Natural Language Engineering (2019), 1–33
doi:10.1017/S1351324919000457

CAMBRIDGE
UNIVERSITY PRESS

SURVEY PAPER

Keyword extraction: Issues and methods

Nazanin Firoozeh[1,2*], Adeline Nazarenko[2], Fabrice Alizon[1] and Béatrice Daille[3]

[1]Pixalione SAS, 75015 Paris, France, [2]Northern Paris Computer Science Laboratory (LIPN), Paris 13 University – Sorbonne Paris Cité & CNRS, 93430 Villetaneuse, France and [3]Laboratory of Digital Sciences of Nantes (LS2N), University of Nantes, 44322 Nantes Cedex 3, France
*Corresponding author. Email: nazanin.firoozeh@pixalione.com

(Received 24 March 2017; revised 26 July 2019; accepted 28 July 2019)

Abstract

Due to the considerable growth of the volume of text documents on the Internet and in digital libraries, manual analysis of these documents is no longer feasible. Having efficient approaches to keyword extraction in order to retrieve the 'key' elements of the studied documents is now a necessity. Keyword extraction has been an active research field for many years, covering various applications in Text Mining, Information Retrieval, and Natural Language Processing, and meeting different requirements. However, it is not a unified domain of research. In spite of the existence of many approaches in the field, there is no single approach that effectively extracts keywords from different data sources. This shows the importance of having a comprehensive review, which discusses the complexity of the task and categorizes the main approaches of the field based on the features and methods of extraction that they use. This paper presents a general introduction to the field of keyword/keyphrase extraction. Unlike the existing surveys, different aspects of the problem along with the main challenges in the field are discussed. This mainly includes the unclear definition of 'keyness', complexities of targeting proper features for capturing desired keyness properties and selecting efficient extraction methods, and also the evaluation issues. By classifying a broad range of state-of-the-art approaches and analysing the benefits and drawbacks of different features and methods, we provide a clearer picture of them. This review is intended to help readers find their way around all the works related to keyword extraction and guide them in choosing or designing a method that is appropriate for the application they are targeting.

Keywords: Information extraction; Keyword extraction; Extraction features; Extraction methods

1. Introduction

Keywords, which are important phrases within documents (Turney 2000), play an important role in different applications of Text Mining, Information Retrieval (IR), and Natural Language Processing (NLP). With the growth in the quantity of available documents, it is no longer possible for a user to read them all in detail. Hence, knowing about the subject of the documents without analysing them in depth is essential and having an automatic approach to keyword extraction is a necessity.

The task of keyword extraction can be defined as the identification of the lexical units that best represent the document. However, automatically extracting keywords is challenging due to the complexities of natural language, heterogeneity in the type of input documents and the type of keywords that need to be extracted. After years of active research and development, numerous methods and tools have been designed. They have been tested on various kinds of input data, including long scientific articles, abstracts, and web pages. However, no approach really emerges as the dominant or standard one and it is difficult for newcomers to select the approach that best fits their problem, input data, and application. A comprehensive analysis of these works is needed to highlight not only the problems and the task complexity but also the diversity of approaches proposed over the years, their underlying logic, their strengths, and their limitations.

Several reviews on keyword extraction have been published during the past years, but they are rather brief in their analysis. They often tackle a specific aspect of the problem: Hussey, Williams, and Mitchell (2012) focus on the relative performance of different extraction features, while Hasan and Ng (2014) study the errors made by the keyphrase extractors in the literature. The papers that give an overview of extraction methods present only brief descriptions and remain rather technical (Kaur and Gupta 2010; Siddiqi and Sharan 2015).

The present paper is meant as a general and comprehensive introduction to the abounding field of keyword or keyphrase extraction. It focuses on metadata and document summarization applications but is not bound to any specific type of document nor advocates in favour of any specific approach. Considering the general issue of extracting key elements from unstructured documents with content expressed in natural language, it presents and discusses the various solutions that have been proposed over the years.

The paper is organized as follows. Since the notion of keyword is difficult to capture and is used in very different contexts, we first present two main applications that rely on keyword extraction (Section 2) and the types of lexical units and semantic properties that are often targeted (Section 3). Before presenting the extraction approaches *per se*, we also address evaluation issues (methodologies, measures, and benchmarks, Section 4). In the presentation of the extraction approaches, we distinguish the features that are used to assess the 'keyness' of lexical units (Section 5) and the methods or algorithms that exploit these features (Section 6). Finally, we discuss the challenges and limitations of evaluating the different approaches (Section 7) and present a conclusion (Section 8).

2. Applications

Many NLP applications are required to extract 'key' words and phrases from unstructured textual data. It is interesting to understand how keywords can be exploited once they have been extracted and which types of keywords are targeted.

Metadata enrichment Keywords are often used as metadata to enrich documents. They can be directly extracted from the source document to emphasize its most important elements or derived from a larger corpus to bring in contextual information. The metadata gives an explicit and computer-processable description of the text content that plays a central role in content management tasks (browsing, indexing, topic detection, classification) and semantic-aware applications (exploratory search, recommendation, reputation analysis, or contextual advertising). An example is given by Mori *et al.* (2004).

Document summarization Automatic summarization is often based on extraction: the key elements (words, phrases, sentences) extracted from a source document are assembled to form a target document much shorter than the source one. As stated by Bharti and Babu (2017), keyword extraction can be regarded as the first step of document summarization. Wan, Yang, and Xiao (2007) also show that keyword and keyphrse extraction tools are core elements in extraction-based summarization, key sentences being those that contain more, and the most significant, key words and phrases. Boudin and Morin (2013) also exploit keyphrases for generating a single sentence summary from a set of related sentences.

A wide diversity of textual sources are exploited for extraction: a single document or a document collection; coherent documents, which discuss a single domain, versus a collection of heterogeneous documents in terms of their domains; long versus short documents; traditional well-written documents or informal ones; mono- or multi-authored documents; and scientific, technical, legal, or commercial documents. Some authors focus on a specific type of document:

Table 1. Statistics on the length of the keywords in SemEval 2010 and Duc

	SemEval 2010 (%)	Duc (%)
Single token	20.2	17.1
Two tokens	53.4	60.8
Three tokens	21.3	17.8

Matsuo and Ishizuka (2003) work on academic papers, whereas others focus on shorter content, such as tweets (Zhang *et al.* 2016) or web pages (Kelleher and Luz 2005; Yih, Goodman, and Carvalho 2006). Some works, however, target several types of documents: Turney (2000) extracts keywords from articles, email addresses, and web pages; and Sterckx *et al.* (2016) target news and magazine articles and also abstracts of academic papers.

This paper focuses on keyword extraction approaches. We put no restriction on the type of documents that are processed or on their language. We consider any type and size of source text, taking into consideration methods designed for processing academic papers as well as web pages. We consider 'unstructured data', that is documents with content mainly expressed in plain natural language, regardless of their structuration into paragraphs, sections, or chapters.

3. From words to keywords

One major source of complexity in the domain of keyword extraction is related to the diversity of the targeted elements. The extraction methods are aimed at extracting 'key elements', which refer to 'important' textual units. However, there are various ways to assess the importance of those elements and various types of units can be targeted, from tokens to *N-Grams* or from words to phrases. Since not all the methods are equally suited to all types of elements, it is important to specify the target elements to determine how to extract them.

3.1 Definition

The terms 'keyword' and 'keyphrase' do not refer to any theory. An element is considered as a 'key' element with respect to a document, when it is an important descriptor of the document content. The opposition word versus phrase simply refers to the mono- or multi-lexical structure of the textual units, which can be composed of one or several tokens. However, the formal word/phrase distinction is often blurred.

In this paper, the terms 'keyphrase' and 'keyword' are used interchangeably and they do not refer to any linguistic or semantic theory. In other words, 'keyword' stands for any key textual unit that can be composed of one or more words and may work as either a common or proper noun.

Keywords can be different lengths. In the literature, 'length' is defined either as the number of the constituent words or the number of the characters in a keyword. A wide variety of length can be observed in the literature on keyword extraction: single token and two tokens, respectively in the approaches proposed by Liu *et al.* (2009) and Muñoz (1997), up to three tokens in the approaches of Frank *et al.* (1999) or Hulth (2003), and four tokens in the work of Matsuo and Ishizuka (2003). Statistics on the number of keyword tokens in two reference benchmarks have been reported by Bougouin (2015) (Table 1): unsurprisingly, the majority of keywords consist of two tokens. The statistic reported in SemEval 2017 (Augenstein *et al.* 2017), however, shows that 51% of the assigned keywords have more than three tokens.

We note that in this paper, we do not focus specifically on term extraction nor on named entity recognition, although there has been a long tradition of work on the former topic (Jacquemin and Bourigault 2003) and much effort put into the latter (Nadeau and Sekine 2007).

In computer science and IR, ‘term’ is a polysemous word: it refers both to the linguistic unit of terminologies, which labels a concept, and to the ‘index term’ or ‘descriptor’, which are expected to describe a document content and are part of a controlled or indexing vocabulary. As stated by Lossio-Ventura *et al.* (2013), there are some fundamental differences between term extraction and keyword extraction approaches. One major difference is that extracting terms requires a big collection of texts, which is not a necessary requirement in keyword extraction, which can take only a single document as an input. To highlight another difference, the former approach aims at extracting term-like units and filtering out those that may not be terms, syntactically or terminologically. On the other hand, the latter one extracts the ‘key’ elements of a document, which are not limited to terms. Thus, while keyword extraction approach can be domain independent, term extraction applies only to specialized fields or professional domains. In addition, the main goal of terminology extraction is to build domain terminologies, that is conceptualization of domains, and not to retrieve documents in a heterogeneous collection of documents.

In this paper, we consider domain and language dependent as well as independent methods of keyword extraction. Whenever the applicability and the performance of the reviewed methods depend on these criteria, we try to make it explicit.

3.2 Keyness properties

In keyword extraction, the goal is always to extract important or ‘key’ elements, but ‘keyness’ is an elusive concept and its interpretation depends on the target user and application. It can be associated with various properties, even if they usually are not equally important in all contexts. We distinguish three main types of keyness properties: informational, linguistic, and domain-based.

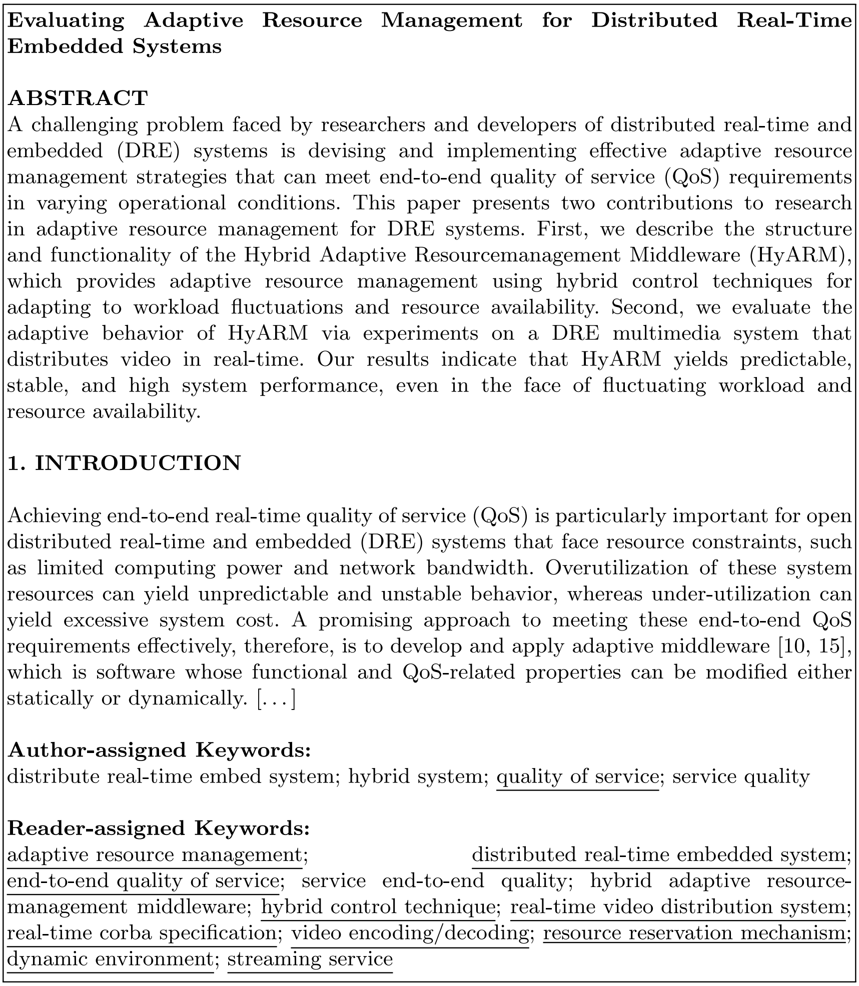
3.2.1 Informational properties

United Nations Educational, Science and Cultural Organization (UNESCO 1975) established several principles to be followed by keywords, independent of any application and of the way keywords are provided, manually or automatically. Two principles, *exhaustivity* and *specificity*, are proposed that are respectively related to a set of keywords and a single keyword. To these two principles, we add three more: *minimality*, *impartiality,* and *representativity*. Minimality, which applies to a set of keywords, is orthogonal to exhaustivity, whereas representativity, which applies to a single keyword, is orthogonal to specificity. In the following, we explain each of the mentioned principles in more detail.

Exhaustivity The set of keywords should cover all the subjects of the studied document, which have potential information value. This principle tends to add implicit keywords to the set of keywords that do not occur in the studied document but target a subject of it.

Specificity Keywords are considered as key elements of a document by contrast with other documents that could belong to other domains. To represent the content of a document, keyword should be ‘as specific as possible’: it is important to select only those linguistic units that are really characteristic of that document and to leave out common textual units that one can find in any similar document. For instance, in legal documents, *common law* and *disclosure statement* are rather common terms and give little information on the content of the documents beyond categorizing them as legal ones. Similarly, *user guide* might be considered as a cross-domain, rather than as a domain-specific term.

