CHAPTER 4

Keywords

Abstract This chapter tackles the task of keyword extraction from corpora. Keywords are extremely helpful to quickly identify the terms (and their associated concepts) that somehow define what a corpus is about. After a quick revision of the concept of *keyword*, I focus on the different methods that have been proposed to extract keywords effectively and efficiently. A key distinction is made between the reference-corpus method traditionally employed in corpus linguistics and the various methods that have been proposed in Natural Language Processing research. Through several experiments, the CCTC is explored using some of the most outstanding methods proposed to date, and a contrastive description of the results is offered.

Keywords Keyword extraction · Topics · Themes · Reference corpus · Machine learning · Graph-based methods · Keyword set comparison

The term *keyword* has different meanings in different contexts and fields. For a programmer, the keywords of a programming language are the commands and reserved words used in that language, that is, the entire "lexicon" of the programming language. For a web developer or SEO (Search Engine Optimization) specialist, keywords are the set of words and phrases contained in the website, which users might type in their search engines and eventually land them on that website. In an archival

A. Moreno-Ortiz, *Making Sense of Large Social Media Corpora*,
https://doi.org/10.1007/978-3-031-52719-7_4

system of documents, such as a library or bibliographical database, each document is usually assigned a set of words that define its contents and topics. Some document types, such as books, do not usually display its keywords within itself, and rely on archival experts to manually assign those keywords, which will help in the indexing and retrieval processes. Other documents inherently contain keywords; for example, scientific articles systematically rely on three elements that define, catalogue, and classify them: the title, the abstract, and a set of keywords. These three elements can be assessed by prospective readers to decide whether the article is relevant to their interests and therefore worth inspecting any further or reading in detail.

Notomo (2023) rightly states that the notion of *keyword* has long defied a precise definition, and quotes Boyce et al. (1994) for the general definition "a surrogate that represents the topic or content of a document", which makes sense in the context they referred to (libraries and information science). Notomo distinguishes four senses or roles: *terminology:* specialized lexical items from a particular domain; *topics*: terms and labels that are part of a systematic concept system, such as Wikipedia category names; *index terms*: terms indicating major concepts, events, or people, including named entities, and *summary terms*: words or phrases meant to serve as a quick description of the content.

Of these four roles or senses of the term, it is probably the last one that is most often thought of. In this sense, keywords can be loosely defined as words that somehow encapsulate the topics discussed in a document or collection of documents; in other words, what those documents "are about". This is what most authors in corpus linguistics studies agree on: keywords are meant to capture the *aboutness* of a document or set of documents, and make up their ontology (Scott and Tribble 2006; Mahlberg 2007; Bondi 2010; Marchi 2018).

When dealing with very large corpora, keywords are extremely useful pointers or *access points* to an otherwise intractable mass of words whose contents we can only guess from the criteria that were used during the corpus compilation process. For example, in the Coronavirus Twitter Corpus, a number of keywords were used to query the Twitter API for tweets containing them ('COVID-19', 'coronavirus', 'lockdown', etc.). It is safe to assume, then, that these words are key in the corpus, but there is probably a myriad other terms which define and summarize the tweets in the corpus, and which, as a whole, build the ontological scaffolding of the corpus. If we are able to identify those keywords, we will have a means to

access that information, cues to help us to further process and "digest" it. Keyword extraction tools facilitate the identification of words and phrases that fulfill this role, and therefore may be used to further investigate the concepts they refer to and the discourse they define.

4.1 The Concept of "Keyword" in Corpus Linguistics

Within the field of corpus linguistics, keywords are one of the "key" elements in the set of tools offered by corpus query applications. However, this is not the first sense that this term had in corpus linguistics. Originally, the term was synonymous with "search word" in a concordance, which can be defined as "a collection of the occurrences of a word-form, each in its own textual environment" (Sinclair 1991, 32). In fact, this sense of the word gave rise to the term "KWIC" (Key Word in Context), a search results format in which the search word (or "key word") is centre-aligned and the contexts are shown on both sides in such a way that they can be sorted following user-defined criteria and facilitate the task of browsing through potentially thousands of results. The use of the term in this sense was abandoned in favour of others (usually "search word", although phrases, lemmas or more complex patterns can be searched in most concordancers nowadays) after Wordsmith Tools (Scott 1996) introduced the Keywords tool, which offered a convenient way to extract a ranked set of words that stood out as representative of a corpus. The definition that Mike Scott provides is tied to the extraction method:

> A key word may be defined as *a word which occurs with unusual frequency in a given text*. This does not mean high frequency but unusual frequency, by comparison with a reference corpus of some kind". (Scott 1997, 236)

This definition is of a procedural nature, as it is based on the specific approach employed to extract keywords, but says nothing about what keywords are or the purpose they serve. In fact, we do not find very specific actual definitions of the term in the literature. Instead, authors tend to rely on functional approximations that use metaphors to describe their nature and the function they serve in corpus research. Thus, keywords have been referred to as "pointers" that "merit chasing up and tracking down" (Scott 2010, 55–56). Similarly, Baker (2006, 137)

states that keywords "act as signposts to the underlying discourses", while Hunt & Harvey (2015, 139) point out that keywords "serve as indicators of expression and style as well as content to provide a sense of the 'aboutness' of a language variety", and Bondi (2010, 1) qualifies keywords as "markers of the aboutness and the style of a text". Stubbs (2010, 25) compares keywords to the "tips of icebergs: pointers to complex lexical objects which represent the shared beliefs and values of a culture".

In summary, keywords in corpus linguistics are regarded as words and phrases that act as *pointers*, *markers*, *indicators*, or *signposts* to the contents, style, and discourse of a corpus. In the context of social media corpora, I would add yet another metaphorical moniker—that of *access points* that allow us to enter the complex network of concepts, ideologies, discourses, and cultural assumptions hidden behind a mass of bite-sized documents.

Scott's methodological definition, however, is necessary to understand the consideration of keywords in corpus linguistics. Scott (2010) expands the aforementioned definition by offering a very clear illustration:

> In the case of a key word procedure such as that used in WordSmith, this [p-value] calculation is repeated for every single type in the text we are interested in. For example, the frequency of THE in the text is compared with the frequency of THE in the reference corpus, and the p value is then computed of any difference. If the text has 9% of THE and the reference corpus has only 5% of THE, say, we might get a p value suggesting that we can believe, with little risk of being wrong, that in our text THE is prominent. This process is repeated with the frequencies of WAS, the frequencies of IS, and so on until all word-forms have been examined. (p. 48)

Thus, he implicitly establishes a parallelism between a statistical property (a keyness score rendered by a certain metric) with a notional one: the quality of words being outstanding in a corpus:

> The actual calculation of "keyness" is done using the chi-square statistic, but the important point to grasp here is that the notion underlying it is one of outstandingness. In other words, if a word occurs outstandingly frequently in our text, it will be key. Finally, when all potentially key items have been identified, they are ordered in terms of their relative keyness. (Scott 1997, 236)

In summary, this method relies on calculating statistically significant differences between the frequency of words—and possibly n-grams—in a *focus corpus* (the document or set of documents from where keywords are to be extracted) and the frequency of those words in another (*reference*) corpus.[1]

There are several problems with this approach, which Scott himself acknowledges. The most important has to do with the choice of a reference corpus, which will determine to a large extent what is considered to be a keyword. In other words, keywords obtained by this method are relative; they are determined by their frequency in the focus corpus vs. their frequency in the reference corpus. This characteristic is a drawback when we do not have an obvious reference corpus or frequency list. State-of-the-art corpus query tools, such as Sketch Engine (Kilgarriff et al. 2014) make this easy by offering a large number of corpora that can be used as reference, but the actual choice is left to the user, who needs to decide which corpus can be considered "normal" from a statistical point of view.

