}$$

$$recall = \frac{Retrieved \bigcap Relevant}{Relevant} \tag{6}$$

$$Fmeasur = (1+\beta^2).\frac{precision.recall}{\left(\beta^2.precision\right)+recall} \tag{7}$$

where

Retrieved designates the set of keywords returned by the system.

Relevant constitutes the set of keywords qualified as correct keywords.

The results are presented in Table 1 whereas an illustrative example, for the average of the three evaluation metrics where the topic is 'query expansion', is shown in Figure 5.

Table 1 The performance of the proposed algorithm vs. some basic works of the literature

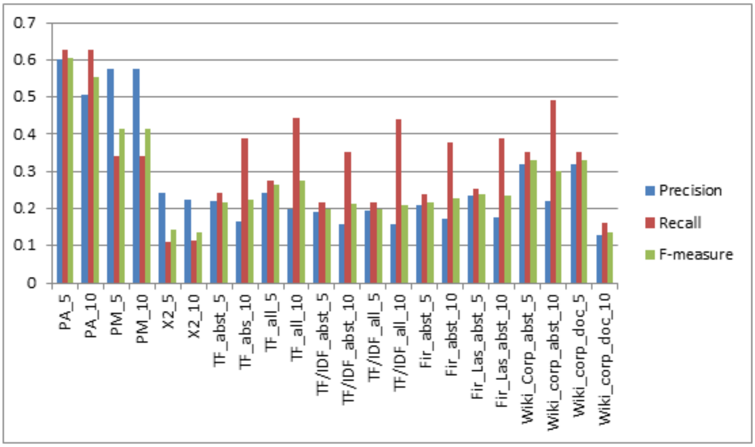
Methods	*Average of the average precision*	*Average of the average recall*	*Average of the average F-measure*	*Average number of returned keywords*	*Average number of author document keywords*
The proposed algorithm	*0.55548*	*0.56835*	*0.55164*	5	4.83
	0.42884	*0.57086*	*0.45226*	10	
Simple probabilistic model	0.401	0.202	0.254	5	
	0.397	0.202	0.252	10	
X^2	0.24163	0.10892	0.14228	5	
	0.2248	0.11529	0.13754	10	
TF model for abstract	0.222	0.244	0.217	5	
	0.166	0.389	0.223	10	
TF model for the document at all	0.243	0.275	0.266	5	
	0.199	0.444	0.276	10	
TF/IDF model for abstract	0.192	0.216	0.199	5	
	0.158	0.351	0.212	10	
TF/IDF model for the document at all	0.194	0.216	0.2	5	
	0.158	0.439	0.208	10	
First occurrence model for abstract	0.211	0.238	0.217	5	
	0.172	0.378	0.229	10	
First occurrence-last occurrence model for abstract	0.234	0.255	0.237	5	
	0.178	0.388	0.236	10	
WikiFier-corpus_abstract	0.308	0.31366	0.29473	5	
	0.175	0.34712	0.22409	10	
WikiFier_corpus_document	0.296	0.32292	0.30087	5	
	0.173	0.34362	0.21978	10	

According to Table 1, our proposed algorithm outperforms the considered basic techniques over the three considered performance metrics namely: precision, recall, and f-measure. The outperformance is evident especially with five returned keywords. Additionally, Table 1 shows the fact that probabilistic model comes in the second position in terms of precision while TF applied on the entire document with ten returned keywords is ranked second regarding f-measure and recall. Moreover, precision values are better with five returned keywords while recall is good with ten returned keywords for all the considered techniques. From granularity view point, the effectiveness of the considered techniques on the entire document is relatively better than the case where they are applied just on abstract (Wikifier model and the *F-measure* of *TF/IDF* with ten returned keywords are some exceptions).

As an example among the ten considered topics, Figure 5 shows the results where the topic is 'query expansion'. As the global evaluation presented in Table 4, Figure 5 sustains the fact that the proposed algorithm outperforms the other considered techniques.

We claim that our solution, belonging to the corpus-based approach, has an online aspect as an advantage compared with the other techniques of the same approach or of the other alternatives based on external resources. Moreover, the proposed algorithm seems to be more efficient for the reason that it is based partially on global available keywords shared by the similar documents of the dynamic corpus. In addition, the second part of the algorithm, relying on title and abstracts, is qualified also efficient compared to the techniques based on the entire document.