Minimality Keywords should differ from each other. The set of keywords has to include unique keywords with different meanings. In Figure 1, *quality of service* and *service quality* in the author-assigned keywords refer to the same meaning and so the set is not minimal.



Evaluating Adaptive Resource Management for Distributed Real-Time Embedded Systems

ABSTRACT

A challenging problem faced by researchers and developers of distributed real-time and embedded (DRE) systems is devising and implementing effective adaptive resource management strategies that can meet end-to-end quality of service (QoS) requirements in varying operational conditions. This paper presents two contributions to research in adaptive resource management for DRE systems. First, we describe the structure and functionality of the Hybrid Adaptive Resourcemanagement Middleware (HyARM), which provides adaptive resource management using hybrid control techniques for adapting to workload fluctuations and resource availability. Second, we evaluate the adaptive behavior of HyARM via experiments on a DRE multimedia system that distributes video in real-time. Our results indicate that HyARM yields predictable, stable, and high system performance, even in the face of fluctuating workload and resource availability.

1. INTRODUCTION

Achieving end-to-end real-time quality of service (QoS) is particularly important for open distributed real-time and embedded (DRE) systems that face resource constraints, such as limited computing power and network bandwidth. Overutilization of these system resources can yield unpredictable and unstable behavior, whereas under-utilization can yield excessive system cost. A promising approach to meeting these end-to-end QoS requirements effectively, therefore, is to develop and apply adaptive middleware [10, 15], which is software whose functional and QoS-related properties can be modified either statically or dynamically. [. . .]

Author-assigned Keywords:

distribute real-time embed system; hybrid system; quality of service; service quality

Reader-assigned Keywords:

adaptive resource management; distributed real-time embedded system; end-to-end quality of service; service end-to-end quality; hybrid adaptive resource-management middleware; hybrid control technique; real-time video distribution system; real-time corba specification; video encoding/decoding; resource reservation mechanism; dynamic environment; streaming service

Figure 1. Example of an annotated text in SemEval 2010 data set. Underlined keywords are those that occur in the text.

Impartiality Keywords should be as objective as possible by reflecting the informational content of the document without involving personal sentiment or opinion. This, however, is not the case for *quality of service* and *service quality* (Figure 1), as the author has been biased by this subject and this repetition of the subject may also bias the readers to the author's point of view.

Representativity Keywords are considered as key elements of a document by contrast with other words or phrases, which reflect minor aspects. While keywords such as *end-to-end quality of service* and *distributed real-time embedded systems* in Figure 1 are representative, phrases like *experiment*, *researchers,* and *resource availability* would not represent the subject of the text.

While there is no arbitrary limit to the number of keywords, the right trade-off between the number of the extracted keywords, which must often be minimized, and the exhaustivity of the document description that they give must be found. Similarly, it is preferable that a keyword is not too specific nor too generic.

3.2.2 Linguistic properties

Keywords are small chunks of a language that should follow the linguistic rules of that language. In the following, we present the desired linguistic properties of a keyword.

Well-formedness is important when the extracted elements are presented to humans. It is essential that keywords are well-formed words or phrases. Truncated forms, such as *emphasi* instead of *emphasis*, incomplete noun phrases such as *legal right to* instead of *legal right*, *onshore wind* instead of *onshore wind farm* are prohibited forms.

Citationess refers to the linguistic form of the keyword that should be retained. Keywords appear in the text under inflectional forms, but only the form without inflection should be kept. Citationess is usually achieved using lemmatization programs. These programs, however, could fail to remove inflection and so could generate wrong lemmas of keywords. In some reference resources, words are in citation forms[a], for example, WordNet (Miller, Beckwith, and Fellbaum 1990).

3.2.3 Domain-based properties

Keyword extraction applies not only to general language documents, such as newspaper articles, but also to documents belonging to specialized domains. Documents with scientific and technical information, which use their own vocabulary, are examples of such domain-specific documents. Domain-based properties are related to keywords within a given domain.

Conformity Each domain has its proper terminology, that is specific terms naming its concepts. These terms have resulted from a consensus within the domain community and keywords should conform with them. Thesauri and domain-specific terminologies provide lists of terms that are recommended to be used as keywords.

Homogeneity Several synonymic forms are often available that refer to the same topic, such as *wind farm* and *wind power farm* in the renewable energy domain. Once a keyword is chosen, it should be used for any document within the same domain dealing with the same topic.

Univocity refers to the unambiguity of the keyword. In a specialized domain, keywords are expected to be less ambiguous than common words (e.g. *aircraft* vs. *plane*), since they are words or phrases whose semantics are stable within a given community and/or context of use. For instance, in aeronautics, *(flight) recorder* always refers to the same type of electronic devices; any domain expert knows what the term means and prefers it to *black box*, which is more colloquial but more ambiguous. Ajgalík, Barla, and Bieliková (2013) provide other examples of such disambiguation.

The above semantic properties do not have the same relative importance for all domains, types of documents and applications. For general domain and web document indexing, unambiguity is crucial. As an example, *penguin* is a bird of Antarctic regions but can also be trademarked as part of a larger name: Google Penguin, GoGo Penguin, Pittsburg Penguins, etc.

Some of these principles, such as well-formedness, might be overridden for sake of efficiency. Truncated keywords that are ill-formed are commonly useful in IR to enlarge the matching of the keyword. We can observe similar overriding with the minimality principle. Looking to the keywords of Figure 1, we notice synonymic keywords in the set of author-assigned keywords: *quality of service* and *service quality*, and in the set of reader-assigned keywords: *end-to-end quality of service* and *service end-to-end quality*.

[a] Some others, however, do not necessarily contain a lemmatized vocabulary, for example, MeSH.

Various types of keyness properties are difficult to capture. In fact, these properties can only be approximated through surface or formal features (Section 5). This explains not only the variety of keyword extraction methods that have been proposed (Section 6) but also the complexity of the evaluation task (Section 4).

4. Evaluation of keyword extraction

Before entering the description and comparison of approaches that have been proposed, we present the methods, measures, and key benchmarks that are commonly used for evaluating keyword extraction approaches and measuring progress. Unsurprisingly, the lack of homogeneity of evaluation methodologies reflects the diversity of the target applications and extraction goals. Keyword evaluation methods adopt a Cranfield evaluation process (Voorhees 2001). Each method is applied to a set of test documents and the extracted keywords are usually compared to a set of manually assigned keywords that constitutes the reference.

4.1 Evaluation methods

Extraction methods can be evaluated extrinsically or intrinsically. In the former case, keywords are directly used in a given application and their quality is measured indirectly through their impact on the performance of that application. As an example, Hammouda, Matute, and Kamel (2005) evaluated keywords in a text summarization application. In intrinsic evaluation, however, keywords are evaluated directly.

Intrinsic evaluation could be done *a posteriori*, where experts are asked to label the extracted units as ‘keyword’ or ‘non-keyword’. Since this approach of evaluation is costly, keywords are mostly evaluated by measuring the match between the human-assigned and the extracted keywords.[b] It should be noted that this approach of evaluation is mostly suitable for low inflectional languages, such as English, and less performant for languages with high inflections, such as Romance languages.

Human judgements must be used carefully, however. People with different degrees of expertise are often asked to distinguish keywords from non-keywords but keyness is not a binary notion and human judgements include an element of arbitrariness and subjectivity (see Section 4.3). It is therefore important to involve different human judges in evaluation and to calculate the degree of agreement among them (Kappa statistics (Viera and Garrett 2005) is widely used for measuring the inter-evaluator agreement).

4.2 Evaluation measures

In the state of the art of keyword extraction, different measures and protocols are used for evaluating the extracted keywords. The most traditional measures are *Precision*, *Recall,* and *F-measure*. *Precision* is the frequency with which retrieved keywords are relevant (Equation 1) and *Recall* is the frequency with which relevant keywords are retrieved by the evaluated approach (Equation 2). To have a trade-off between precision and recall, *F-measure* is calculated (Equation 3).[c]

$$Precision = \frac{Retrieved \cap Relevant}{Retrieved} \tag{1}$$

[b] Note that, generally, the match with the automatic extraction methods cannot be perfect, as expert-assigned keywords are not necessarily extracted from documents.

[c] *F*1-*measure* is the traditional and the most widely used *F-measure*, which calculates the harmonic mean of precision and recall. Depending on the target application and its sensitivity to afford wrong instances or to miss correct ones, β takes different values in calculation of *F-measure*.

$$Recall = \frac{Retrieved \cap Relevant}{Relevant} \tag{2}$$

$$F_{\beta} = (1+\beta^2).\frac{precision.recall}{(\beta^2.precision) + recall} \tag{3}$$

While *a posteriori* evaluation of keywords lets us compute the precision of the extracted keywords, evaluating them through the match with the human-assigned keywords gives both the precision and recall over the extracted keywords.

In spite of the fact that these measures are traditional and basic, they are widely used in well-known challenges, such as SemEval (Kim *et al.* 2010; Augenstein *et al.* 2017) and DÉfi Fouille de Textes (DEFT).[d] Hasan and Ng (2014), however, consider the improvement of the evaluation scheme as a remaining challenge in the field. In fact, the above traditional measures are questionable. They presume the existence of a 'gold' standard but one often has to deal with a long list of less or more relevant keywords which cannot be considered either as a 'gold' reference or as a standard.

In IR, it is common to evaluate the methods according to the quality of the keywords' ranking: one approach performs better than another one if it can give a higher rank to more important keywords. The above-mentioned traditional measures, however, do not take this into account and so in the case of having a list of ranked keywords, other evaluation measures are required. *Precision@K* is one of the measures that can be used for this purpose. This measure ignores keywords ranked lower than K and computes the precision value over the top-K keywords on the list. Singhal *et al.* (2017) use the precision@K measure to compare the different approaches of keyword extraction for different values of K.

Mean average precision (MAP) is another information retrieval-based measure that can be used for evaluating a list of ranked keywords. MAP is the average of precision values at K, where K takes on the ranks at which correct keywords are returned. Both precision@K and MAP have been exploited by Jiang, Hu, and Li (2009) for evaluating their ranking approach of keyword extraction.

Mean reciprocal rank (MRR) (Voorhees 1999) and *binary preference measure (Bpref)* (Buckley and Voorhees 2004) are two other measures that can be exploited for evaluating ranked keywords (Liu *et al.* 2010). MRR evaluates how the first correct keyword of a document is ranked in the list of the keywords extracted by a method. This measure is computed using Equation (4), where D is the set of documents and $rank_d$ denotes the rank of the first correct keyword of d.

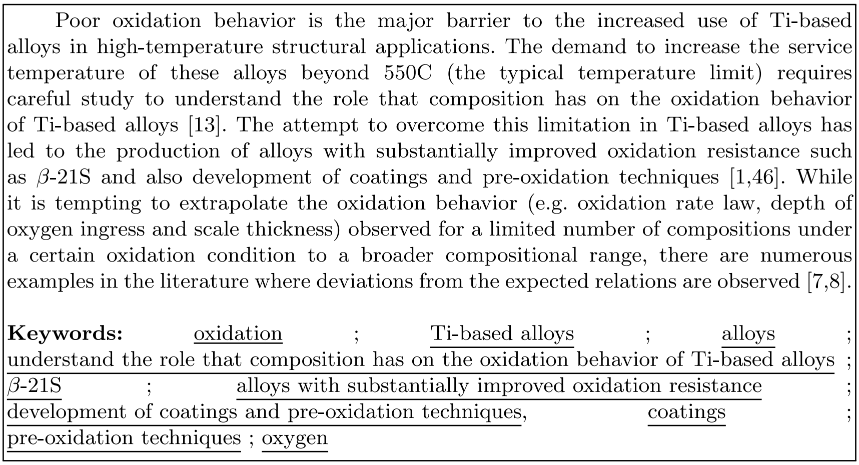
$$MRR = \frac{1}{|D|}\sum_{d \in D}\frac{1}{rank_d} \tag{4}$$

Bpref evaluates the portion of the bad keywords, which are ranked higher than the good ones. More specifically, it is calculated using Equation (5), where M is the number of all the extracted keywords, in which r is a correct keyword and n is an incorrect one, and R shows the list of all the correct keywords within M.

$$Bpref = \frac{1}{|R|}\sum_{r \in R} 1 - \frac{|n \text{ ranked higher than } r|}{M} \tag{5}$$

Considering the 'exact match' between the gold standard keywords and the extracted ones has limitations as it fails to detect two similar but morphologically variant keywords. In order to tackle this problem, Zesch and Gurevych (2009) proposed an approximate matching strategy, which deals with such variations between keywords. These authors believe that this way, keyword extraction approaches can be evaluated in a more precise way.

[d] https://deft.limsi.fr/2012/; http://deft2016.univ-nantes.fr/accueil/



Poor oxidation behavior is the major barrier to the increased use of Ti-based alloys in high-temperature structural applications. The demand to increase the service temperature of these alloys beyond 550C (the typical temperature limit) requires careful study to understand the role that composition has on the oxidation behavior of Ti-based alloys [13]. The attempt to overcome this limitation in Ti-based alloys has led to the production of alloys with substantially improved oxidation resistance such as β-21S and also development of coatings and pre-oxidation techniques [1,46]. While it is tempting to extrapolate the oxidation behavior (e.g. oxidation rate law, depth of oxygen ingress and scale thickness) observed for a limited number of compositions under a certain oxidation condition to a broader compositional range, there are numerous examples in the literature where deviations from the expected relations are observed [7,8].

Keywords: oxidation ; Ti-based alloys ; alloys ; understand the role that composition has on the oxidation behavior of Ti-based alloys ; β-21S ; alloys with substantially improved oxidation resistance ; development of coatings and pre-oxidation techniques, coatings ; pre-oxidation techniques ; oxygen

Figure 2. Example of an annotated text in SemEval 2017 data set. Underlined keywords are those that occur in the text.