Then of course there is the issue of the choice of statistical metric to apply. Gabrielatos (2018) discusses this issue at length. He states that "definitions of the terms *keyness* or *keyword* have tended to conflate their nature with the proposed metric for measuring keyness". He goes on to perform a very detailed analysis of the appropriateness of several statistics, concluding that effect-size metrics should be used to measure keyness rather than statistical significant ones.

Since Scott's implementation was ground-breaking, his definition and conception of keywords has stuck within the corpus linguistics community, with few attempts to further elaborate on the actual concept of what the extraction method actually tries to achieve. For example, Stubbs (2010) describes and discusses three different concepts of *keyword*. The first concept dates back to the German tradition of *Schlüsselwörter* dictionaries and glossaries of the early twentieth century and until the 1980s, such as Teubert's (1989) *politishche Vexierwörter* ("ambiguous political words"). In English, he mentions Williams' (1976) work, and in French he mentions Matoré's (1953) work on *mots clés*. This sense of the term

[1] I will use the term "focus corpus" in this book, which is attributed to Kilgarriff (2012). Scott (1997) uses the term "node corpus". Brezina (2018) uses the more explicit term "corpus of interest". The corpus used as reference is usually called "reference corpus", the term that I will use in this book, but other authors have used alternative terms, such as "comparator corpus" (Johnson and Ensslin 2006).

refers to collections of words (and their definitions) that represent and distinguish a society and a culture. Stubbs' second sense of the term "keyword" is conceptually closer to what is generally understood by *keywords* nowadays, but he literally calls it "statistical: keywords are words which are significantly more frequent in a sample of text than would be expected, given their frequency in a large general reference corpus" (Stubbs 2010, 25). Thus, he follows the tradition of tying the definition to the extraction method, specifically referencing Scott's work.

It appears, then, that within the corpus linguistics community there is an implicit understanding that the terms "keyness" and "aboutness" are one and the same thing; and, since keyness is a numerical score obtained by the application of some statistical metric, it follows that words are said to be defining of a text if their relative frequency is statistically significant as compared to their frequency in some other collection of texts against which they are measured.

However, as we will see in the examples, not all words retrieved by this method can be said to be keywords. If anything, they are keyword *candidates* whose actual status as a keyword needs to be validated by the application of certain rather subjective criteria. In other words, a ranked list of keyword candidates resulting from the comparison of word frequencies in a focus corpus against those in a reference corpus computed using a particular statistical metric (that is, the output of all "reference-corpus" keyword extraction tools) cannot be said to exclusively contain keywords that satisfy the criteria of all users. This is because the concept of keyword is rather subjective and depends on the objectives that are being pursued. As Gabrielatos (2018, 26) states, "the identification of an item as key depends on a multitude of subjective decisions regarding a) thresholds of frequency, effect-size, and statistical significance, b) the nature of the linguistic units that are the focus of analysis, and c) the attributes of the compared corpora".

Regardless of the precise statistical metric employed to extract keywords using this method, which I will be referring to as the "reference-corpus method", it is quite apparent that it is a useful system to compare two corpora and highlight differences, which can then be scrutinized in detail. In the words of Alessi and Partington (2020, 3) "this keyword list, providing an ordered series of items which are salient in one corpus compared to another corpus, is likely to suggest items which warrant further investigation".

Many corpus linguistics studies have made extensive use of the reference-corpus keyword extraction method to successfully address linguistic issues. For example, Johnson and Ensslin (2006) used WordSmith Tools to analyse how language and linguistics are represented in articles in the press, specifically from a corpus derived from two British newspapers, The Times and The Guardian. They derived four subcorpora by searching for four "node terms" ('language', 'languages', 'linguistic', and 'linguistics') and extracting all articles that contained these terms. Then they extracted separate keyword lists from each of these subcorpora by using the British National Corpus as a reference corpus (which they refer to as "comparator corpus"). This study is interesting for many reasons. First, because they identify types of words that should be filtered out and considered "noise" results, or false positives:

1. Words that reflect newspaper discourse in general such as 'is', 'has', 'who', and 'says'.
2. Words that refer to the circumtext of the text, such as 'author', 'paper', 'section', 'date', etc.
3. Word forms of the same lemma.
4. Proper names of central public figures.
5. Terms relating to recent technological innovations such as 'WWW', 'Google', and '.com', which did not exist when the reference corpus was created.
6. Word forms which only occurred in one single article or type of article that was considered irrelevant.

Second, they use keyword classification or grouping based on their semantics, which they implement by manually identifying and assigning the automatically extracted keywords to semantic fields, specifically four: "languages", "education", "media culture", and "identity".

Another reason why this study is relevant is that the authors raise two very specific methodological concerns. The first one has to do with the choice of reference corpus, an issue I have already discussed and is well illustrated by this study. In their case, the choice of "an asynchronous comparator corpus" (Johnson and Ensslin 2006, 6)—the BNC—had a strong negative impact on their study because they wanted to analyse the discourse in the media concerning language and linguistics, but the types and very nature of the media at the time when they compiled

their focus corpus were very different than the media at the time when the BNC was created and closed—in 1994, right before the advent of the World Wide Web, and the explosion of Internet technologies, which seriously impacted traditional mass media. Consequently, each and every word related to these aspects were immediately pushed to the top of their ranked keyword lists, since no occurrences were found in the reference corpus. The solution to this problem is not a simple one, because if the choice is made to remove these candidate keywords from the list, then the actual relevant keywords are likely to be ignored, as internet technologies have been key in the development of the media in general. The second major issue they raise is ultimately caused by the same problem (choice of reference corpus), but has to do with proper names; their interest, as critical sociolinguists, aimed to identify "real social actors" engaged in debates over language and linguistics, but since those names occurred worth a statistically insignificant frequency, they were not taken as keywords, and only irrelevant household names (Blair, Chirac, Beckham) ranked high in the lists.