Figure 5 Average precision, average recall and average f-measure of the considered keywords extraction models where the topic is 'query expansion' (see online version for colours)



where

*PA*_5 the proposed algorithm taking into consideration the five first returned keywords.

*PA*_10 the proposed algorithm taking into consideration the ten first returned keywords.

*PM*_5 the probabilistic model taking into consideration the five first returned keywords.

*PM*_10 the probabilistic model taking into consideration the ten first returned keywords.

*X*2_5 the X^2 model taking into consideration the five first returned keywords.

*X*2_10 the X^2 model taking into consideration the ten first returned keywords.

*TF_abst*_5 TF Model applied into abstract taking into consideration the five first returned keywords.

*TF_abst*_10 TF model applied into abstract taking into consideration the ten first returned keywords.

*TF_all*_5 TF model applied into the entire document taking into consideration the five first returned keywords.

*TF_all*_10 TF model applied into the entire document taking into consideration the ten first returned keywords.

*TF/IDF_abst*_5 TF/IDF model applied into the abstract taking into consideration the five first returned keywords.

*TF/IDF_abst*_10 TF/IDF model applied into the abstract taking into consideration the ten first returned keywords.

*TF/IDF_all*_5 TF/IDF model applied into the entire document taking into consideration the five first returned keywords.

*TF/IDF_all*_10 TF/IDF model applied into the entire document taking into consideration the ten first returned keywords.

*Fir_abst*_5 first occurrence model applied into the abstract taking into consideration the five first returned keywords.

*Fir_abst*_10 first occurrence model applied into the abstract taking into consideration the ten first returned keywords.

*Fir_Las*_*abst*_5 first-last occurrence model applied into the abstract taking into consideration the five first returned keywords.

*Fir_Las_abst*_10 first-last occurrence model applied into the abstract taking into consideration the ten first returned keywords.

*Wiki_corp_abstr*_5 Wikifier model based on corpus applied into the abstract taking into consideration the five first returned keywords.

*Wiki_corp_abstr*_10 Wikifier model based on corpus applied into the abstract taking into consideration the ten first returned keywords.

*Wiki_corp_doc*_5 Wikifier model based on corpus applied into the entire document taking into consideration the five first returned keywords.

*Wiki_corp_doc*_10 Wikifier model based on corpus applied into the entire document taking into consideration the ten first returned keywords.

As an illustrative example, Table 2 presents respectively author keywords and those returned by our prototype, of one document, for a 'query expansion' topic. As we see from the table, our solution gives relevant keywords, not taken into account in the evaluation process for the reason that they are different, in terms of vocabulary, with those given by the author of the document. In consequence, the effectiveness of our solution may be better whether we consider semantic comparison, in evaluation process, rather than simple vocabulary comparison.

Table 2 Set of keywords given by the author vs. set of keywords returned by our prototype for a document of 'query expansion' topic

Set of keywords given by the author	*Set of keywords returned by our prototype using our proposed solution*
Query expansion	Query expansion
Log mining	Information retrieval
Probabilistic model	Probabilistic method
Information retrieval	Query Log and probabilistic query expansion
Search engine	Search performance

5 Conclusions

In this paper, we have introduced a new solution for keywords extraction based on a corpus built dynamically from the web. The proposed solution, based on global and local keywords, combines some ideas from basic techniques of the literature. Compared with what exists as keywords extraction techniques, the returned experimental results, on our proper English benchmark, seem to be very promising. We claim that our fully online proposed solution, partially based on some available keywords of similar documents, is efficient compared with other techniques especially those based on the entire document. In addition, a valuable background, exploring the various aspects of keywords extraction, has been presented and an experimental comparison of some basic techniques of the literature, over different granularity levels, has been achieved. We claim that this work open a new research avenue that of the well selection for the suitable corpus. Indeed, the well selection of the suitable corpus may improve the effectiveness of the keywords extraction system no matter what the considered machine extraction technique. As a perspective, we will prove, with experimental results, the importance of corpus-selection in the performance of keywords extraction techniques.