4.3 Benchmarks

The cost of collecting training or evaluation data motivates the publication of data sets as public benchmarks, which can be used for comparing approaches or evaluating them against state-of-the-art methods. For instance, in 2010 and 2017, SemEval[e] proposes tracks[f] for keyword extraction. A specific experimental setting is defined. At the end, the participating systems are ranked based on their performance.

There are important features related to how keywords are assigned to documents including the number of annotators involved in the selection of the keywords, their degree of expertise, the guidelines they have to follow regarding the number of keywords to extract, and their length. These features have an impact on the reliability and homogeneity of the resulting keyword lists. In general, type of the assigned keywords can be divided into three main categories depending on the profile of people who provide them. Examples of different annotations have been shown in Figures 1 and 2. Although annotations assigned by authors and readers have been differentiated in Figure 1, we did not find such a distinction between the keywords assigned by readers and professional indexers in the SemEval 2017 data set.

- *Author-assigned keywords* are provided by the author(s) of documents that are aware of their domains and take them into account while annotating the documents. However, Caragea *et al.* (2014) point out that in spite of the expertise, because they want to raise awareness on their paper, the authors may assign it several morphological variants of a keyword, even synonyms. These keywords could be subjective too: authors could be biased by the content of their paper while annotating it.
- *Reader-assigned keywords* are provided by the readers of documents, who can be of various ages and backgrounds. Some have more general knowledge about the domain of the studied document, whereas others may not be very familiar with the domain. Readers are mostly given some guidelines for annotating documents but the reader-assigned keywords could be nevertheless subjective: a reader might consider a keyword as representative of a document, while another reader may not see it as representative enough. In designing NUS benchmark, Nguyen and Kan (2007) reported the Kappa statistic inter-agreement between their two employed readers as 0.70, which is rather low.

[e] An ongoing series of evaluations designed for computational semantic analysis systems.

[f] The task 5 of SemEval 2010 and the task 10 of SemEval 2017 are, respectively, presented by Kim *et al.* (2010) and by Augenstein *et al.* (2017).

Comparing the assigned keywords in Figure 1, the reader-assigned ones are more likely to be selected from the content, while authors would probably choose a keyword that is morphologically different than the existing keywords in the text. Due to the knowledge that authors have about their documents, they can also assign more abstract keywords, which target the main points of the documents. Readers, on the other hand, assign more keywords that reflect different detailed points in the documents.

- *Professional indexer-assigned keywords* are provided by expert indexers, who mainly make use of a large controlled vocabulary for assigning keywords to documents. Here, the quality of the indexation depends on the knowledge of the thesaurus or the reference vocabulary. Comparing to the other kinds of assigned keywords, the ones by professional indexers are more objective, even if the subjectivity is never absent.

Annotation guidelines are different according to the target application and the requirements and they are mainly designed for annotators who are ‘readers’ of documents. As an example, in SemEval 2017 (Augenstein *et al.* 2017) some guidelines are given to the readers, which define the procedure and the rules of the annotation.[g] Figure 3 also shows the reader guideline provided by Sterckx *et al.* (2017). Clearly, the more detailed a guideline is, the more precise the annotations will be. Additionally, by having such guidelines, the level of disagreement between annotators would decrease.

Unlike reader-assigned keywords, commonly, there are no guidelines for authors of documents and they assign keywords to their texts liberally. Examples are the SemEval 2010 (Kim *et al.* 2010) benchmark and also the one generated by Caragea *et al.* (2014), in which all the scientific papers selected for the benchmarks already had some author-assigned keywords. Similarly, professional indexers, who are usually asked to perform very controlled tasks, are not provided with any general guidelines but they have to comply with many specific criteria while annotating documents.

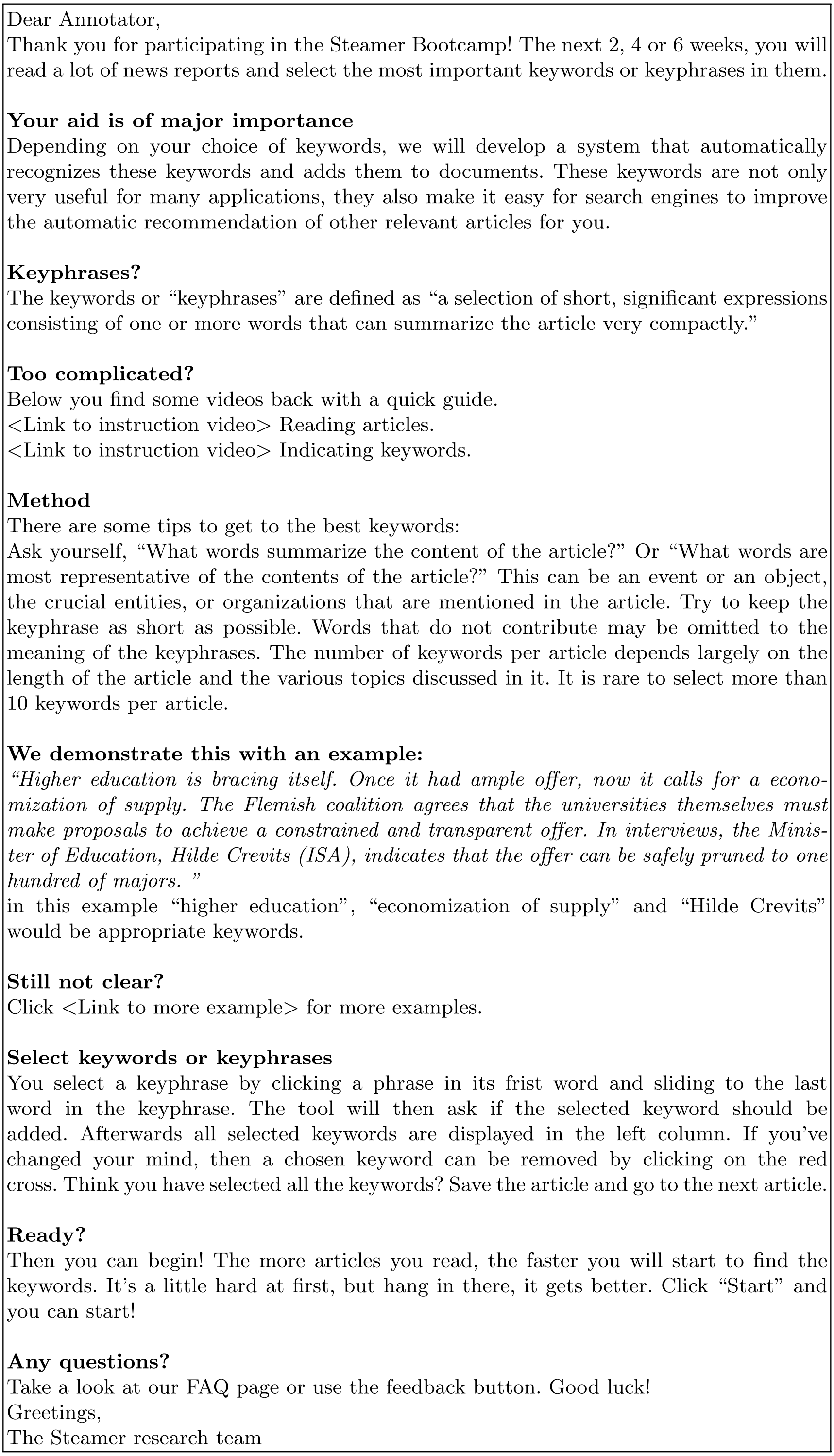
Due to the difficulties in assigning reliable keywords to documents, in some works, more than one way of annotating the gold standard set is used. As an example, Kim *et al.* (2010) made use of both author- and reader-assigned keywords in generating the SemEval 2010 data set. Nevertheless, they found a degree of overlap between the two sets of keywords.[h] Augenstein *et al.* (2017) also recruited both students (readers) and professional indexers for the annotation task. One paragraph was extracted from each article of the data set and two annotations were assigned to it: one by a reader and one by a professional indexer. In case of disagreement between the two annotators, the expert-assigned annotation was taken into consideration. The authors reported the inter-annotator agreement between the readers and the professional indexers in terms of Cohen’s kappa statistics, which ranges between 0.45 and 0.85. This shows that some readers are not qualified enough for annotating documents and their annotations are less reliable.

There are two categories of benchmarks for keyword extraction, depending on whether the keywords assigned to the documents are freely chosen or belong to a controlled vocabulary. Tables 2 and 3, respectively, give examples of widely used public benchmarks in each category. Among the presented benchmarks, Inspec (Hulth 2003) contains both uncontrolled and controlled keywords, where any suitable keyword and thesaurus unit can be assigned to the documents. The other vocabulary-based benchmarks presented in Table 3 and the CiteULike-180 benchmark in Table 2 have been developed by Medelyan (2009).

DEFT, which is a French scientific evaluation campaign, provides French benchmarks for keyword extraction task. More specifically, DEFT 2012 and DEFT 2016 focus on the task of evaluating keywords, extracted by different participant systems. In 2012, the training and the test corpora consisted of 234 full scientific papers published in journals of Humanities. To evaluate the different methods, their extracted keywords were compared with the author-assigned keywords.

[g] The guideline can be found here: `https://scienceie.github.io/resources.html`

[h] Among the 387 author-assigned keywords in the test set, 125 keywords match exactly with the reader-assigned keywords.



Dear Annotator,
Thank you for participating in the Steamer Bootcamp! The next 2, 4 or 6 weeks, you will read a lot of news reports and select the most important keywords or keyphrases in them.

Your aid is of major importance
Depending on your choice of keywords, we will develop a system that automatically recognizes these keywords and adds them to documents. These keywords are not only very useful for many applications, they also make it easy for search engines to improve the automatic recommendation of other relevant articles for you.

Keyphrases?
The keywords or "keyphrases" are defined as "a selection of short, significant expressions consisting of one or more words that can summarize the article very compactly."

Too complicated?
Below you find some videos back with a quick guide.
<Link to instruction video> Reading articles.
<Link to instruction video> Indicating keywords.

Method
There are some tips to get to the best keywords:
Ask yourself, "What words summarize the content of the article?" Or "What words are most representative of the contents of the article?" This can be an event or an object, the crucial entities, or organizations that are mentioned in the article. Try to keep the keyphrase as short as possible. Words that do not contribute may be omitted to the meaning of the keyphrases. The number of keywords per article depends largely on the length of the article and the various topics discussed in it. It is rare to select more than 10 keywords per article.

We demonstrate this with an example:
"Higher education is bracing itself. Once it had ample offer, now it calls for a economization of supply. The Flemish coalition agrees that the universities themselves must make proposals to achieve a constrained and transparent offer. In interviews, the Minister of Education, Hilde Crevits (ISA), indicates that the offer can be safely pruned to one hundred of majors. "
in this example "higher education", "economization of supply" and "Hilde Crevits" would be appropriate keywords.

Still not clear?
Click <Link to more example> for more examples.

Select keywords or keyphrases
You select a keyphrase by clicking a phrase in its frist word and sliding to the last word in the keyphrase. The tool will then ask if the selected keyword should be added. Afterwards all selected keywords are displayed in the left column. If you've changed your mind, then a chosen keyword can be removed by clicking on the red cross. Think you have selected all the keywords? Save the article and go to the next article.

Ready?
Then you can begin! The more articles you read, the faster you will start to find the keywords. It's a little hard at first, but hang in there, it gets better. Click "Start" and you can start!

Any questions?
Take a look at our FAQ page or use the feedback button. Good luck!
Greetings,
The Steamer research team

Figure 3. Annotation guideline designed by Sterckx *et al.* (2017).

Table 2. Examples of public free-text benchmarks

Title/Generator *Annotators*	Type of docs	Docs Number — length	#Tag/Doc
Inspec *Professional indexer*	Abstracts from Inspec database	2000 — 115 words (avg)	9·63
NUS *Authors and readers*	Scientific papers	211 — 4–12 pages	10
DUC (Wan and Xiao 2008) *Readers*	News articles from DUC 2001	308 — 740 words (avg)	10 (max) 8·08 (avg)
CiteULike-180 *Readers*	Publications from CiteULike website	180 — n/a	5 (avg)
SemEval 2010 *Authors and Readers*	ACM conference and workshop papers	284[a] — 6–8 pages	15
DEFT 2012 *Authors*	Scientific papers from journal of humanities	234 — n/a	n/a
(Bougouin *et al.* 2013) *Readers*	News articles from wikinews	100 — 309·6	9·6
(Caragea *et al.* 2014) *Authors*	WWW & KDD titles and abstracts	790 — n/a	4·87 (WWW) 4·03 (KDD)
DEFT 2016 *Professional indexers*	Titles and abstracts of scientific papers	2921 — 151 words (avg)	12
SemEval 2017 *Readers and Professional indexers*	Publications from ScienceDirect	500[b] — n/a	16·37 (avg)
(Sterckx *et al.* 2017) *Readers*	Sport & news articles Lifestyle Magazines Printed press	6908 – n/a	13·8 (avg)

[a] size of the training set: 144
[b] size of the training set: 350

In 2016, the keywords extracted by participant systems were compared with the ones assigned by professional indexers. As in DEFT 2012, the documents of the corpus are all from the same type, that is scientific papers.[i]

It should be mentioned that benchmarks could also be generated from free available data. As an example, Bougouin, Boudin, and Daille (2013) exploited the French version of WikiNews to generate their own reference corpus.

One should choose a benchmark carefully. The type of documents that compose the benchmark and their length are important features to take into account, as extractors are often designed for a specific type and length of document. Some benchmarks like Inspec are made of abstracts of scientific papers, while others are composed of longer documents or web pages.