Baker (2004) is another piece of research that illustrates well the shortcomings of the reference-corpus method of keyword extraction. He used this method of keyword analysis to compare the discourses of gay male erotic narratives and lesbian erotic narratives by using two corpora of one million words each.[2]

When two focus corpora are to be compared, several different reference-corpus approaches can be used: first, both corpora can be compared against one reference corpus, which means extracting keywords from both using the same reference corpus, and then comparing the results. Second, the researcher can extract keywords from focus corpus A using focus corpus B as reference and then invert the procedure. Finally, focus corpora A and B can be merged into one and kept as two subcorpora, which can then be individually compared against the whole. Of these three, only the first method can highlight both differences and similarities. The second approach, which Baker uses, has the predictable issue that the analysis will focus on lexical differences, not similarities. The author himself warns about this problem, which "may result in the

[2] He mentions, however, that in this paper he is more focused on the method of analysis than in the discourses themselves. He also warns that he does not seek to denigrate keyword analysis, but "to make researchers aware of possible areas of over- or under-interpretation and suggest ways of ameliorating these issues" (Baker 2004, 249).

researcher making claims about differences while neglecting similarities to the point that differences are over-emphasised" (Baker 2004, 251). Therefore, he also explores the first approach listed above, (comparing both focus corpora against a third reference corpus); for this he uses the Frown (Freiberg-Brown) corpus of general American English, taken from the same time period.

Baker also mentions other practical problems with the application of the reference-corpus method. First, keywords with relatively low frequency may end up ranking high in the list, depending on the specified p-value. He also mentions a well-known issue: in a focus corpus with many individual texts, it is possible that some words with a high frequency may occur only in one or a few texts, which is an indicator that those particular words could only be considered "key" in those specific texts, not in the whole corpus.[3] For example, in one of his two focus corpora, the word "wuz" is listed as a keyword, but it occurs in just one text where these non-standard spelling of "was" is frequently used.

Keyword sets in Baker's paper—and, being quite representative of the type of study commonly found in corpus linguistics, many others in this field—are compared using intuition and, in general, fairly informal methods. This is possible if very high cut-off points are used, that is, only when a manageable number of keywords is considered. However, more strict, formal ways can be devised to compare large sets of keywords. In Sect. 4.3 of this book I propose to use basic set theory to perform this task, which can be used to quickly find differences and similarities between sets and visually represent those using Venn diagrams.

4.1.1 Experiment: The Keywords of Keywords

To conclude this section, I present a brief—and rather "meta"—experiment on keyword extraction with the aim of providing first-hand evidence of some of the issues discussed thus far and others that will become apparent, but will be difficult to identify due to the large amounts of data involved, when a systematic keyword analysis of the CCTC is performed in the next sections. The experiment consists in extracting the keywords from the book *Keyness in Texts* (Bondi and Scott 2010). The book is a collection of articles around the notion of keyness and keywords;

[3] As Baker (2004) reminds us, Scott (1997) proposes the use of *key keywords* to overcome this issue.

it consists of 13 chapters divided into three sections titled "exploring keyness", "keyness in specialized discourse", and "critical and educational perspectives", plus one introductory chapter by Marina Bondi.

For this analysis, all front and back matter was removed, as well as the list of references at the end of every chapter. Headers, which contain page numbers and the names of the various authors and chapter titles were also removed. The remaining text was uploaded as one single file to Sketch Engine, with no mark-up whatsoever. Keyword extraction was performed with the *Keywords* tool using the default settings—focus on rare (N = 1), minimum frequency = 1, case insensitive. In total the focus corpus contains 100,244 tokens (80,783 words). The chosen reference corpus was the 2021 English version of the TenTen corpus family (Jakubíček et al. 2013), which is over 61 billion tokens (52.3 billion words).

The *Keywords* tool in Sketch Engine allows the extraction of two 1,000 keyword sets, one set for single words and one set for multi-words. Output can be visualized on the web app itself or downloaded as CSV, TXT, or Excel files. The online view only permits sorting results by score, but items can easily be sorted by any of the data columns using the downloaded files. These columns are "item", "frequency (focus corpus)", "frequency (reference corpus)", "relative frequency (focus corpus)", "relative frequency (reference corpus)", and "score".

The score, which is the actual keyness indicator, is calculated in Sketch Engine using the *simple maths* approach (Kilgarriff 2009), which is very simple indeed, as it is the result of dividing the normalized frequency (per million words) of a word or n-gram in the focus corpus by its normalized frequency in the reference corpus; an N value ranging from 0.001 to 1,000,000 (1 by default) is added to both the numerator and denumerator. This function gives users the possibility to change the focus of the results, as lower values will return rarer words and higher values more common words. Values can be provided in increments of one order of magnitude.

Other corpus query applications offer considerably more sophisticated statistical methods and options. For example, WordSmith Tools v. 8 (Scott 2022) by default runs four different statistical tests to compare frequencies (Ted Dunning's log-likelihood test, Log ratio, BIC Score, and dispersion difference), and words are only returned as keywords if they pass all statistical tests, although some tests can optionally be skipped. AntConc v. 4.2.0 (Anthony 2023a), on the other hand, allows users to choose between two variants of three different statistical tests

(chi-squared, log-likelihood, and text dispersion keyness) plus a choice of thresholds (p-values) in the range $p < 0.00001$to $p < 0.5$, with or without Bonferroni adjustment. Therefore, these two desktop applications are more suited to advanced users who wish to tweak the comparison methods, whereas Sketch Engine may be more appealing to users who do not care which statistical test is used but want a very wide choice of reference corpora and effective management of their own corpus, as it allows user-defined subcorpora, as described in Sect. 3.5.

I will not focus here on the differences that result from the use of different reference corpora and statistical tests, as this would take ample discussion and this can be found elsewhere.[4] The reference corpus (RC) chosen for this experiment is meant to be general enough to serve as a good reference to extract keywords from a focus corpus (FC) that deals with a very specific topic, although no claim is made that it is representative of the English language.

Table 4.1 displays the top 20 single-word keywords returned by the described method. One important advantage offered by Sketch Engine (SE henceforth) is that statistics can optionally be calculated over lemmas rather than words, which generally returns better results. This is possible because all corpora in SE are not only indexed, but tagged by part of speech and lemmatized. The results shown have been computed over lemmas and sorted by score. All frequencies are relative per million words.

As other authors have mentioned, e.g. Gabrielatos (2018), judging a ranked list of candidate keywords is not easy due to the subjectivity involved, although some objective criteria can be applied. To begin with, none of the items in Table 4.1 are grammatical words, which is sometimes the case; for example, although it is not a fair comparison, as a different reference corpus was used and no lemmatization intervened, AntConc with the default settings did return the preposition "of" in 14th place. Next, almost all of the words clearly refer to concepts in the field of linguistics and, more specifically, corpus linguistics ('corpus', 'concordance', 'collocation', 'collocate', 'n-gram'). Also, the top term is

[4] Brezina (2018) offers a good overview of statistical methods in corpus linguistics, as well as the criteria that enter into play when choosing a reference corpus. Gabrielatos (2018) contains a thorough discussion and comparison of the impact of using the various statistical tests mentioned in this section in relation to keyword extraction. Anthony (2023b) summarizes the most important statistics used in Linguistics.