References

Ali, N.G. and Omar, N. (2014) 'Arabic keyphrases extraction using a hybrid of statistical and machine learning methods', in *Proceedings of the 6th International Conference on Information Technology and Multimedia*, IEEE, November, pp.281–286.

Anju, R.C., Ramesh, S.H. and Rafeeque, P.C. (2018) 'Keyphrase and relation extraction from scientific publications', in *Advances in Machine Learning and Data Science*, pp.113–120, Springer, Singapore.

Beliga, S., Kitanovic, O., Stankovic, R. and Martincic-lpsic, S. (2017) 'Keyword extraction from parallel abstracts of scientific publications', in *Semantic Keyword-Based Search on Structured Data Sources*, pp.44–55, Springer, Cham, September.

Beliga, S., Mestrovic, A. and Martincic-Ipsic, S. (2016) 'Selectivity-based keyword extraction method', *International Journal on Semantic Web and Information Systems (IJSWIS)*, Vol. 12, No. 3, pp.1–26.

Bracewell, D.B., Ren, F. and Kuriowa, S. (2005) 'Multilingual single document keyword extraction for information retrieval', in *2005 International Conference on Natural Language Processing and Knowledge Engineering, IEEE*, October, pp.517–522.

Campos, R., Mangaravite, V., Pasquali, A., Jorge, A., Nunes, C. and Jatowt, A. (2020) 'YAKE ! keyword extraction from single documents using multiple local features', *Information Sciences*, Vol. 509, No. 1, pp.257–289.

Chen, Y., Wang, J., Li, P. and Guo, P. (2019) 'Single document keyword extraction via quantifying higher-order structural features of word co-occurrence graph', *Computer Speech and Language*, Vol. 57, No. 1, pp.98–107.

Coursey, K.H., Mihalcea, R. and Moen, W.E. (2008) 'Automatic keyword extraction for learning object repositories', *Proceedings of the American Society for Information Science and Technology*, Vol. 45, No. 1, pp.1–10.

Do, N. and Ho, L. (2015) 'Domain-specific keyphrase extraction and near-duplicate article detection based on ontology', in the *2015 IEEE RIVF International Conference on Computing and Communication Technologies-Research, Innovation, and Vision for Future (RIVF)*, IEEE, January, pp.123–126.

Ecran, G. and Cicekli, I. (2007) 'Using lexical chains for keywords extraction', *Information Processing and Management*, Vol. 43, No. 6, pp.1705–1714.

Firoozeh, N., Nazarenko, A., Alizon, F. and Daille, B. (2020) 'Keyword extraction: issues and methods', *Natural Language Engineering*, Vol. 26, No. 3, pp.259–291.

Florescu, C. and Caragea, C. (2017) 'A new scheme for scoring phrases in unsupervised keyphrase extraction', in *European Conference on Information Retrieval*, Springer, Cham, April, pp.477–483.

Garg, M. (2021) 'A survey on different dimensions for graphical keyword extraction techniques', *Artificial Intelligence Review*, Vol. 54, No. 6, pp.4731–4770.

Gazendam, L., Wartena, C. and Brussee, R. (2010) 'Thesaurus based term ranking for keyword extraction', in *2010 Workshops on Database and Expert Systems Applications*, IEEE, August, pp.49–53.

HaCohen-Kerner, Y. (2003) 'Automatic extraction of keywords from abstracts', in *International Conference on Knowledge-Based and Intelligent Information and Engineering Systems*, Springer, Berlin, Heidelberg, September, pp.843–849.

Hammouda, K.M., Matute, D.N. and Kamel, M.S. (2005) 'Corephase: keyphrase extraction for document clustering', in *International Workshop on Machine Learning and Data Mining in Pattern Recognition,* Springer, Berlin, Heidelberg, July, pp.265–274.

He, G., Wang, J., Zhang, Y. and Peng, Y. (2014) 'Keyword extraction of web pages based on domain thesaurus', in *2014 IEEE 3rd International Conference on Cloud Computing and Intelligence Systems, IEEE*, November, pp.310–314.

Kaur, B. and Jain, S. (2017) 'Keyword extraction using machine learning approaches', in *2017 3rd International Conference on Advances in Computing, Communication and Automation (ICACCA) (Fall),* IEEE, September, pp.1–6.