Length of the extracted keywords mainly depends on the domain under study. According to Medelyan (2009), physics and medical terms are mostly longer than agricultural ones. This property, that is the length of keywords, is less affected by the type of annotators. As reported by Wan

[i] In DEFT 2012, full papers are analysed, while in DEFT 2016 only titles and abstracts are taken into consideration.

Table 3. Examples of public vocabulary-based benchmarks

Title/Generator		Docs	
Vocabulary/Thesaurus	Type of docs	Number — #words	#Tags/Doc
Inspec *Inspec*	Abstracts from Inspec database	2000 — 115 words (avg)	4.47
NLM-500 (2009) *MeSH*	Biomedical research articles	500 — 4,500 (avg)	15 (avg)
FAO-780 (2009) *Agrovoc*	Documents from FAO's repository	780 — 30,800 (avg)	8 (avg)
CERN-290 (2009) *HEP*	Physics docs from CERN server	290 — 6,300 (avg)	7 (avg)
WIKI-20 (2009) *Wikipedia titles*	Computer Science papers	20 — n/a	5 (min)

and Xiao (2008), the average length of the keywords assigned by readers is 2.09. Caragea *et al.* (2014) have also reported that almost half of their author-assigned keywords are bi-grams and they rarely appear as tri-grams or longer *N-Grams*.

In some benchmarks, the assigned keywords are found in the documents. Some other benchmarks, however, may contain keywords out of vocabulary of the studied documents and so the recall value of different approaches to keyword extraction can never reach 100% on these benchmarks. In this case, people mostly report the *reachable recall* value, computed over the keywords which are found in the studied documents. As an example, in Inspec, both controlled and uncontrolled keywords may or may not be in the studied abstracts. In SemEval 2010, the readers were asked to assign keywords only from the content of the documents. Analysing the test set keywords, assigned for 100 documents, showed that 15% of the keywords were not found in the texts. This value was, however, less than the one for author-assigned keywords (19%).

In real applications, one may aim to evaluate robustness of an approach on different types of documents within different domains. In this case, multiple benchmarks should be used to meet the requirements. New benchmarks should also be generated in order to cover more properties and languages. The existing ones are all a bit the same: most of them are in English, which is a problem in evaluating approaches on non-English documents; most of the public ones contain scientific papers and cannot be used for evaluating keywords extracted from web pages or commercial documents; most of the benchmarks also contain a single type of document all from the same domain. The large corpus proposed by Sterckx *et al.* (2017) is a noticeable exception as it combines different types of documents targeted for a 'diverse and layman audience', but it has nevertheless been designed for a specific application (metadata enrichment). Benchmarks provided in DEFT 2016 and SemEval 2017 are also examples of multi-domain benchmarks, which can be used to measure the robustness of keyword extraction approaches along different domains.[j]

5. Keyness computation features

As the keyness properties (see Section 3.2) are difficult to exploit as such in keyword extraction, they are approximated through a variety of features that can be derived from a formal analysis of the source text. Depending on the type of input data and the keyness properties of the desired keywords, keyword extraction approaches make use of different types of features for capturing

[j] DEFT 2016 and SemEval 2017 contain documents from four and three different domains, respectively.

Table 4. Morpho-syntactic feature values associated with the word ‘Cities’

Feature	Value	Feature	Value
Token	Cities	Part of speech	Noun
Lemma	City	Number	Plural

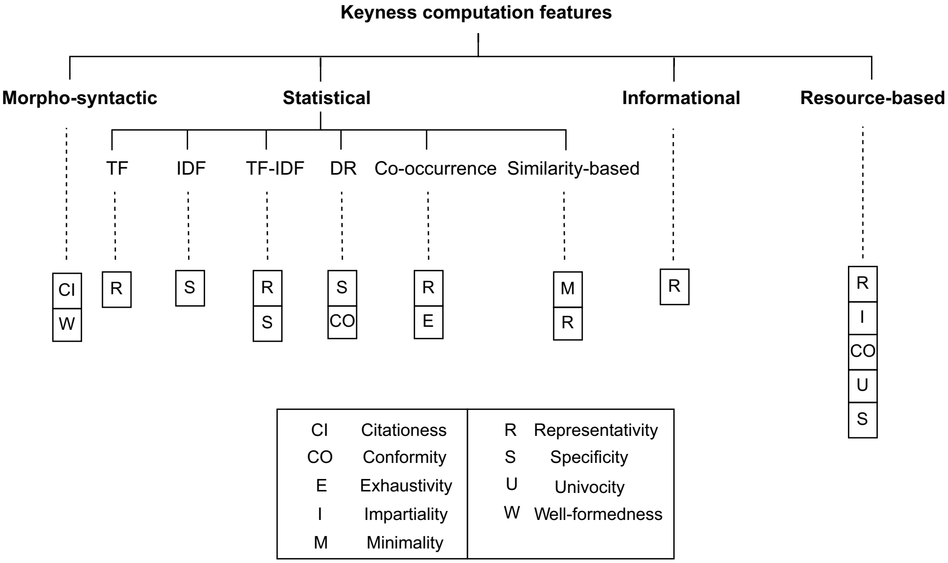


Figure 4. Mapping between the keyness computation features and the keyness properties.

these properties. This section presents the main ones and Figure 4 summarizes the properties that can be captured using any of the presented features.

5.1 Morpho-syntactic features

Extraction tools exploit morphological and syntactical features of textual units. All types of words and word sequences do not have the same probability to be selected as keywords. For instance, some parts of speech, such as nouns and adjectives, are more likely to appear in keywords than others, such as adverbs and determiners, as they provide more information about the text under study. As an example, Mihalcea and Tarau (2004) showed that nouns and adjectives are two important parts of speech tags commonly used when extracting keywords from abstracts of scientific papers.

Extraction methods rely not only on plain words (tokens) but also on their lemmatized forms, their parts of speech (POS tag), and some of their morphological features, such as gender or number (singular, plural). See Table 4 for an example. When the extracted keywords are to be presented to human users, they are mainly retrained without any inflection so as to satisfy the citationess property.

It is also important to take into account the syntactic structure of word sequences, as the sequences that do not correspond to well-formed syntactic phrases are usually discarded from the keyword candidate lists. Hulth (2003) was the first who studied the impact of linguistic information in keyword extraction task. This study showed that having noun phrase chunks as candidate keywords favours precision, while extracting words or phrases with some other pre-defined POS patterns improves the recall value of the extracted keywords. Analysing the annotations on the SemEval 2017 benchmark also shows that 93% of the assigned keywords by readers and professional indexers are noun phrases. Examples of the sequences of POS tags used by Hulth (2003) are as follows:

- Adjective Noun (singular or mass)
- Noun Noun (both singular or mass)
- Adjective Noun (plural)
- Noun (singular or mass) Noun (plural).

Morpho-syntactic features could also be used as a type of filtering at the end of keyword extraction to verify if the extracted keywords have valid linguistic structures.

Although morpho-syntactic features are language-dependent, they are available for a large family of languages for which reliable morpho-syntactic taggers exist. Advanced syntactic features, which require the parsing of the source text, are less widely available as the performance of parsers varies greatly from one language to another. The performance of those tools often decline on texts which contain many technical or out-of-vocabulary tokens or on non-standard language.

5.2 Statistical features

Statistical features were introduced to IR in the 1970s (Salton, Yang, and Yu 1975). These features are widely used on large corpora, even if rarely in isolation: they are mainly language and domain independent and most of them are easy to compute, even on big data.

5.2.1 Frequency-based features

Term frequency (TF) (Luhn 1957) is a very low-level statistical feature, which is widely used in keyword extraction task (Ohsawa, Benson, and Yachida 1998; Turney 2000; Hulth 2003; Yih *et al.* 2006; Rose *et al.* 2010; Sterckx *et al.* 2016). The assumption behind this feature is that the more representative a term is in a text, the more frequent it appears in it. Of course, this feature is both more reliable and more useful when processing long documents than short ones.

Since TF strongly correlates with the size of documents, one usually considers a *normalized frequency* (NF). Equations (2) and (3) show two traditional formulae of NF for a given text. One limitation of TF feature is that it does not distinguish grammatical or common words from content ones as the former are usually highly frequent.

$$NF_{w_i} = \frac{\#\ of\ occ.\ of\ w_i}{Total\ \#\ of\ word\ occ.} \tag{6}$$

$$NF_{w_i} = \frac{\#\ of\ occ.\ of\ w_i}{\#\ of\ occ.\ of\ the\ most\ frequent\ word} \tag{7}$$

Inverse document frequency (IDF) (Sparck Jones 1972) is a quantity borrowed from IR. It is defined by Equation (8), where the document frequency (DF) of a term corresponds to the number of documents in a target collection in which it occurs. The IDF is lower for the common terms that appear in many documents of the collection and higher for those which have a low DF. This measure provides a valuable indication of the specificity of a term in relation to a document but using IDF requires a collection to which the document can be confronted.

$$IDF_{w_i} = log\left(\frac{1}{DF_{w_i}}\right) = log\left(\frac{Number\ of\ documents\ in\ the\ collection}{Number\ of\ documents\ in\ which\ w_i\ occurs}\right) \tag{8}$$

This statistical measure has been adapted to different types of documents. Examples are *inverse tweet frequency* (De Maio *et al.* 2016) and *inverse webpage frequency* (Chung, Chen, and Nunamaker 2003), which respectively compute IDF on tweets and web pages.

Term and document frequencies are traditionally combined in the TF-IDF feature (Equation (9)), which is widely used in IR and keyword extraction (Salton *et al.* 1975; Medelyan and Witten 2006; Wan and Xiao 2008; Zhang *et al.* 2008; Lopez and Romary 2010; Caragea *et al.* 2014). The IDF factor tends to counterbalance the high TF value of common terms present in most documents and to increase the weight of words that appear rarely in the rest of the collection. This way, it favours both representativity and specificity properties of the extracted keywords.

$$TF - IDF_{w_i} = TF_{w_i} * IDF_{w_i} \tag{9}$$

In the literature, there are more variations of the TF-IDF score. *Term frequency-inverse sentence frequency* (Martins *et al.* 2001) is one example, which is computed by replacing 'document' with 'sentence' in Equations (8) and (9). The goal is to compute the score of a term in a document according to its frequency and its distribution in the sentences of that document. Another well-known variation is *Okapi BM25* (Robertson *et al.* 1996), which improves TF-IDF by taking into account the length of documents.

Specificity and conformity of keywords within a target domain can also be captured through the *domain relevance* (DR) score proposed by Navigli and Velardi (2002). The DR of a term is high if it appears frequently in the target domain and rarely in other domains. The formula for computing the DR of term t is presented in Equation (10), where D_i is the target domain, containing a set of documents, and $D_1, ..., D_{i-1}, D_{i+1}, ..., D_N$ represent other domains.

$$DR_{D_i}(t) = \frac{freq(t, D_i)}{\max_j \left(freq(t, D_j) \right)} \tag{10}$$

5.2.2 Co-occurrence-based features

Word co-occurrence is a statistical feature designed to catch keywords representativity and exhaustivity. The basic idea behind using this feature is to capture words which tend to appear together within a given type of context. The basic formula is given by Equation (11),[k] where c is a type of context.

$$Cooc_c(w_i, w_j) = \#\ of\ contexts\ c\ in\ which\ w_i\ and\ w_j\ co-occur \tag{11}$$

Co-occurrence features mainly differ by the type of context which is considered for the co-occurrences, that is the context in which two words are considered as co-occurrent. The computing complexity of the extraction process increases with the size of the context. Momtazi, Khudanpur, and Klakow (2010) list four categories of term co-occurrence features that we present below.

In *sentence-wise co-occurrence*, such as the one used by Matsuo and Ishizuka (2003) and Palshikar (2007), only words occurring in the same sentence are said to be co-occurrent (c = sentence) but it is often less costly to take a smaller and a fixed context into account. In *window-wise co-occurrence*, a sliding window with a fixed size is set as an input parameter (c = window of n words). Any time two words appear in the same window, one co-occurrence is counted. Of course, it is also possible to restrict to well-formed syntactical phrases to avoid accidental co-occurrences of unrelated terms (c = well-formed phrase) but *syntax-wise co-occurrence* requires the initial chunking or parsing of the text. TextRank (Mihalcea and Tarau 2004) is one of the state-of-the-art approaches, which makes use of window-wise co-occurrence. The authors performed experiments on the size of windows being from 2 to 10 words. They, however, reported a higher precision and *F-measure* values for 2-word window, indicating that long windows may

[k] Co-occurrence can be defined as a binary feature (presence/absence of co-occurrence) rather than as a scalar one (number of co-occurrences). The actual formulae are usually more complex: they are sensitive to word order ($Cooc_c(w_i, w_j) \neq Cooc_c(w_j, w_i)$) and only significant co-occurrence scores are considered.

fail to capture the words relation effectively. In the same way, when the whole document is taken into account as the context, the extracted pairs of words are not as strongly associated as in the previous cases. *Document-wise co-occurrence* gives weak semantic association of words. The feature is nonetheless useful as an indication of the semantic similarity of words.

5.2.3 Similarity-based features

Redundant keywords are not interesting for users in real applications and approaches must ensure that their extracted keywords are diverse enough. Computing similarity between the extracted candidate keywords helps to detect the redundant ones and to satisfy the minimality property of the extracted keywords.

The proposed approaches for measuring the similarity of words are often based on the comparison of word vectors, which show the distributions of words within a corpus (Harris 1954; Hindle 1990). Cosine similarity, Dice coefficient, and Jaccard index (Manning and Schütze 1990) are examples of these measures. Considering A and B as two word vectors, Equations (12), (13), and (14), respectively, show the formulae for computing these measures.

$$Cosine(A, B) = \frac{A.B}{||A||_2||B||_2} \tag{12}$$

$$Dice(A, B) = \frac{2|A \cap B|}{|A| + |B|} \tag{13}$$

$$Jaccard(A, B) = \frac{|A \cap B|}{|A \cup B|} \tag{14}$$

Nowadays, similarity between two words or phrases is also computed using a more advanced statistical measure: word embedding (Mikolov *et al.* 2013). As an example, Bennani-Smires *et al.* (2018) exploit word embeddings to remove the redundant keywords extracted by their approach.