Table 4.1 Top 20 single-word keywords extracted from the book *Keyness in Texts*

Rank	*Item*	*Freq. (FC)*	*Freq. (RC)*	*Score*
1	keyness	1,207.05481	0.00332	1,204.062
2	concgram	718.24750	0.00063	718.797
3	aboutness	708.27179	0.02546	691.662
4	aboutgram	658.39349	0.00005	659.363
5	lexical	1,416.54358	1.37757	596.215
6	corpus	3,710.94531	5.53609	567.915
7	closed-class	508.75864	0.00366	507.898
8	concordance	708.27179	0.80941	391.991
9	phraseological	379.07504	0.01134	375.814
10	collocate	418.97769	0.19374	351.817
11	**tribble**	389.05072	0.15736	337.019
12	phraseology	428.95334	0.34685	319.229
13	**hyperlink**	907.78503	2.04574	298.379
14	text-type	309.24545	0.05883	293.008
15	collocation	349.14807	0.33713	261.865
16	**wuli**	259.36716	0.00614	258.777
17	**kws**	279.31845	0.09387	256.264
18	n-gram	279.31845	0.12420	249.35
19	**hunston**	249.39148	0.01227	247.357
20	**key-key**	239.41583	0.00035	240.332

'keyness', closely followed by 'aboutness', both of which surely refer to the core concept discussed in this book.

But this list of single-word keywords also contains some awkward items, which have been highlighted in bold. The word 'tribble' is ranked in 11th position. This is because it is a fairly uncommon family name that has 39 occurrences in the FC, as many authors in the book cite Scott and Tribble's (2006) book. Proper names may be argued to be part of the ontology of a corpus, but if this is true, Scott's name should be up there too, as he is the one to actually be credited with the concept of *keyness*; however, 'scott' is listed in position 733, since it is a much more common name in English (relative frequency is 778.1 in FC vs. 42.72 in RC, keyness score = 17.82). The same can be said of 'hunston', in reference to the linguist Susan Hunston, who is mentioned 52 times in the book.

Another issue is raised by the word 'hyperlink', which does not belong in the realm of linguistics. Its absolute frequency is 91, resulting in a

very high relative frequency compared to the RC (97.78 vs. 2.04), and therefore a very high keyness score. However, literally all occurrences of this word take place in one specific chapter of the book—"Hyperlinks: Keywords or key words" by Jukka Tyrkko—which focuses on the status of hyperlinks as keywords. This is a well-known problem with this method of keyword extraction that has been pointed out by many authors. In fact, Egbert and Biber (2019) have proposed the concept of *text dispersion keyness* as an alternative, or perhaps complimentary, method of keyword extraction to overcome this problem, and has been implemented by some corpus query packages, such as the latest version of AntConc. The same can be said of the word 'kws', which is exclusive to the chapter by Mike Scott, whose familiarity with keywords after many years of closely studying them probably leads him to use this abbreviated form.

A similar, but distinct issue is raised by the word 'wuli', whose dispersion plot is limited to the chapter by Fraysse-Kim, a corpus-based analysis of school textbooks that focuses on the Korean word 'wuli' ('we', 'our'). This illustrates a recurrent problem in keyword extraction using the reference-corpus method: foreign words tend to rank high in the lists, as few cases (or none) may be present in the reference corpus.

Finally, the last item in the list ('key-key') refers to the multi-word item 'key-key word'. Hyphenation, compounding, and word boundary marking in general are also a source of problems. First, many keyword extraction tools can only extract single words, but even those that are able to deal with n-grams, such as SE, do not discriminate between actual compounds and the constituent items that make it up. Thus, they sometimes return the whole compound, and also parts of it. Table 4.2 lists the top 20 multi-word keywords identified by SE, listed by score. An example of this issue is apparent here: both 'issue of climate' and 'issue of climate change' are given, when in fact only the latter is an actual multi-word unit. Also, this is another example of the "condensation" issue, as all of the occurrences of this multi-word expression come from the chapter by Denize Milizia, which deals with the importance of looking at phraseological combinations and not just individual words when it comes to keyword analysis.

Similarly, we have the inclusion of 'school textbooks' and 'history textbooks'; the reason is that the last two chapters of the book, by Soon Hee Fraysse-Kim and Paola Leone, focus on the analysis of these two text types, respectively.

Table 4.2 Top 20 multi-word keywords extracted from the book *Keyness in Texts*

Rank	*Item*	*Freq. (FC)*	*Freq. (RC)*	*Score*
1	key word	1,516.30017	2.02158	502.16
2	reference corpus	448.90466	0.00547	447.46
3	closed-class keyword	409.00204	0	410
4	speech act	359.12375	0.17581	306.28
5	semantic field	279.31845	0.03362	271.2
6	metaphor theme	259.36716	0	260.37
7	target fragment	259.36716	0.00427	259.26
8	concordance line	239.41583	0.00566	239.06
9	key-key word	229.44017	0	230.44
10	lexical item	249.39148	0.08844	230.05
11	lexical word	219.46451	0.00751	218.82
12	**issue of climate**	239.41583	0.15798	207.62
13	**issue of climate change**	229.44017	0.13350	203.3
14	specialised corpus	199.51318	0	200.51
15	**pos neg**	179.56187	0	180.56
16	discourse community	179.56187	0.04972	172.01
17	speech event	169.58621	0.02205	166.91
18	**school textbook**	189.53752	0.15944	164.34
19	**history textbook**	189.53752	0.16197	163.98
20	**la repubblica**	169.58621	0.06052	160.85

As for the 'pos neg' n-gram, all of the occurrences are headers in a particular data table in the book where they are used as abbreviated forms of 'positive' and 'negative'. Finally, 'la reppublica' is an example of both the proper nouns and the foreign words issues already mentioned.

This short analysis gives us an idea of what can be achieved through the reference-corpus method of keyword extraction commonly used in corpus linguistics, as well as some of its limitations and issues.

It is critical to understand that proper manual assessment of keyword lists, such as the one I have attempted to carry out in this experiment, is only possible when the contents of the focus corpus are actually known to the researcher. This, however, is not the case when keyword tools are used for the purpose of exploring and identifying key concepts in an unknown corpus, which is the main objective of keyword extraction when applied to very large corpora. This is one aspect that corpus linguists fail to mention or even be aware of, as they often analyse corpora of themes, topics, or

authors they are already familiar with, and their aim is to discover the finer details of the underlying discourse.

4.2 Keyword Extraction Methods in Natural Language Processing

The reference-corpus method commonly employed in corpus linguistics is inherently statistical, as it uses various such metrics to compare the frequency of words and phrases in the corpus of interest (or focus corpus) with those in another—reference—corpus. However, there are other ways to identify keywords that do not make use of a reference corpus, and have some practical advantages, the most obvious one being that a reference corpus is not needed.

Outside the corpus linguistics community, in particular Natural Language Processing (NLP), other approaches to keyword extraction are regularly employed. Specifically, unsupervised and graph-based methods have been shown to be very effective in keyword extraction. Supervised machine learning approaches are also effective to extract keywords in some specific scenarios in which training data is available, which is not the case in social media corpora.