Kim, S.N., Medelyan, O., Kan, M.Y. and Baldwin, T. (2013) 'Automatic keyphrase extraction from scientific articles', *Language Resources and Evaluation*, Vol. 47, No. 3, pp.723–742.

Kovacevic, A., Ivanovic, D., Milosavljevic, B., Konjovic, Z. and Surla, D. (2011) 'Automatic extraction of metadata from scientific publications for CRIS systems', *Program*, Vol. 45, No. 4, pp.376–396.

Kumar, N., Srinathan, K. and Varma, V. (2016) 'A graph-based unsupervised N-gram filtration technique for automatic keyphrase extraction', *International Journal of Data Mining, Modelling and Management,* Vol. 8, No. 2, pp.124–143.

Li, D. and Li, S. (2011) 'Hypergraph-based inductive learning for generating implicit key phrases', in *Proceedings of the 20th International Conference Companion on World Wide Web*, March, pp.77–78.

Li, J., Fan, Q.N. and Zhang, K. (2007) 'Keyword extraction based on tf/idf for chinese news document', *Wuhan University Journal of Natural Sciences*, Vol. 12, No. 5, pp.917–921.

Li, W. and Zhao, J. (2016) 'TextRank algorithm by exploiting Wikipedia for short text keywords extraction', in *2016 3rd International Conference on Information Science and Control Engineering (ICISCE)*, IEEE, July, pp.683–686.

Li, X., Wu, X., Hu, X., Xie, F. and Jiang, Z. (2008) 'Keyword extraction based on lexical chains and word co-occurrence for Chinese news web pages', in *2008 IEEE International Conference on Data Mining Workshps, IEEE*, December, pp.744–751.

Liu, F., Liu, F. and Liu, Y. (2008) 'Automatic keyword extraction for the meeting corpus using supervised approach and bigram expansion', in *2008 IEEE Spoken Language Technology Workshop, IEEE*, December, pp.181–184.

Liu, L., He, G., Shi, X. and Song, H. (2007) 'Metadata extraction based on mutual information in digital libraries', in *2007 First IEEE International Symposium on Information Technologies and Applications in Education*, IEEE, November, pp.209–212.

Matsuo, Y. and Ishizuka, M. (2004) 'Keyword extraction from a single document using word co-occurrence statistical information', *International Journal on Artificial Intelligence Tools*, Vol. 13, No. 1, pp.157–169.

Merrouni, Z.A., Frikh, B. and Ouhbi, B. (2020) 'Automatic keyphrase extraction: a survey and trends', *Journal of Intelligent Information Systems*, Vol. 54, No. 2, pp.391–424.

Nguyen, T.D. and Kan, M.Y. (2007) 'Keyphrase extraction in scientific publications', in *International Conference on Asian Digital Libraries*, Springer, Berlin, Heidelberg, December, pp.317–326.

Palshikar, G.K. (2007) 'Keyword extraction from a single document using centrality measures', in *International Conference on Pattern Recognition and Machine Intelligence*, Springer, Berlin, Heidelberg, December, pp.503–510.

Pan, S., Li, Z. and Dai, J. (2019) 'An improved TextRank keywords extraction algorithm', in *Proceedings of the ACM Turing Celebration Conference-China*, May, pp.1–7.

Pasquier, C. (2010) 'Single document keyphrase extraction using sentence clustering and latent dirichlet allocation', in *Proceedings of the 5th International Workshop on Semantic Evaluation*, July, pp.154–157.

Poulimenou, S., Stamou, S., Papavlasopoulos, S. and Poulos, M. (2014) 'Keywords extraction from articles' title for ontological purposes', in *Proceedings of the 2014 International Conference on Pure Mathematics, Applied Mathematics, Computational Methods (PMAMCM 2014)*, pp.120–125.

Qin, Y. (2012) 'Applying frequency and location information to keyword extraction in single document', in *2012 IEEE 2nd International Conference on Cloud Computing and Intelligence Systems*, IEEE, pp.1398–1402.

Rousseau, F. and Vazirgiannis, M. (2015) 'Main core retention on graph of words for single-document keyword extraction', in *European Conference on Information Retrieval*, Springer, Cham, March, pp.382–393.