It should be noted that some approaches overcome the redundancy problem by grouping keywords into different clusters, where each cluster represents highly similar (overlapped) keywords (Boudin 2013; Bougouin *et al.* 2013). The highest ranked keyword in each cluster is then selected as its representative, whereas the other keywords are considered as the redundant ones.

In addition to the redundancy issue, similarity-based features can be exploited in keyword extraction approaches to capture the relation between words and to detect the representative ones. Turney (2003) captures the words association using the pointwise mutual information computed on Web data. Topic modelling techniques are also widely used for capturing semantic relationship using a corpus of documents. As an example, Liu *et al.* (2010) use the latent Dirichlet allocation (LDA) model (Blei, Boudin, and Daille 2003) to find the topical relationship between words, using Wikipedia as the source of information.

5.3 Informational features

Informational or textual features are language and domain independent. They exploit the information clues that authors use to bring attention to important points in their texts. In fact, these features assume that words or phrases with such author-assigned information are more likely to be representative of a studied document. We distinguish four types of informational features, respectively, based on typography, document structure, keyword position within the source documents, and keyword length.

Table 5. Examples of approaches exploiting informational features (*F*)

Approach	F1	F2	F3	F4
(Frank *et al.* 1999)	Positional			
(Turney 2000)	Positional	Typographical[a]	Length	
(Hulth 2003)	Positional			
(Yih *et al.* 2006)	Positional	Typographical[a]	Structural[b]	Length
(Medelyan and Witten 2006)	Positional	Length		
(Zhang *et al.* 2008)	Positional	Structural[c]		
(Lopez and Romary 2010)	Positional	Structural[d]	Length	
(Sterckx *et al.* 2016)	Positional	Typographical[a]	Length	

[a] Capitalization
[b] Occurrence in 'anchor text', 'Meta' tags, 'title' tag, 'URL'
[c] Occurrence in 'title', 'abstract', 'body (full-text)', 'heading', 'first paragraph', 'last paragraph', 'references'
[d] Occurrence in 'title', 'abstract', 'introduction', 'at least one section titles', 'conclusion', 'at least one reference or book title'

Examples of *typographical features* are underlined, bold, italicized, and capitalized elements but words and keywords can also be highlighted using quotation marks. Any type of typographical emphasis can be exploited to spot the most relevant keywords. Even if the productivity of these features depends on the type and typographical convention of the source text, they are quite reliable when available. As an example, Yih *et al.* (2006) considered capitalized words to be important ones and took advantage of this assumption to extract advertising keywords from web pages.

If the source text is *structured*, one can exploit the fact that some specific text areas are more informative than others. This is obviously the case not only for titles and abstract sections but also for link anchors and some captions. In web pages, more important words are usually put in titles and meta descriptions (SEOmoz 2012). Lopez and Romary (2010) also consider title, abstract, introduction, section title, conclusion, and reference sections of a scientific paper as its informative parts, which are likely to contain keywords.

The *position of the keyword occurrences* within the source text – 'spatial use of the words' for Herrera and Pury (2008) – can also be analysed as an indication of the importance of the studied terms. In academic documents, terms that appear at the beginning or at the end of a chapter are expected to be more informative than the others.

Informational features can be modelled as Boolean values (presence/absence of the term in a given area) or as scalar ones (position of the first occurrence of the term or the average position of the term occurrences), possibly normalized with respect to the length of the text. In any case, informational features are used to measure the representativity of the terms.

The *length* of keywords may also give information about them. In some approaches, keywords with less than three characters are assumed uninformative and so are filtered out from the list of candidate terms. An example is the pre-processing performed by Turney (2000), where words with less than three characters are removed. On the other hand, some approaches may assume that longer keywords contain more specific information.

Examples of approaches which exploit these informational features are listed in Table 5. The 'positional' property in the table indicates the position of the first occurrence of the keywords in the studied document. According to the provided examples, this property is the most widely used among the informational features. Unlike structural property, positional property does not depend on the type of the studied document. Hence, it can be used in a more generic way for targeting various types of input documents.

5.4 Resource-based features

The quality of the extracted keywords can also be measured with respect to an external semantic resource, such as a dictionary or a thesaurus, which provides additional information on the studied words. Due to the dependency on external resources, these features are considered as domain and language-dependent features.

One can exploit an existing gold standard set to spot the word and phrase occurrences in the source text. This amounts to keyword identification rather than extraction but such a *dictionary-based validation* can be exploited to assess representativity and univocity of the candidate keywords within the studied domain. In addition, these features can evaluate the conformity of the candidate keywords with respect to the domain-specific terms. In the case of document annotation for the evaluation task, exploiting such resources reduces the judgement or subjectivity of the annotators and as a result satisfies the impartiality property of the assigned keywords.

Structured semantic resources also help to identify *semantic relationships*, such as groups of synonyms or topic-based clusters, assuming that related terms are more likely to be important than isolated ones. As an example, Wang, Liu, and Wang (2007) exploit the WordNet database (Miller *et al.* 1990) for capturing the semantic relationship between words. Budanitsky and Hirst (2001) give a review on the semantic-based measures, which compute the word relationship using the WordNet semantic network.

Resource-based features are also exploited to assess the specificity of the extracted keywords. Yih *et al.* (2006) use query logs as a resource and assess the relevance of keywords depending on how frequent they are in queries. A domain-dependent thesaurus was also exploited by Medelyan and Witten (2006) for keyword indexing task.

5.5 Conclusion on extraction features

A long list of features has been tested for keyword extraction and only the main categories are presented here. Table 6 presents the categories of features used by some of the approaches introduced in this survey, taking all their steps, including the pre-processing, into consideration.

According to Table 6, statistical features can be considered as the elementary features in extraction approaches. Any new approach to keyword extraction exploits these features in order to take advantage of the basic properties of words within the studied document. Among the statistical features, the frequency-based ones are the most exploited and can be considered as a ‘must’ feature for developing a new approach to keyword extraction. More specifically, in the case of having a corpus of documents, TF-IDF can effectively capture representativity and specificity. However, if such a corpus is not available, TF has been shown to perform effectively enough for extracting representative candidate keywords.

Morpho-syntactic features are mainly to eliminate ill-formed candidate keywords and linguistically uninformative ones. Although they are not used as frequently as the statistical features, many works exploit them as a basic feature to reduce the number of candidate keywords in the extraction task. Moreover, for many types of input documents, such as scientific papers and web pages, the positional feature has been shown to be effective for the keyword extraction task. On the contrary, resource-based features are not frequently exploited. As a matter of fact, finding a relevant resource can be challenging and for some domains such a resource may not be available. In addition, domain-independent approaches are often preferred and the resource-based feature does not meet that requirement.

As shown in Table 6, the presented features are mainly used in combination and not always on an explicit basis. In fact, most of these features have been introduced on an empirical basis to improve the quality of the extracted keywords and it is often afterwards that they have been related to semantic properties and justified.

Table 6. Categories of features exploited in example approaches

Approach	Morpho-syn	Statistical	Info	Resource-b
(Salton *et al.* 1975)		✓		
(Ohsawa *et al.* 1998)		✓		
(Frank *et al.* 1999)		✓	✓	
(Turney 2000)	✓	✓	✓	
(Hulth 2003)	✓	✓	✓	
(Matsuo and Ishizuka 2003)	✓	✓		
(Turney 2003)		✓	✓	
(Mihalcea and Tarau 2004)	✓	✓		
(Yih *et al.* 2006)	✓	✓	✓	✓
(Medelyan and Witten 2006)		✓	✓	✓
(Wan and Xiao 2008)	✓	✓		
(Zhang *et al.* 2008)	✓	✓	✓	
(Lopez and Romary 2010)		✓		✓
(Rose *et al.* 2010)		✓		
(Boudin 2013)	✓	✓		
(Caragea *et al.* 2014)		✓		
(Sterckx *et al.* 2016)		✓		

6. Extraction methods

This section focuses on the core of keyword extraction, that is on extraction methods, that take one or several documents as input, possibly exploit external resources and output a (possibly ranked) list of keywords. We do not consider the way the result is used in applications, nor the interaction with the user if any, although not all systems meet the same needs, as mentioned above.

State-of-the-art keyword extraction approaches are divided into two main categories: *supervised* and *unsupervised*. In the following, we present different approaches of these two categories that have been successively proposed, even if the more recent methods often re-exploit the previous ones. It should be noted that in both supervised and unsupervised categories, the previously explained features for computing the keyness properties are used for either ranking the candidate keywords or classifying them into ‘keyword’ or ‘non-keyword’ categories. In the following, we explain this in more detail.

The degree of language or domain dependency is an important factor to take into account while studying the approaches: some approaches are domain dependent, such as that of Zhang *et al.* (2008), while others are both language and domain independent, such as those of Salton *et al.* (1975) or Mihalcea and Tarau (2004).

Independent of that categorization, there are two main strategies for extracting keywords. The *synthetic* one, such as that of Yih *et al.* (2006), consists of extracting the relevant keywords at once, regardless of the number of their constituent tokens, and then filtering out the least relevant ones. The *analytic* approaches, on the other hand, first extract the most relevant single words, which are then extended to surrounding words and/or merged to generate the final list of (possibly long) keywords. TextRank (Mihalcea and Tarau 2004) is an example of the analytic strategy, where a list of top single words is firstly extracted and then the adjacent ones are merged to generate compound words.

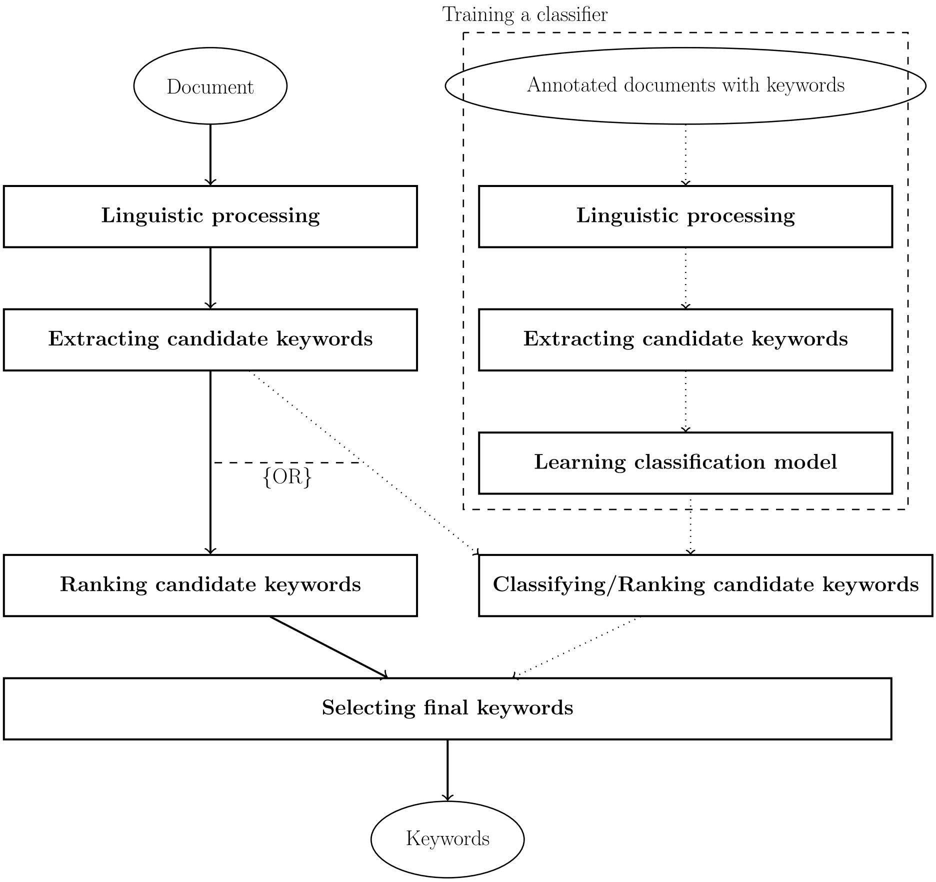

Training a classifier
Document
Annotated documents with keywords
Linguistic processing
Linguistic processing
Extracting candidate keywords
Extracting candidate keywords
Learning classification model
{OR}
Ranking candidate keywords
Classifying/Ranking candidate keywords
Selecting final keywords
Keywords

Figure 5. Overall framework of supervised and unsupervised extraction methods with the synthetic strategy.

State-of-the-art approaches to keyword extraction are mainly synthetic. As explained in Section 5, these approaches make use of pre-defined POS patterns in order to have a list of candidate keywords. While this strategy is mainly regarded as 'keyword extraction', the analytic strategy is considered as 'keyword generation', where the single words may not be necessarily adjacent in the source document. Hence, comparing the two strategies, the synthetic one can better satisfy the well-formedness property of the extracted keywords. However, the analytic one can extract (generate) keywords, which do not appear as such in the document and this may lead to a higher coverage over the studied document. Morpho-syntactic filters, however, could be applied as a post-processing on the generated keywords to discard the ill-formed ones.

Figure 5 illustrates the overall framework of supervised and unsupervised categories with the synthetic strategy, where dotted arrows indicate the supervised category and the solid ones are related to the unsupervised category.

6.1 Supervised methods

Keyword extraction can be regarded as a classification problem, each candidate keyword being labelled as either a keyword or a non-keyword. In the case of using a probabilistic model, probability of a candidate keyword being a keyword is returned. Supervised methods take training data as input and rely on training features for training a classifier. These features are the ones which capture the keyness properties of words and phrases (see Section 5). The trained classifier is then applied to documents for which keywords are aimed to be extracted. As will be seen in the following, the supervised approaches differ in the features used for training classifiers and the types of their classifier(s).