In addition, topic modelling, a common NLP task, can be said to fulfil the same role as keyword extraction, as the objective of these algorithms is to identify salient words and cluster them into semantically related sets which, as a whole, are said to identify a given topic. Topic modelling itself is a complex task where multiple methods and algorithms have been proposed over the years. We explore these in Chapter 5.

4.2.1 Machine Learning Approaches

Generally speaking, the—supervised—machine learning approach to information retrieval consists of creating a prediction model using training documents containing known labels, and then employs the model to identify those labels in new documents, "new" meaning not used during training. In the case of keyword extraction, this means that known, "good" keywords assigned to documents need to exist for the model to be created in the first place. This is why proposed machine learning systems have focused on extracting "metadata keywords", that is, keywords used to summarize the contents of research articles used for archival purposes.

A good exemplar of a machine learning-based keyword extractor is Kea (Witten et al. 1999), which uses Naïve Bayes as the learning algorithm for keyword extraction. Kea builds upon the work of Turney (2000), who was the first to approach this problem as one of supervised learning from examples. Kea's creators build and evaluate the predictive model using a dataset of research articles with known keywords[5] (manually assigned by the original authors of the articles). Specifically, they used a subset of the Computer Science Technical Reports section (46,000 documents) from the New Zealand Digital Library. The subset consisted of the 1,800 documents that had assigned keywords, of which they used 1,300 for training and 500 for testing. As training features for the Naïve Bayes classifier, they used fundamentally discretized TF-IDF scores. Instead of using the common evaluation method used in information-retrieval, they simply counted the number of true positives in the top 20 keywords retrieved by Kea, i.e. the number of matches between keywords that were retrieved by Kea and those that were assigned to the original articles. They found that, on average, Kea matched between one and two of the five keywords chosen by the authors, which they considered good performance.

This example illustrates very well the limitations of supervised machine learning approaches to keyword extraction, the most important of which is that such systems require labelled data for training and testing the system, which is only available for a very specific concept of *keyword*, i.e., the one that refers to keywords as metadata in archival systems. Also, supervised methods have a relatively long training time (Campos et al. 2018).

4.2.2 Unsupervised Approaches

Unsupervised, statistical approaches have been shown to be effective in keyword extraction. Of these, TF-IDF is the most common method in NLP, to the point that it has become the baseline method against which others are measured (Sun et al. 2020). Other, simpler methods have been used, such as noun phrase (NP) chunking (Hulth 2004), which, operating under the assumption that most keywords are nouns or noun phrases, extracts these and then uses some filtering strategy, such as frequency.

[5] Kea's authors use the term *keyphrases*, and they explain that it is meant to subsume the term *keywords*. This use of the term, i.e. *keyphrase* to refer to both single and multi-word items has stuck with many authors in the NLP literature.

TF-IDF is really the combination[6] of two individual calculations: *term frequency* and *inverse document frequency*. The former is literally the relative frequency of a word in a document (i.e. the result of dividing the absolute frequency of a word by the total number of words). The inverse document frequency of a term or word is the—logarithmically scaled—division of the total number of documents in the corpus by the number of documents that contain that word. If multi-word keywords are also extracted, the calculations are then applied to the n-grams in the texts. The IDF part of the equation, which was proposed by Karen Spärck Jones in 1972 (Spärck Jones 1972) with the name "term specificity", plays the role of the reference corpus, as it provides a score indicating the expected probability for a given term to occur in a document that is part of a corpus.

There is an important difference, however, between the reference-corpus method and the TF-IDF method, as the latter assumes that the corpus is organized as a set of *documents* and the terms will be extracted from a subset of documents (typically one) from the whole set. This is very different from the reference-corpus method, where no internal organization of the focus and reference corpora is assumed (although it may of course exist). IDF will return zero for any word that occurs in all documents in a corpus, which is an indication that it does not have a "special" status in the corpus.

There are some important considerations to bear in mind when using TF-IDF for keyword extraction. Since TF-IDF is a multiplication of the term's relative frequency by its inverse document frequency, it follows that any term that occurs in all documents will return a TF-IDF score of zero, regardless of how high its frequency is in the "focus" document or set of documents. Thus, if we have a corpus consisting of 1,000 tweets about the COVID-19 pandemic, and the term "COVID-19" occurs in all of them, it will be discarded it as a keyword of any subset of tweets in that corpus, and it will obtain a low score if it occurs in a high proportion of them. Of course, this situation is unlikely in the case of tweets, given the very special nature of this type of document, but it may be an issue in certain scenarios.

Consequently, when using TF-IDF, it is important to decide exactly what is taken as a document and what is taken as the whole collection

[6] It is in fact the multiplication of these two scores, which is the reason why it is sometimes expressed as "TFxIDF".

of documents (i.e. the corpus). In most situations this will be straightforward, but in the case of a diachronic Twitter corpus, not so much, as it will be dictated by our interests. For example, if we want to extract keywords from a particular time span, say a week, we may take the "document" to be all of the tweets in that week, and the whole corpus would be all of the tweets in the corpus aggregated by week (i.e. one week, one document). This would probably return the word "lockdown" as a keyword candidate for weeks when lockdowns were announced, since it will occur with a higher frequency in those weeks, and it will not occur in all weeks. However, if it does occur in all weeks, the term will get a score of zero, and so it will be discarded as a keyword. As a result, the TF-IDF tends to give higher scores to rare words, which may result in ranking misspellings high. Nonetheless, this method does have advantages from a purely technical perspective, as it is easy to implement and is also extremely fast.

Therefore, TF-IDF is rarely used in isolation, and there have been many other keyword extraction techniques that incorporate it into a more sophisticated process, such as KP-Miner (El-Beltagy and Rafea 2009).

Yake! (Campos et al. 2018) is an interesting tool because it takes into account a number of textual and linguistic parameters to calculate keyword scores, including language (TF-IDF is language-independent). It proceeds in six steps: text pre-processing, feature extraction, individual terms score, candidate keyword list generation, data deduplication, and ranking. The list of features that are used to obtain keyword candidates includes capitalization, word position, word frequency, word relatedness to context, and "word DifSentence", which quantify how often a candidate word appears within different sentences.

Another keyword extractor that has gained attention in the NLP community is RAKE (Rapid Automatic Keyword Extraction) (Rose et al. 2010). The authors' motivation to develop RAKE was "to develop a keyword extraction method that is extremely efficient, operates on individual documents to enable application to dynamic collections, is easily applied to new domains, and operates well on multiple types of documents" (p. 5). RAKE uses an extremely simple approach that uses stopwords and phrase delimiters to divide the document text into candidate keywords, which are sequences of content words occurring in the text. It assumes that most keywords are in fact multi-word units that rarely contain any stopwords and therefore they mostly extract multi-word keywords and are hardly applicable to languages which do make use

of stopwords in noun phrases. Finally, the system takes into account co-occurrences of words, which it measures using word association metrics, to score candidate keywords.