Shi, T., Jiao, S., Hou, J. and Li, M. (2008) 'Improving keyphrase extraction using wikepedia semantics', in *2008 Second International Symposium on Intelligent Information Technology Application*, IEEE, December, pp.42–46.

Siddiqi, S. and Sharan, A. (2015) 'Keyword and keyphrase extraction techniques: a literature review', *International Journal of Computer Applications*, Vol. 109, No. 2, pp.18–23.

Sun, P., Wang, L. and Xia, Q. (2017) 'The keyword extraction of Chinese medical web page based on WF-TF-IDF algorithm', in *2017 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC)*, IEEE, October, pp.193–198.

Turney, P.D. (2000) 'Learning algorithms for keyphrase extraction', *Information Retrieval,* Vol. 2, No. 4, pp.303–336.

Ventura, J. and Silva, J. (2013) 'Automatic extraction of explicit and implicit keywords to build document descriptors', in *Portuguese Conference on Artificial Intelligence*, Springer, Berlin, Heidelberg, September, pp.492–503.

Vidal, M., Menezes, G.V., Berlt, K., de Moura, E.S., Okada, K., Ziviani, N., … and Cristo, M. (2012) 'Selecting keywords to represent web pages using wikipedia information', in *Proceedings of the 18th Brazilian Symposium on Multimedia and the Web*, October, pp.375–382.

Wang, J., Peng, H. and Hu, J.S. (2006) 'Automatic keyphrases extraction from document using neural network', in *Advances in Machine Learning and Cybernetics*, Springer, Berlin, Heidelberg, pp.633–641.

Wang, M., Zhao, B. and Huang, Y. (2016) 'PTR: phrase-based topical ranking for automatic keyphrase extraction in scientific publications', in *International Conference on Neural Information Processing*, Springer, Cham, October, pp.120–128.

Wang, X., Li, H., Jia, Y. and Jin, S. (2013) 'Chinese text filtering based on domain keywords extracted from Wikipedia', in *Proceedings of the 2012 International Conference on Information Technology and Software Engineering*, Springer, Berlin, Heidelberg, pp.991–1000.

Wang, X., Wang, L., Li, J. and Li, S. (2012) 'Exploring simultaneous keyword and key sentence extraction: improve graph-based ranking using Wikipedia', in *Proceedings of the 21st ACM International Conference on Information and Knowledge Management*, October, pp.2619–2622.

Witten, I.H., Paynter, G.W., Frank, E., Gutwin, C. and Nevill-Manning, C.G. (2005) 'Kea: practical automated keyphrase extraction', in *Design and Usability of Digital Libraries: Case Studies in the Asia Pacific*, pp.129–152, IGI global.

Wu, C., Marchese, M., Jiang, J., Ivanyukovich, A. and Liang, Y. (2007) 'Machine learning-based keywords extraction for scientific literature', *J. Univers. Comput. Sci.*, Vol. 13, No. 10, pp.1471–1483.

Wu, Y.F.B., Li, Q., Bot, R.S. and Chen, X. (2005) 'Domain-specific keyphrase extraction', in *Proceedings of the 14th ACM International Conference on Information and Knowledge Management*, October, pp.283–284.

Xie, F.X. and Hu, X. (2010) 'Keyphrase extraction based on semantic relatedness', in *9th IEEE International Conference on Cognitive Informatics (ICCI'10)*, IEEE, July, pp.308–312.

Zhang, K., Xu, H., Tang, J., and Li, J. (2006). 'Keyword extraction using support vector machine'. In International Conference on Web-Age Information Management, Springer, Berlin, Heidelberg, June, pp.85-96.

Zhang, Y., Tuo, M., Yin, Q., Qi, L., Wang, X. and Liu, T. (2020) 'Keywords extraction with deep neural network model', *Neurocomputing*, Vol. 383, No. 1, pp.113–121.

Zhou, B., Luo, P., Xiong, Y. and Liu, W. (2009) 'Wikipedia-graph based key concept extraction towards news analysis', in *2009 IEEE Conference on Commerce and Enterprise Computing, IEEE,* July, pp.121–129.

Websites

https://www.cis.uni-muenchen.de/~schmid/tools/TreeTagger/ (accessed on 10 July 2021).