Supervised approach to keyword extraction was first proposed by Turney, who tried two classifiers : (1) C4.5 decision tree (Quinlan 1993), with 12 statistical and morpho-syntactic features, and (2) GenEx algorithm, itself based on Extractor (Turney 2000), a keyphrase extraction algorithm that also exploits a combination of 12 statistical and morpho-syntactic features. Results show that GenEx outperforms C4.5. One drawback of the approach is that words with less than three characters are dropped as uninteresting words, which rules out most of the units and abbreviations.

Various improvements have been proposed on Turney's approach. The Keyphrase Extraction Algorithm (KEA) ,[l] proposed by Frank *et al.* (1999), is based on a Naïve Bayes classifier and uses a smaller set of features, both statistical and informational. More specifically, this algorithm performs a pre-processing on the studied document: splitting the text into phrases based on a set of delimiters and removing non-alphanumeric characters and also all numbers. It then takes phrases up to three tokens as candidate keywords. Ill-formed or uninformative phrases, which either start or end with a stop word or consist only of a proper noun, are discarded. KEA then trains a Naïve Bayes classifier using two features: TF-IDF and the position of the first occurrence of the phrase within the studied document. On a new document, the trained classifier returns the probability of each extracted phrase being a keyword, based on its features.

Turney (2003) improved KEA by increasing the coherence of the extracted keywords using statistical associations between them; using a rule induction approach, Hulth (2003) showed that exploiting morpho-syntactic features improves the performance of the previously proposed supervised machine learning approaches; KEAWeb (Kelleher and Luz 2005) exploits hyperlink structure to improve the keywords extracted by KEA; and KEA++ (Medelyan and Witten 2006) uses the semantic information of a domain-specific thesaurus to overcome the limitation due to word synonymy.

Many different machine learning approaches have been tested for extracting keywords: Zhang *et al.* (2006) applied support vector machine (SVM); Yih *et al.* (2006) trained a classifier using logistic regression algorithm, while Caragea *et al.* (2014) and also Bulgarov and Caragea (2015) used a Naïve Base classifier; and Sarkar, Nasipur, and Ghose (2010) extracted keywords from scientific articles using multi-layer perceptron neural network. According to the results, these approaches outperform KEA algorithm. Augenstein *et al.* (2017) also confirm these trends.

Zhang *et al.* (2008) were the first to consider keyword extraction as a string labelling problem and used the conditional random fields (CRFs) model to extract keywords. These authors exploited statistical and morpho-syntactic features and showed that the CRF model outperforms other machine learning methods such as SVM and multiple linear regression model.

Recently, recurrent neural network (RNN) models are exploited in keyword extraction domain. As an example, Zhang *et al.* (2016) proposed a joint-layer RNN model to overcome the limitation of short content in tweets. Tsujimura, Miwa, and Sasaki (2017) also exploited a neural network-based system for extracting entities and their relations. The authors showed the positive impact of word embeddings on the performance of their neural model. The word embeddings have also been exploited by Meng *et al.* (2017), who aim at inferring 'absent keywords' rather than only extracting the existing ones in the studied document. According to the analysis in their work, absent keywords, which do not exist as such in the source document, appear often in common benchmarks.[m] Hence, predictive approaches of keyword extraction could positively affect the experimental results on these benchmarks.

[l] `http://community.nzdl.org/kea/`

[m] The authors reported the rate of the absent keywords in Inspec, NUS, and SemEval 2010 to be, respectively, 44.31%, 32.25%, and 57.99%.

Supervised machine learning approaches have promising performance in extracting keywords but the training data requirement is a limitation. When no 'naturally' annotated data are available, it must be generated manually for evaluation, and this is not a trivial task. It is not even possible to generate a training set for all types of sources: according to Chen *et al.* (2009), it is impossible to collect a large enough training data set for all types of web content, which hinders the ability to use supervised approaches for extracting keywords from web documents. Sterckx *et al.* (2016) also discuss the difficulty of having reliable training data and its consequences on supervised approaches. The authors overcome this limitation by treating the supervised keyword extraction task as positive unlabelled learning, where non-selected phrases are considered as unlabelled and not as negative examples in the training set in order to model uncertainty in the set.

Among the approaches presented in Table 6, the supervised ones widely use statistical and informational features and this can be an indication that both features perform effectively in this category of approaches. Some approaches also make use of morpho-syntactic features by giving a higher importance to nouns. As mentioned before, using external resources can limit the applicability of the approaches. Consequently, resource-based features are not used very often in the supervised methods.

6.2 Unsupervised methods

As explained before, having a reliable and a rich training data set is challenging in supervised approaches. To circumvent the use of training data, unsupervised approaches have also been tested, considering keyword extraction as a ranking problem. In these approaches, candidate keywords are scored using different kinds of techniques.

6.2.1 Basic statistical methods

Some traditional approaches of unsupervised keyword extraction simply use statistical features in order to extract the most significant words and phrases of a given text. These approaches are mainly language and domain independent. Despite how naive they may be, they have been widely used and compared to other methods. We note that although these statistical methods can be applied independently as unsupervised methods, they can also be used as features in supervised approaches or as a way for selecting candidate keywords in that category of the approaches.

TF-IDF is one of the dominant statistical features. Salton *et al.* (1975) proposed the Theory of Term Importance and showed that the importance of a textual term depends not only on its frequency (or representativity) but also on its specificity.

N-Grams are sequences of elements extracted from a text and *N-Gram* extraction is a common approach for extracting a candidate set of keywords (Hulth 2003; Grineva, Grinev, and Lizorkin 2009; Medelyan, Frank, and Witten 2009). *N-Grams* hypothesis is that a sequence of any length could be a keyword and that statistics will rule out wrong sequences.

N-Grams can be defined either at the character level or at the word level. *Character-level N-Grams* are seldom used in keyword extraction but they help to identify recurring words in spite of small variations, like the plural/singular alternation or words belonging to the same stem (e.g. *visualize, visualizing, visualization*) (Cohen 1995). This feature helps to identify morphological word classes when no lemmatizer or stemmer is available. It is mostly used for single-token extraction, except for languages like German in which the single/multi-token distinction is blurred due to the frequency of compounds. The motivation behind using *word-level N-Grams* is to study all possible sequences of words within a text and to select the important ones as candidate keywords. This allows skipping POS tagging and parsing but it generates many candidate keywords, some of which contain invalid grammatical patterns. For most applications, the resulting list must be then filtered out, a filtering step which adds complexity to the extraction process. Approaches proposed by Hulth (2003) and HaCohen-Kerner (2003) are two examples of, respectively, supervised and unsupervised approaches, which are further applied on the list of candidate *N-Grams* in order to distinguish the representative ones.

Sequential frequent patterns[n] can be considered as a variant of *N-Grams* but they can be of arbitrary length (whereas *N-Grams* are limited to *N* elements). They can also be discontinuous, which is useful to abstract from the surface variations of terms. However, this approach raises the same overgeneration problem as the *N-Gram* one and a much higher computation complexity.

As another statistical method, Tomokiyo and Hurst (2003) proposed to exploit language modelling methods to score candidate keywords based on their 'keyness' (whether or not they are a good descriptor of the studied document) and also the degree to which they can be considered as a phrase.

6.2.2 Entropic methods

Alternative unsupervised approaches are entropy-based and rely on the assumption that 'keyness' is reflected in the spatial distribution of the occurrences of words. In more detail, these approaches are statistical and rely on Shannon's theory of information (Shannon 1948) to quantify the information content of a keyword in a given text based on the distribution of its occurrences in the text. Based on such an entropy measure, one can rank the keywords of the text and detect the more informative ones. The underlying assumption is that words which are more relevant to the topic of the studied text mostly concentrate in some limited areas to represent the author's purpose. On the other hand, irrelevant words have random positions throughout the text. The main advantage of this method is that it needs no external information and no *a priori* knowledge about the structure of the studied document. It is also language independent and requires no additional corpus of documents, as opposed to unsupervised approaches based on TF-IDF.

The first entropy-based approach to automatically extract keywords was designed for literary texts (Herrera and Pury 2008) and experiments on a large book gave promising results. However, the main challenge of this approach is that the studied text needs to be initially partitioned and, according to Carretero-Campos *et al.* (2013), the result of extraction depends on the choice of partitions. Mehri and Darooneh (2011) later defined three other entropic metrics.

The underlying intuition of these approaches was validated by Mehri, Jamaati, and Mehri (2015), who compared a studied text with a shuffled text, where words of the studied text were positioned randomly. The authors showed that the spatial distribution of relevant words significantly differs between original text and the shuffled one, whereas for the irrelevant terms, the distributions are very close.

Carretero-Campos *et al.* (2013) compared the entropic methods to older and simpler 'clustering' ones. This comparison was based on the idea that occurrences of relevant keywords tend to 'cluster', whereas basic terms have a more homogeneous distribution. This phenomenon can be easily captured through the standard deviation of the distance between consecutive occurrences of a word, as shown by Ortuño *et al.* (2002), and it does not require any partitioning of the source text. These clustering approaches seem to perform better on short documents.

6.2.3 Graph-based methods

A wide range of unsupervised keyword extraction approaches is graph-based. In these approaches, the goal is to take into account the connectivity of words and to capture the 'centrality' of keywords. As stated by Beliga, Meštrović, and Martinčić-Ipšić (2015), in the graph theory, 'centrality' indicates the importance of vertices within a graph.[o] Hence, it reflects the representativeness of the vertices and thus the representativity of the keywords.

[n] Cellier *et al.* (2014) give an interesting overview of these works.

[o] Beliga *et al.* (2015) provide details on the centrality measures in the graph-based structure.

The overall approach consists of generating a graph of elements and using it to cluster or to rank those elements.

Graph generation. Depending on the type of analysis and the goal of extraction, the generated graphs can be directed or undirected, and weighted or unweighted.

Connectivity can be measured locally, at the document level, or globally on a collection of documents. In the first case, the idea is to exploit the relation and the connectivity of the words within a single document as proposed by Ohsawa *et al.* (1998). However, more distant cross-document relationships can be exploited. Some works make use of both local and global context to extract keywords of a studied document: the results of Wan and Xiao (2008) show that adding global context increases the performance of the proposed approach.

Graph-based approaches also differ in the selection of vertices and edges. Approaches with different goals use different units as vertices of the graph and also various relations and metrics for generating the edges.

Most often, vertices correspond to candidate single or compound keywords of a given document or collection of documents. The edges reflect the semantic relation of the candidates, which is often measured through co-occurrence (two words are connected in a document graph if they co-occur within a certain window in that document). This type of graphs has been popularized by TextRank (Mihalcea and Tarau 2004) but various algorithms have taken that idea since then. Alternative metrics can also be used to capture the semantic relationship. For instance, Grineva *et al.* (2009) rely on a semantic relationship derived from Wikipedia, and Huang *et al.* (2006) exploit syntactic relations.

A marginal approach considers documents as vertices and relies on the connectivity between documents (Kelleher and Luz 2005). The goal is to capture the intertextuality expressed through the hyperlinks of web documents, in legal document networks for Mimouni, Nazarenko, and Salotti (2015) or in chats for Abilhoa and de Castro (2014). The underlying idea is that documents related to a document *d* within a corpus provide additional information for identifying relevant keywords for *d*.

A more recent approach considers graphs of topics rather than graphs of words or of documents: TopicRank algorithm (Bougouin *et al.* 2013) has been proposed as an improvement over TextRank.

Graph-based analysis. Different *unsupervised* analyses can be performed on the generated graph to get a list of keywords. They are mainly based on clustering and/or ranking algorithms.

Clustering algorithms are used to cluster the nodes of the graph, each cluster corresponding to a set of variant keywords or a group of semantically related ones. For instance, Ohsawa *et al.* (1998) show that clustering a co-occurrence graph outperforms both TF-IDF and *N-Gram* approaches for indexing. Grineva *et al.* (2009) apply community detection techniques on a weighted semantic graph to identify topically related terms and to rule out unimportant ones.

The most common approach relies on *ranking algorithms* and aims at ranking the vertices of keyword graph using the global information recursively computed from the entire graph. This approach was initially proposed for TextRank (Mihalcea and Tarau 2004) and was shown to outperform that of Hulth (2003) in terms of precision and *F-measure*. More specifically, TextRank starts by generating a word graph, where words are restricted to nouns and adjectives within the source document. It then uses PageRank algorithm (Brin and Page 1998) for ranking the nodes of the graph, but the authors claimed that other algorithms, such as HITS (Kleinberg 1999), can also be applied. After selecting the highest ranked nodes, the ones that appear adjacent in the source document are merged to generate compound keywords.

Many variant approaches have been proposed as extensions of TextRank: TimedTextRank (Wan 2007) uses a temporal dimension in order to deal with evolving topics. Wan and Xiao (2008) also extended TextRank by proposing SingleRank and ExpandRank. Unlike the unweighted graph

Table 7. Comparison over graph-based and statistical approaches on the existing benchmarks in terms of *F-measure* values (Bougouin *et al.* 2013)

	Inspec	SemEval 2010	WikiNews	DEFT 2012
TF-IDF	33.4	10.5	34.3	13.2
TextRank	12.7	5.6	8.6	5.7
SingleRank	**35.2**	3.7	19.7	5.9
TopicRank	27.9	**12.1**	**35.6**	**15.1**

in TextRank, in these extensions, the degree of co-occurrence between any two nodes of the graph is represented as the weight of the corresponding edge. While in SingleRank nodes of the graph are scored based on the local information in the studied document, ExpandRank uses a collection of nearest neighbour documents to provide a global knowledge and to improve the effectiveness of the keyword extraction task.

A group of unsupervised approaches extended TextRank by incorporating topical information of the documents under study. Liu *et al.* (2010) proposed Topical PageRank (TPR), which first detects the topics of the studied document and returns its word topic distributions using the LDA model (Blei *et al.* 2003). Using this topical information, TPR then detects keywords of different topics and so achieves the maximum coverage over the studied document by taking all its topics into consideration.