The performance of RAKE was measured in terms of precision and recall against TextRank, the graph-based system proposed by Mihalcea and Tarau (2004), which is described in the next section. In the dataset used by the authors, RAKE performed marginally better than TextRank (F-score of 37.2 for RAKE, 36.2 for TextRank). However, this dataset consisted of short technical abstracts, for which RAKE seems particularly well-suited. However, its performance leaves much to be desired when extracting keywords from large texts, as will be made evident in the experiment that follows.

The most obvious advantage of unsupervised methods in general is that they can be easily implemented and run over large amounts of text, as they are generally fast and do not require any labelled data.

Experiment: Unsupervised Methods vs. Reference-Corpus Keyword Extraction

The aim of this experiment is to compare the performance of these two methods of keyword extraction. I will use a simple script that extracts keywords using the three algorithms that were described—TF-IDF, Yake!, and RAKE[7]—from a subset of the geotagged Coronavirus Twitter Corpus (see Sect. 3.4), specifically the 50% sample of the tweets generated in the U.K. The tweets from the two years that the corpus comprises were aggregated by week and saved to individual weekly files for a total of 102 weeks/files, which were saved as raw text, XML, and JSONL formats. For this experiment the raw text files were used, which were fed to all three keyword extractors. The subcorpus contains over 17 million words (709,099 tweets). Thus each week, which to these keyword extractors are "documents", consists of approximately 7,000 tweets and 173,000 words on average.

In the experiment, the top 100 keywords were extracted for each week, and extraction was limited to n-grams in the range 1–3. The full

[7] The script uses existing Python implementations of these systems. For TF-IDF, it employs Scikit-learn library (Pedregosa et al. 2011); for Yake!, it uses the authors' own implementation found in https://github.com/LIAAD/yake (Campos et al. 2018); for RAKE, it uses the code in https://github.com/u-prashant/RAKE [Accessed 3 May 2023].

results are provided in the book's repository[8] Here we show the top 20 keywords returned by each system corresponding to three different periods of the whole dataset: week 2 (January 27 to February 2, 2020), shown in Table 4.3, week 31 (August 31 to September 6, 2020), shown in Table 4.4, and week 85 (September 13–21, 2021), shown in Table 4.5. Of the three systems, RAKE was the fastest (about 1 minute), then TF-IDF (about 3 minutes) and finally Yake!, which was the slowest by far (14 minutes).[9]

Table 4.3 Unsupervised keyword extraction methods (U.K. Week 2)

TF-IDF	*Yake!*	*RAKE*
coronarvirus	coronavirus	🙏🙏💯
declared global	China	😱😱😱🤔😟 id recommend
wuhan coronavirus	Wuhan Chinese Coronavirus	🟥 ⓢⓚⓨ ⓣⓔⓒⓗ
wirral	Wuhan Coronavirus	Please Take Care
brexitday	Coronavirus outbreak	🟥 cadeaux gifts
global health emergency	Wuhan	vaping lung injury
confirmed uk	CHINA CORONAVIRUS	usual terrorist attacks
declared global health	Chinese	unusual beggers belief
coronaravirus	coronavirus cases	trades persons van
coronavirus confirmed uk	Coronavirus Wuhan	subconsciously chew pens
coronavirus declared global	Coronavirus Wuhan diary	rewarding excellence conference
coronavirus confirmed	Wuhan China	repost whitley bay
ighalo	coronavirus cases confirmed	quid pro quo
china coronavirus	Chinese coronavirus	model 👧: tatiana
coronavirus coronavirus	people	minju kins creations
coronavirusuk	corona	matt hancock enlisted
arrowe park	Coronavirus confirmed	jimdavidson jim davidson
kobe	virus	confidently predict armageddon
coronavirus outbreak	Chinese people	challenged ronnie pickering
coronavirusoutbreak	China virus-hit Wuhan	bill gates foundation

[8] https://osf.io/h5q4j/.

[9] The script was run on a 2.3 GHz 8-core Intel MacBook Pro.

Table 4.4 Unsupervised keyword extraction (U.K. Week 31)

TF-IDF	*Yake!*	*RAKE*
push parliamentary debate	Government COVID support	egunje primate
push parliamentary	covid	cliffordstott h
help push parliamentary	Government COVID	waterhall 3g
support help push	lockdown	
debate sign share	CoVid support	**Who Are We**
work small micro	social distancing	@ mertonlibdems
covid support help	Covid test	wirral tankard
awareness retweet followers	people	whoa whoa whoa
clients raise awareness	Covid pandemic	twisted terrace takeover
friends family advise	pandemic	trevor francis tracksuits
advise clients raise	coronavirus	totes beauts !′.
family advise clients	covid lockdown	stunningly revised choreography
family advise	COVID cases	stuffing guylian shells
raise awareness retweet	back	rhondda cynon taf
clients raise	Covid times	recite surah ka
small micro business	Covid deaths	poacher boyle pounces
micro business government	government	niki 01,908 395,692
business government covid	time	newly refurbished omniplex
sign share ask	Post Covid	mydaddy speculating
share ask friends	COVID safe	multifunctional workhorse robots

Although all three methods appear to have several issues and biases towards particular types of words and phrases, RAKE's results seem entirely random, with no keywords in reference to the relevant topics whatsoever. The conclusion is that this system was designed to rapidly extract keywords from short texts, such as the scientific abstracts on which it was evaluated, and seems to be absolutely useless to work with lengthy texts or large corpora.

Both TF-IDF and Yake! do seem to capture the "aboutness" of the corpus and the differences between time frames are evident: keywords in week 2 capture the geographical origin of the virus as well as the alarm generated by the outbreak, keywords in week 31 include several references to the British government relief initiatives, and keywords in week 85 are mostly about COVID-19 tests and vaccines.

Table 4.5 Unsupervised keyword extraction methods (U.K. Week 85)

TF-IDF	*Yake!*	*RAKE*
free pcr covid	PCR Covid tests	sends kashmir
tests travel sign	COVID	@ lucygrievevet
government provide free	free PCR covid	xi jinping drakeford
uk government provide	Covid fucking covid	versus 390 unvaxinated
provide free pcr	PCR Covid	thingie mi bob
travel sign petition	Covid tests	thankyounhs ♡ xxx
travel sign	long covid	teamearlychildhood acc freaks
pcr covid tests	Covid vaccine	steffiegregg steffie gregg
covid tests travel	covid deaths	spelling errors …)
vaccinated	Covid pandemic	smelly dirty hippies
free pcr	positive Covid test	slugs ate brassicas
tests travel	pandemic	sg adverts galore
government provide	people	select cttee investigations
provide free	Covid cases	professor andrew watterson
pcr covid	Covid vaccination	preparatory communications begin
pcr	lockdown	phdlife raheem sterling
ve finally singing	NHS Covid Pass	paint expressive flowers
given scenes	NHS Covid test	mayflower400 diy audax
come mean given	catch Covid	jamia masjid bilal
song belong written	covid PCR test	insular damaging viewpoint

There are some important differences, however. Yake! captures some relevant topics that TF-IDF does not (e.g. “lockdown”, “social distancing” in week 31, “long covid” in week 85). Similarly, Yake! takes into account features such as case and part of speech, thus clearly favouring noun phrases and capitalized words, whereas TF-IDF returns many syntactically irrelevant word sequences (e.g. “covid tests travel”, “support help push”, “come mean given”). Thus, Yake! appears to offer the best performance in terms of quality, although not so in terms of computational efficiency, as it takes as much as seven times longer to run, thus requiring much more computing power.

When comparing these results with those returned by the reference-corpus method using Sketch Engine, the efficiency should not be taken into account, as this online platform indexes all corpora, and therefore word frequencies (the only feature it uses to identify keywords) are calculated beforehand. Also, being online, time delays are possible due to server load and network issues.

Table 4.6 shows the results for week 2, Table 4.7 for week 31, and Table 4.8 for week 85; as before, the tables include the top 20 keywords, but since Sketch Engine returns two different lists of single and multi-word units with different scores, it is not possible to offer a properly ranked merged list, the top ten single-word items and the top 10 multi-word items are listed separately. In all three tables, two sets of results are shown: using a general-language corpus as reference (enTenTen21), and using the rest of the focus corpus as reference.[10]

Generally speaking, keyword sets extracted using enTenTen21 as reference corpus are rather in line with those extracted by unsupervised methods, referencing the topics in each of the time periods. The main differences are those that are caused by the low frequency of certain words. For example, "ighalo" is in reference to Manchester United's footballer Odion Ighalo, who made the headlines when he was isolated from the rest of the team as a precaution after his return from China in the early stages of the pandemic.

Table 4.6 Reference corpus keywords extraction (U.K. Week 2)

Single words (RC: enTenTen21)	*Multi-words (RC: en-TenTen21)*	*Single words (RC: RoC)*	*Multi-words (RC: RoC)*
wuhan	corona virus	kobe	global health emergency
coronavirus	coronavirus outbreak	brize	coach driver
corona	global health emergency	arrowe	arrowe park hospital
wirral	health emergency	ighalo	health emergency
coronaviru	coronavirus case	wirral	high sense
arrowe	bbc news	huawei	high sense of responsibility
brize	case of coronavirus	horseman	surrounding country
ighalo	coronavirus fear	norton	horseman coach
quarantine	coach driver	bryant	wuhan flight
outbreak	arrowe park hospital	evacuation	sense of responsibility

[10] The full set of results is provided in the book's repository in CSV format.

Table 4.7 Reference corpus keywords extraction. (U.K. Week 31)

Single words (RC: enTenTen21)	*Multi-words (RC: en-TenTen21)*	*Single words (RC: RoC)*	*Multi-words (RC: RoC)*
lockdown	government covid support	dwayne	push for a parliamentary debate
covid	push for a parliamentary debate	micro	government covid support
covid19	micro business	zante	parliamentary debate
distancing	parliamentary debate	parliamentary	micro business
retweet	social distancing	welch	negative multiple time
corona	covid test	pattinson	dwayne johnson
coronavirus	post lockdown	ftfc	review need
pre-covid	local lockdown	two-decade	review need for social distancing
scaremongering	bbc news	dsa	private island
post-lockdown	global pandemic	Tissier	coronavirus stat

Table 4.8 Reference corpus keywords extraction. (U.K. Week 85)

Single words (RC: enTenTen21)	*Multi-words (RC: en-TenTen21)*	*Single words (RC: RoC)*	*Multi-words (RC: RoC)*
covid	covid test	minaj	jodie comer
jab	covid vaccine	comer	nicki minaj
lockdown	covid passport	jodie	stephen graham
vax	covid jab	nicki	uk covid-19 child
vaccinate	long covid	trinidad	lost summer
pre-covid	covid death	cartel	symptom list
unvaccinated	care home	governement	mel morris
minaj	covid pass	reshuffle	uc cut
tory	covid restriction	tobago	full chamber
bollock	covid case	jody	vaccine dispersal

The rest-of-corpus (RoC) method, on the other hand, returns a larger proportion of proper names, both in the single- and multi-word lists, and, perhaps counterintuitively, seems to be less appropriate than the "general-language" reference-corpus method, as it does not highlight the specific topics of the time periods. It is also surprising that the term "PCR" is not

listed in the top ten keywords, as is the case in the set extracted by both TF-IDF and Yake! In fact, it is in position 28 in the score-ranked list.[11]

4.2.3 Graph-Based Approaches

Graph-based approaches are a kind of unsupervised algorithms, since they also rely solely on the text itself. Graphs are data structures that consist of *vertices* (or *nodes*) joined by *edges*. They are used for many practical applications, such as navigation and route planning, to calculate the shortest path, or network flow analysis, including social networks, where nodes represent people and edges represent relationships or interactions; thus, they are a versatile tool that can be used in computer science, engineering, sociology, or biology.

Graphs have been used in NLP for text summarization, as they can identify the most relevant sentences in a text, and keyword extraction, as they are able to extract the most "relevant" words and phrases in a text, which is why they are also referred to as *ranking* algorithms. The most popular implementation is TextRank (Mihalcea and Tarau 2004), which is inspired by PageRank (Brin and Page 1998), the revolutionizing web search algorithm developed by the creators of Google that was directly responsible for the company's initial success. Search results using PageRank vastly improved on existing methods used by other search engines, which were based on keyword matching and meta tags.

Just like PageRank treats the Web as a vast graph, with web pages as nodes and hyperlinks as edges, so does TextRank, where words are treated as nodes and edges represent co-occurrence within a text window (span) of a certain size. The type of edge, however, is different: whereas web pages are linked by directed graphs, TextRank uses undirected, weighted graphs. The weights determine the "importance" of words and they are calculated by a voting system; each word will "vote" for the words within its window, and the weight of each word depends not only on the number of votes but also on the importance of the words voting for it. The voting system is recursive, in such a way that words that are frequently connected to other high-ranking words get higher scores too, which helps identify those words that truly capture the essence of the text. The same principle is used for summarization, where sentences rather than words are

[11] All 12 full lists of keywords are included in the book's repository.

edges, and each sentence "votes" for other sentences according to their similarity, which is calculated by word overlap (the number of words that sentences have in common).

Experiment: Graph-Based vs Reference-Corpus Keyword Extraction
The experiment that follows aims to compare results from two keyword extraction methods: the TextRank algorithm and the reference-corpus method.

I use the PyTextRank (Nathan 2016) library, which is a Python implementation of the original proposal by Mihalcea and Tarau (2004) in the form of a SpaCy extension. SpaCy (Honnibal et al. 2020) is a powerful, general-purpose NLP toolkit that can be used for many high-level tasks, including part-of-speech tagging, dependency parsing, semantic analysis using word embeddings, named entity recognition, and many others. It also allows for third-party add-ons and extensions, as is the case of PyTextRank.