Bougouin *et al.* (2013) also proposed TopicRank: a graph of topics,[p] which aims at ranking topics rather than words. In their approach, keywords are selected from the top ranked topics so as to better target the main topics of the studied document. A comparison over the effectiveness of TopicRank with respect to the state-of-the-art approaches has been presented by Bougouin *et al.* (2013) (Table 7). On all the benchmarks, except for Inspec, TopicRank outperforms the other two graph-based approaches. It, however, performs comparable to the statistical TF-IDF approach.

Zhang, Huang, and Peng (2013) also proposed a topical graph-based approach, where words are assumed to have different relations in different topics. Similar to the approach of Li *et al.* (2010), the latent topics of a document are detected using the LDA model. These authors then proposed a measure for capturing word relatedness, which relies on their relation in different topics.

Structural properties of graphs are also exploited for clustering and/or ranking. The notions of connectedness and betweenness centrality were used by Huang *et al.* (2006) and several authors consider that co-occurrence graphs are similar to small-world graphs (Matsuo, Ohsawa, and Ishizuka 2001). Boudin (2013) compared the efficiency of the PageRank measure with other centrality measures on an undirected and weighted word co-occurrence graph. Results showed that the degree centrality performs comparable to the PageRank but the closeness centrality has the best performance on short documents. Lahiri, Choudhury, and Caragea (2014) extended this analysis to directed word collocation and noun phrase collocation networks. These authors showed that some other centrality measures, such as degree and strength, perform very similarly, or slightly better, when compared with PageRank and are much less computationally expensive. Centrality was, for instance, used by Abilhoa and de Castro (2014), who generated undirected and weighted/unweighted graphs from tweet messages in order to extract their keywords.

6.3 Conclusion on extraction methods

Most of the methods proposed for extracting keywords fall into one of the above categories but even in a given category, they often differ in the features they rely on and in the type and number of keywords that they aim at extracting.

[p] Each topic is regarded as a set of similar single- or multi-token words.

Each method has its own strengths and domain of application but recent works show that combining different types of methods and features leads to more generic and performant keyword extraction approaches. Danesh, Sumner, and Martin (2015), for instance, report promising results with an unsupervised method that combines *N-Grams* candidate generation approach with various ranking steps, respectively, based on traditional statistical features, the position of the first occurrences and a co-occurrence graph. Nevertheless, obtaining a high value of *F-measure* in the task of keyword extraction remains challenging. According to the results reported in SemEval 2010 (Kim *et al.* 2010), the state-of-the-art performance in terms of *F-measure* value is mostly 20–30%. *F-measure* score reported by Hasan and Ng (2014) is also another indication of the difficulty of the keyword extraction task. In that study, the best *F-measure* score was achieved on a dataset of abstracts (45.7%), while on a dataset of papers, the authors reported a score of 27.5%. More recent results, obtained in SemEval 2017 (Augenstein *et al.* 2017), also confirm the difficulty of keyword identification, reporting the highest achieved *F-measure* to be 56%.

Supervised methods show a good performance in state-of-the-art approaches. The combination of different extraction features is an effective way of capturing the importance of words in documents and supervised methods make these combinations easy to test. While choosing a supervised approach, however, one needs to ensure the availability of rich training data, which sometimes needs to be purposefully generated.

The complexity of the supervised approaches depends on the chosen classifier. For instance, the Naïve Bayes classifier used by Frank *et al.* (1999) has been shown to be faster than the one proposed by Turney (2000). The choice of the extraction features is also another important factor in determining the effectiveness of the approaches. As an example, the approach of Yih *et al.* (2006) performed better than KEA (Frank *et al.* 1999) thanks to the exploitation of more diverse extraction features. However, a trade-off has to be found between performance and complexity, when using a wide range of features.

According to the results reported in state-of-the-art approaches, when a corpus of annotated data is available, supervised approaches can be a good choice. However, since such data may not be available for a specific domain or language and generating it could be costly, unsupervised approaches are becoming popular. As an example, Litvak and Last (2008) showed that their supervised graph-based approach outperforms the unsupervised one in terms of precision if a large enough training data set is available. These authors, however, concluded that in the case of a lack of such training data, the unsupervised graph-based approach is a better choice for extracting keywords.

It should be noted that although unsupervised approaches do not have the limitations of the supervised ones, they mostly do not perform as effectively as them and there are still opportunities for further enhancements. In the category of unsupervised methods, the purely statistical ones are quite old and among them, TF-IDF has been widely used as an evaluation baseline. Recent unsupervised methods mainly combine the statistical approaches with other types of approaches, such as the graph-based ones.

Graph-based methods have been widely used since the mid-2000s due to the advantages of 'unsupervised' approaches. With certain combinations of extraction features, their performance is comparable to those of supervised approaches. As an example, Mihalcea and Tarau (2004) obtained higher values of precision and *F-measure* compared to the approach proposed by Hulth (2003). The recall value was, however, higher in the supervised approach.

According to the reported results in this paper, the TF-IDF approach is comparable to some graph-based approaches on specific benchmarks. We note that in spite of the performance, graph-based approaches capture different keyness properties than the TF-IDF. While TF-IDF focuses on the specificity and representativity of each extracted keyword, graph-based approaches also take the exhaustivity of a set of keywords into consideration. Hence, the latter approaches could be more effective in some applications, such as text summarization, where all the topics of a document must be covered by the extracted keywords.

Table 8. Type of the input data in example approaches

Approach	Academic	Tech[a]	Abs[b]	WP[c]	Email	News
(Ohsawa *et al.* 1998)	✓	✓				
(Frank *et al.* 1999)	✓	✓		✓		
(Turney 2000)	✓			✓	✓	
(Matsuo and Ishizuka 2003)		✓				
(Hulth 2003)			✓			
(Mihalcea and Tarau 2004)			✓			
(Kelleher and Luz 2005)				✓		
(Yih *et al.* 2006)				✓		
(Zhang *et al.* 2008)	✓					
(Rose *et al.* 2010)			✓			
(Boudin 2013)	✓		✓			
(Meng *et al.* 2017)	✓		✓[d]			
(Bennani-Smires *et al.* 2018)	✓		✓			✓

[a] Technical
[b] Abstract
[c] Web page
[d] They also evaluated their approach on a dataset of abstracts and titles of academic documents

Our review of the state-of-the-art approaches shows that entropic methods are not as frequently exploited as the others, neither as an extraction method nor as a baseline.

7. Discussion

Evaluation remains a challenge in keyword extraction. Table 8 shows, for instance, that lots of methods have been designed for and tested on limited types of documents. Since the type of data is not the same in all the approaches, comparing them is not feasible. We note that the categorization in Table 8 is based on the data used in the experiments.

The methods are also too different in their objectives and fields of application to be compared directly to each other. The comparative evaluations that have been done and the evaluation tasks that have been proposed focus on a particular family of applications and only cover a small number of approaches.

Another limitation is that the proposed approaches are often compared with the most traditional and generic ones, which usually do not show the same level of performance as the state-of-the-art methods. For instance, TF-IDF and KEA are widely used for the purpose of comparison. For the first one, the main motivation is that TF-IDF is an easy to use approach, which can be applied on any benchmark. For KEA algorithm, there is an open-source software distributed under the GNU General Public License, where both the algorithm and the test data are publicly available. Another limitation of the existing evaluations is that they mainly focus on technical and scientific literature, which leaves out the methods designed for other types of documents. Organizing a large variety of competitions should help to overcome these problems: they should allow for the comparison of the most recent approaches and address different combinations of datasets and target applications.

8. Conclusion

NLP and IR applications require efficient methods for extracting keywords. However, these applications impose various requirements, regarding language and domain dependency, the length and the number of the keywords to extract, the type of the input documents, or the availability of training data.

In spite or because of this complexity, the domain remains very active. Many different approaches have been proposed for selecting, ranking, and clustering candidate keywords extracted from various types of documents, using a large variety of formal features. It is always a matter of capturing some aspects of the 'keyness' notion.

The review of state-of-the-art approaches shows that no single method or feature set can efficiently extract keywords in all different applications. The choice of an extraction tool remains challenging as it must often be carefully designed with respect to the target application and dataset. Moreover, keyword extraction research is often guided by technology and the context of use of methods is not always clearly specified. All this makes it difficult to choose or design an extraction method in practice.

This paper sheds some light on these issues. We tried to map the domain to guide newcomers and help them find their way around this abundant field of keyword extraction. We showed that the target application, the type of source documents, the expected properties, and the type and quality of the extracted keywords are the key questions that need to be addressed when designing an extraction tool and we positioned the traditional features and methods with respect to these questions.

Financial support. This work is (partially) supported by a public grant overseen by the French National Research Agency (ANR) as part of the program 'Investissements d'Avenir' (reference: ANR-10-LABX-0083).

References

Abilhoa W.D. and de Castro L.N. (2014). A keyword extraction method from twitter messages represented as graphs. *Applied Mathematics and Computation* **240**, 308–325.

Ajgalík M., Barla M. and Bieliková M. (2013). From ambiguous words to key-concept extraction. In *24th International Workshop on Database and Expert Systems Applications*, pp. 63–67.

Augenstein I., Das M., Riedel S., Vikraman L. and McCallum A. (2017) SemEval 2017 Task 10: ScienceIE - Extracting keyphrases and relations from scientific publications. In *Proceedings of the International Workshop on Semantic Evaluation (SemEval)*, Vancouver, Canada, pp. 546–555.

Beliga S., Meštrović A. and Martinčić-Ipšić S. (2015). An overview of graph-based keyword extraction methods and approaches. *Journal of Information and Organizational Sciences* **39**, 1–20.

Bennani-Smires K., Musat C., Hossmann A., Baeriswyl M. and Jaggi M. (2018). Simple unsupervised keyphrase extraction using sentence embeddings. In *Proceedings of the 22nd Conference on Computational Natural Language Learning*, Brussels, Belgium, pp. 221-229.

Bharti S.K. and Babu K.S. (2017). Automatic keyword extraction for text summarization: A survey. In *Computing Research Repository (CoRR)*, Volume abs/1704.03242.

Blei D., Boudin F. and Daille B. (2013). Latent Dirichlet allocation. *Journal of Machine Learning Research* **3**, 993–1022.

Boudin F. (2013). A comparison of centrality measures for graph-based keyphrase extraction. In *Proceedings of the Sixth International Joint Conference on Natural Language Processing*, Nagoya, Japan, pp. 834–838.

Boudin F. and Morin E. (2013). Keyphrase extraction for N-best reranking in multi-sentence compression. In *North American Chapter of the Association for Computational Linguistics (NAACL)*, Atlanta, United States, pp. 298–305.

Bougouin A. (2015). *Automatic Domain-Specific Keyphrase Annotation*. PhD thesis, Université de Nantes.

Bougouin A., Boudin F. and Daille B. (2013). TopicRank: Graph-based topic ranking for keyphrase extraction. In *Sixth International Joint Conference on Natural Language Processing (IJCNLP)*, Nagoya, Japan, pp. 543–551.

Brin S. and Page L. (1998). The anatomy of a large-scale hypertextual web search engine. *Computer Networks* **30**, 107–117.

Buckley C. and Voorhees E.M. (2004). Retrieval evaluation with incomplete information. In *Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, Sheffield, United Kingdom, pp. 25–32.

Budanitsky A. and Hirst G. (2001). Semantic distance in WordNet: An experimental, application-oriented evaluation of five measures. In *Workshop on Wordnet and Other Lexical Resources, Second Meeting of the North American Chapter of the Association for Computational Linguistics*.

Bulgarov F. and Caragea C. (2015). A comparison of supervised keyphrase extraction models. In *Proceedings of the 24th International Conference on World Wide Web*, Vol. 30, Florence, Italy, pp. 13–14.

Caragea C., Bulgarov F., Godea A. and Gollapalli S. (2014). Citation-enhanced keyphrase extraction from research papers: A supervised approach. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Doha, Qatar, pp. 1435–1446.

Carretero-Campos C., Bernaola-Galvan P., Coronado A. and Carpena P. (2013). Improving statistical keyword detection in short texts: Entropic and clustering approaches. *Physica A: Statistical Mechanics and its Applications*, **392**(6), 1481–1492.

Cellier P., Charnois T., Hotho A., Matwin S., Moens M. and Toussaint Y., (eds.) (2014). *Proceedings of the 1st International Workshop on Interactions between Data Mining and Natural Language Processing (DMNLP@PKDD/ECML)*, volume 1202 of *CEUR Workshop Proceedings*, Nancy, France.

Chen D., Li X., Liu J. and Chen X. (2009). Ranking-constrained keyword sequence extraction from web documents. In Bouguettaya, A. and Lin, X. (eds), *Proceedings of the 20th Australasian Database Conference (ADC)*, Vol. 92. Wellington, New Zealand: ACS, pp. 161–169.

Chung W., Chen H. and Nunamaker J.F. (2003). Business intelligence explorer: A knowledge map framework for discovering business intelligence on the Web. In *Proceedings of the 36th Annual Hawaii International Conference on System Sciences*.

Cohen J.D. (1995). Highlights: Language- and domain-independent automatic indexing terms for abstracting. *Journal of the American Society for Information Science (JASIS)*, **46**(3), 162–174.

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. In Palmer, M., Boleda, G. and Rosso, P. (eds), *Proceedings of the 4th Joint Conference on Lexical and Computational Semantics (SEM)*, Denver, Colorado, USA, pp. 117–126.

De Maio C., Fenza G., Loia V. and Parente M. (2016). Time aware knowledge extraction for microblog summarization on Twitter. *Information Fusion Journal* **28**, 60–74.

Frank E., Paynter G.W., Witten I.H., Gutwin C. and Nevill-Manning C.G. (1999). Domain-specific keyphrase extraction. In *Proceedings of the 6th International Joint Conference on Artificial Intelligence (IJCAI)*. San Francisco, CA, USA: Morgan Kaufmann, pp. 668–673.

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

HaCohen-Kerner Y. (2003). Automatic extraction of keywords from abstracts. In *Knowledge-Based Intelligent Information and Engineering Systems*. Berlin, Heidelberg: Springer, pp. 843–849.

Hammouda K.M., Matute D.N. and Kamel M.S. (2005). CorePhrase: Keyphrase extraction for document clustering. In Perner P. and Imiya A. (eds), *Machine Learning and Data Mining in Pattern Recognition*, Springer, Berlin, Heidelberg, pp. 265–274.

Harris Z. (1954). Distributional structure. *Word* **10**(23), 146–162.

Hasan K.S. and Ng V. (2014). Automatic keyphrase extraction: A survey of the state of the art. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*. Baltimore, USA: ACL, pp. 1262–1273.

Herrera P.J. and Pury A.P. (2008). Statistical keyword detection in literary corpora. *The European Physical Journal B* **63**(1), 135–146.

Hindle D. (1990). Noun classification from predicate-argument structures. In *Proceedings of the annual meeting of the Association for Computational Linguistics (ACL)*, Pittsburgh, Pennsylvania, pp. 268–275.

Huang C., Tian Y., Zhou Z., Ling C.X. and Huang T. (2006). Keyphrase extraction using semantic networks structure analysis. In *Proceedings of the 6th International Conference on Data Mining (ICDM)*, IEEE, pp. 275–284.

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

Hussey R., Williams S. and Mitchell R. (2012). Automatic keyphrase extraction: A comparison of methods. In *Proceedings of the 4th International Conference on Information Process, and Knowledge Management (eKNOW)*, Valencia, Spain, pp. 18–23.

Jacquemin C. and Bourigault D. (2003). Term extraction and automatic indexing. In Mitkov R. (ed), *The Oxford Handbook of Computational Linguistics*, Oxford University Press, pp. 599–615.

Jiang X., Hu Y. and Li H. (2009). A ranking approach to keyphrase extraction. In *Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, Boston, MA, USA, pp. 756–757.

Kaur J. and Gupta V. (2010). Effective approaches for extraction of keywords. *International Journal of Computer Science Issues (IJCSI)* **7**(6), 144–148.

Kelleher D. and Luz S. (2005). Automatic hypertext keyphrase detection. In Kaelbling, L.P. and Saffiotti, A. (eds), *Proceedings of the 19th International Joint Conference on Artificial intelligence (IJCAI)*, San Francisco, CA, USA, pp. 1608–1609.

Kim S., Medelyan O., Kan M. and Baldwin T. (2010). SemEval-2010 Task 5: Automatic keyphrase extraction from scientific articles. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, Uppsala, Sweden, pp. 21–26.

Kleinberg J.M. (1999). Authoritative sources in a hyperlinked environment. *Journal of the ACM* **46**(5), 604–632.

Lahiri S., Choudhury S.R. and Caragea C. (2014). Keyword and keyphrase extraction using centrality measures on collocation networks. *In Computing Research Repository (CoRR)*, abs/1401.6571.

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*, Manchester, United Kingdom, pp. 17–24.

Liu Z., Huang W., Zheng Y. and Sun M. (2010). Automatic keyphrase extraction via topic decomposition. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Cambridge, Massachusetts: ACL, pp. 366–376.

Liu F., Pennell D., Liu F. and Liu Y. (2009). Unsupervised approaches for automatic keyword extraction using meeting transcripts. In *Proceedings of Human Language Technologies (NAACL)*. Boulder, Colorado: ACL, pp. 620–628.

Lopez P. and Romary L. (2010). HUMB: Automatic key term extraction from scientific articles in GROBID. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, pp. 248–251.

Lossio-Ventura J.A., Jonquet C., Roche M. and Teisseire M. (2013). Combining C-value and keyword extraction methods for biomedical terms extraction. In *LBM: Languages in Biologyand Medicine*, Tokyo, Japan.

Luhn H.P. (1957). A statistical approach to mechanized encoding and searching of literary information. *IBM Journal of Research and Devlopment* **1**(4), 309–317.

Manning C. and Schütze H. (1990). *Foundations of statistical Natural Language Processing*. Cambridge, MA: MIT Press.

Martins C.B., Pardo T.A.S., Espina A.P. and Rino L.H.M. (2001). Introducão à sumarizacão automática. Technical report RT-DC 002/2001, ICMC-USP.

Matsuo Y. and Ishizuka M. (2003). Keyword extraction from a single document using word co-occurrence statistical informationl. *International Journal on Artificial Intelligence Tools* **13**(1), 157–169.

Matsuo Y., Ohsawa Y. and Ishizuka M. (2001). Keyworld: Extracting keywords from a document as a small world. In *Proceedings of the 4th International Conference on Discovery Science (DS)*, volume 2226 of *LNCS*, pp. 271–281.

Medelyan O. (2009). *Human-Competitive Automatic Topic Indexing*. PhD thesis, The University of Waikato.

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*, Vol. 3, Singapore, pp. 1318–1327.

Medelyan O. and Witten I.H. (2006). Thesaurus based automatic keyphrase indexing. In *Proceedings of the 6th Joint Conference on Digital Libraries (JCDL)*, ACM, pp. 296–297.

Mehri A. and Darooneh A.H. (2011). The role of entropy in word ranking. *Physica A: Statistical Mechanics and its Applications*, **390**, 3157–3163.

Mehri A., Jamaati M. and Mehri H. (2015). Word ranking in a single document by jensen-shannon divergence. *Physics Letters A* **379**(28), 1627–1632.

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*, Vol. 1, Vancouver, Canada, pp. 582–592.

Mihalcea R. and Tarau P. (2004). TextRank: Bringing order into texts. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Barcelona, Spain, pp. 404–411.

Mikolov T., Chen K., Corrado G. and Dean J. (2013). Efficient Estimation of Word Representations in Vector Space. In *Computing Research Repository (CoRR)*, pp. 1-12.

Miller G.A., Beckwith R. and Fellbaum Ch. (1990). WordNet: An on-line lexical database. *International Journal of Lexicography*, **3**(4), 235–244.

Mimouni N., Nazarenko A. and Salotti S. (2015). Search and discovery in legal document networks. In *Legal Knowledge and Information Systems (JURIX)*, Braga, Portugal, pp. 187–188.

Momtazi S., Khudanpur S. and Klakow D. (2010). A comparative study of word co-occurrence for term clustering in language model-based sentence retrieval. In *Proceedings of Human Language Technologies (NAACL)*. Los Angeles, CA, USA: ACL, pp. 325–328.

Mori J., Matsuo Y., Ishizuka M. and Faltings B. (2004). Keyword extraction from the web for foaf metadata. In *Workshop on Friend of a Friend, Social Networking and the Semantic Web*.

Muñoz A. (1997). Compound key word generation from document databases using a hierarchical clustering {ART} model. *Intelligent Data Analysis*, **1**(1–4), 25–48.

Nadeau D. and Sekine S. (2007). A survey of named entity recognition and classification. *Linguisticae Investigationes*, **30**(1), 3–26. John Benjamins.

Navigli R. and Velardi P. (2002). Semantic interpretation of terminological strings. In *Proceedings of the Conference on Terminology and Knowledge Engineering (TKE)*, pp. 95–100.

Nguyen T. and Kan M. (2007). Keyphrase extraction in scientific publications. In *Proceedings of the 10th International Conference on Asian Digital Libraries: Looking Back 10 Years and Forging New Frontiers*, Hanoi, Vietnam, pp. 317–326.

Ohsawa Y., Benson N.E. and Yachida M. (1998). KeyGraph: Automatic indexing by co-occurrence graph based on building construction metaphor. In *Proceedings of the Advances in Digital Libraries Conference (ADL)*. Washington, DC, USA: IEEE, pp. 12–18.

Ortuño M., Carpena P., Bernaola-Galván P., Muñoz E. and Somoza A.M. (2002). Keyword detection in natural languages and dna. *EPL (Europhysics Letters)* **57**(5), 759–764.

Palshikar G.K. (2007). Keyword extraction from a single document using centrality measures. In *Proceedings of the 2nd International Conference on Pattern Recognition and Machine Intelligence*, Kolkata, India, pp. 503–510.

Quinlan J.R. (1993). *C4.5: Programs for Machine Learning*. San Mateo, CA: Morgan Kaufmann.

Robertson S.E., Walker S., Jones S., Hancock-Beaulieu M.M. and Gatford M. (1996). Okapi at TREC-3. In *Overview of the Third Text REtrieval Conference (TREC-3)*. Gaithersburg, MD: NIST, pp. 109–126.

Rose S., Engel D., Cramer N. and Cowley W. (2010). Automatic keyword extraction from individual documents. In Berry M.W. and Kogan J. (eds) *Text Mining: Applications and Theory*, Chichester, UK: JohnWiley & Sons Ltd, pp. 1–20.

Salton G., Yang C.S. and Yu C.T. (1975). A theory of term importance in automatic text analysis. *Journal of the American Society for Information Science* **26**(1), 33–44.

Sarkar K., Nasipuri M. and Ghose S. (2010). A new approach to keyphrase extraction using neural networks. In *Computing Research Repository (CoRR)*, abs/1004.3274.

SEOmoz (2012). The beginners guide to SEO. Technical report.

Shannon C.E. (1948). A mathematical theory of communication. *The Bell System Technical Journal* **27**(3), 379–423.

Siddiqi S. and Sharan A. (2015). Keyword and keyphrase extraction techniques: A literature review. *International Journal of Computer Applications* **109**(2), 18–23.

Singhal A., Kasturi R., Sharma A. and Srivastava J. (2017). Leveraging web resources for keyword assignment to short text documents. *In Computing Research Repository (CoRR).*

Sparck Jones K. (1972). A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation* **28**, 11–21.

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, Texas, pp. 1924–1929.

Sterckx L., Demeester T., Deleu J. and Develder C. (2017). Creation and evaluation of large keyphrase extraction collections with multiple opinions. *Language Resources and Evaluation*, **52**, 503–532.

Tomokiyo T. and Hurst M. (2003). A language model approach to keyphrase extraction. In *Proceedings of the ACL 2003 Workshop on Multiword Expressions: Analysis, Acquisition and Treatment*, Sapporo, Japan, pp. 33–40.

Tsujimura T., Miwa M. and Sasaki Y. (2017). TTI-COIN at SemEval-2017 Task 10: Investigating embeddings for end-to-end relation extraction from scientific papers. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, Vancouver, Canada, pp. 985–989.

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

Turney P.D. (2003). Coherent keyphrase extraction via web mining. In *Proceedings of the 18th International Joint Conference on Artificial Intelligence (IJCAI)*, Morgan Kaufmann, pp. 434–439.

Unesco. (1975). UNISIST Indexing Principle SC.75/WS/58.

Viera A. and Garrett J. (2005). Understanding interobserver agreement: The kappa statistic. *Family Medicine* **37**(5), 360–363.

Voorhees E.M. (1999). The TREC-8 question answering track report. In *Proceedings of The Eighth Text REtrieval Conference*, pp. 77–82.

Voorhees E.M. (2001). The philosophy of information retrieval evaluation. In *Evaluation of Cross-Language Information Retrieval Systems: Second Workshop of the Cross-Language Evaluation Forum*, pp. 355–370.

Wan X. (2007). TimedTextRank: Adding the temporal dimension to multi-document summarization. In *Proceedings of the 30th Annual International Conf on Research and Development in Information Retrieval (SIGIR)*, ACM, Amsterdam, The Netherlands, pp. 867–868.

Wan X. and Xiao J. (2008). Single document keyphrase extraction using neighborhood knowledge. In *Proceedings of the 23rd National Conference on Artificial Intelligence (AAAI)*, Chicago, Illinois, pp. 855–860.

Wan X. and Xiao J. (2008). Collabrank: Towards a collaborative approach to single-document extraction. In *Proceedings of 22nd International Conference on Computational Linguistics*, pp. 969–976.

Wan X., Yang J. and Xiao J. (2007). Towards an iterative reinforcement approach for simultaneous document summarization and keyword extraction. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics (ACL)*. Prague, Czech Republic: ACL, pp. 552–559.

Wang J., Liu J. and Wang C. (2007). Keyword extraction based on Pagerank summarization and keyword extraction. In *Proceedings of the 11th Pacific-Asia Conference on Advances in Knowledge Discovery and Data Minin*, Nanjing, China, pp. 857–864.

Yih W.-t., Goodman J. and Carvalho V.R. (2006). Finding advertising keywords on web pages. In *Proceedings of the 15th International Conference on World Wide Web (WWW)*, ACM, Edinburgh, Scotland, pp. 213–222.

Zesch T. and Gurevych I. (2009). Approximate matching for evaluating keyphrase extraction. In *Proceedings of the 7th International Conference on Recent Advances in Natural Language Processing*, Borovets, Bulgaria, pp. 484–489.

Zhang C., Wang H., Liu Y., Wu D., Liao Y. and Wang B. (2008). Automatic keyword extraction from documents using conditional random fields. *Computational Information Systems* **4**, 1169–1180.

Zhang K., Xu H., Tang J. and Li J. (2006). Keyword extraction using support vector machine. In *Proceedings of the 7th International Conference on Advances in Web-Age Information Management (WAIM)*, Springer Verlag, pp. 85–96.

Zhang F., Huang L. and Peng B. (2013). WordTopic-MultiRank: A new method for automatic keyphrase extraction. In *Proceedings of the Sixth International Joint Conference on Natural Language Processing, Asian Federation of Natural Language Processing*, Nagoya, Japan, pp. 10–18.

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*, Association for Computational Linguistics, Austin, Texas, pp. 836–845.

Cite this article: Firoozeh N, Nazarenko A, Alizon F and Daille B. Keyword extraction: Issues and methods. *Natural Language Engineering* https://doi.org/10.1017/S1351324919000457