As for the corpus, the aim is to extract keywords from the 1% sample of the full CCTC, which consists of over 11 million tweets and 300 million words (see Table 3.4). Analysing text with SpaCy involves certain limitations, as a SpaCy "doc" object, in which text is analysed in a pipeline, needs to be created for each single text. Since our texts are tweets and there are many millions of them, this may quickly become extremely slow. Thus, a decision was made to optimize the script to analyse tweets in batches of 100, which does not impact TextRank's performance, as document size does not affect its results (Mihalcea and Tarau 2004, 407). Frequencies of items were multiplied by the mean of the magnitudes of the tweets in the batch, as specified by the tweet's frequency (n, see Sect. 3.2.3); although this may not be entirely accurate, it is an acceptable approximation for the purpose of this experiment.

TextRank returns a large number of keyword candidates, sorted by score (it literally ranks every word that is not a stop-word). The script allows the specification of a minimum score as a cut-off point, which was set at 0.010 after some experimentation, and also a minimum frequency within batches, which was set to 1. The keywords in each batch were aggregated by averaging their scores and adding their frequencies. For this experiment, data was extracted from the daily files and results were subsequently aggregated by month.

Unlike Sketch Engine, TextRank makes no distinction between single-word and multi-word keywords, but in order to facilitate comparison of

results, the extraction script automatically makes two subsets by checking for the presence of spaces. Similarly, 1,000 single-word and 1,000 multi-word keywords were extracted and kept in the monthly aggregated files, as this is the maximum number of keywords offered by Sketch Engine. The result is therefore 96 sets of 1,000 items each (12 months * 2 keyword types * 2 extraction methods).

PyTextRank does not take any parameters, so a number of parameters were coded in the script itself to filter and improve results. These include the following:

- Case-sensitive: the script allows to have keywords analysed in either case-sensitive mode or not. It was "off" for this experiment.
- Minimum rank: the score threshold below which candidate keywords are to be discarded (0.010 in this experiment).
- Exclusion list: a list of banned words to be ignored. These include common words in tweets, such as the names of week days and months, and certain stopwords, quantifiers, numerals, etc. Also Twitter mentions (handles).
- Allow entities in keywords: having this option set to "false" will discard keywords consisting of or containing entities. This relies on SpaCy's built-in entity recognition capabilities. Since we aim to compare results with Sketch Engine, this setting was set to "true" for this experiment.

Results were saved as monthly CSV files.

To extract the keywords with Sketch Engine, the XML version of the corpus was used (see Sect. 3.5). A subcorpus was created on Sketch Engine for each month of the two years that the corpus covers based on the metadata we embedded in the XML exported files, and extracted the top 1,000 keywords and keyphrases for each month. As before, the reference corpus used was enTenTen21. For keyword extraction, Sketch Engine makes a distinction between *keywords* (single-word items) and *terms* (multi-word items).

The analysis of results is described in the next section. The full set of extracted keywords by both systems can be found in the book's repository.

4.3 Comparing Keyword Sets

As we have seen, comparing the quality of the results of keyword extraction, i.e. judging how accurately a set of words qualifies the "aboutness" of a corpus, is a rather subjective task (Gabrielatos et al. 2012). This is the approach employed to analyse the results of the previous experiments in this book: presenting ranked lists of items and assessing their "quality" in a rather subjective manner applying certain—rather vague—criteria. Although this is rather inevitable as no clear objective criteria exist, here I introduce a quantitative—and therefore more objective—method to assist in the comparison of large sets of keywords.

Comparing two sets of keywords obtained through two different methods is not easy, but in this case things are further complicated by the scale of the data. Qualitatively comparing 48 pairs of sets of 1,000 items each is not practical or even worth the tremendous work involved. Quantitative methods can help to attain a global overview of the data and then use some other methods that can facilitate the manual, qualitative analysis of a few cases.

In the analysis that follows I use set operations (intersection and difference) as a quantitative aid to compare results and visualize results using Venn-type diagrams, generated automatically from the data, as well as tables with lists based on intersection and difference. The script used to generate these data and graphs takes two score-ranked lists of keyword items (whether single or multi-words) in CSV format, where the first column contains the items themselves and any number of data columns may be present. All lists in the sets contain the top 1,000 keywords generated by TextRank and Sketch Engine, extracted using the parameters described in the previous section, but the scripts allow to specify a cut-off point so that only the top n items are taken to calculate their intersection and difference. For each pair, which in this case are each of the 24 months sampled in the corpus, the script generates three elements:

1. Counts of the intersection to later obtain the statistics, as presented in Table 4.9. These are printed at runtime and saved as text files.
2. A Venn diagram of word clouds that can visually help understand the similarities and differences between sets. These are only generated when the top 100 items or less are selected, as larger lists can hardly be readable in this format.

3. An HTML table containing alphabetically sorted lists of words in the intersection and difference.

Table 4.9 quantitatively summarizes the results of comparing the keyword sets generated by each of the two methods. It contains the monthly intersection figures for the top 30, 50, and 100 keywords.[12] After discussing these results, three months in different stages of the time frame will be analysed in detail using the lists of words and Venn diagrams generated by the script, as, ultimately, subjective, qualitative analysis is necessary to assess how well different sets of keywords tell us about the "aboutness" of a corpus.

There is clearly a significant difference in the intersections percentages between single words and multi-words ($M = 41.36\%$ for the former; $M = 26.24\%$ for the latter). The reason for this, for which ample evidence will be available in the lists of keywords presented below, is that whereas for single words Sketch Engine allows users to specify which attribute to use for the calculations, this is not the case for multi-word items (see Fig. 4.1) and, although nothing is mentioned in the user's manual or the interface, it is evident that it uses lemmas, not word forms.

Thus, Sketch Engine will retrieve "coronavirus case", "covid death", and "health expert" rather than "coronavirus cases", "covid deaths", and "health experts", which is what TextRank will retrieve, as no lemmatization is performed. A possibility to equalize this situation is to lemmatize the corpus prior to keyword extraction with TextRank, but this is not a good idea, as lemmatization does have an enormous impact on many text processing tasks, especially part-of-speech tagging.

Other than that, percentages of intersections are rather consistent across months and top-n sets within each category ($SD = 0.0093$ for single words, $SD = 0.0238$ for multi-words), which suggests that both keyword extraction systems follow consistent patterns—and deliver similar results—reliably.

As in the previous experiment, I will now analyse in detail the results corresponding to three time periods—months this time—in different stages of the timeframe that the corpus covers. The months chosen

[12] The book's repository also contains the data for the top 500 and 1,000 keyword sets.

Table 4.9 Monthly keywords intersections (TextRank ∩ Sketch Engine)

	Top 30 keywords		*Top 50 keywords*		*Top 100 keywords*	
	SW	*MW*	*SW*	*MW*	*SW*	*MW*[13]
2020-01	15	10	25	16	48	31
2020-02	13	11	22	20	46	40
2020-03	12	9	25	18	49	38
2020-04	11	9	21	17	44	33
2020-05	10	6	20	16	41	31
2020-06	12	5	22	14	43	31
2020-07	11	5	20	9	47	25
2020-08	12	8	20	16	41	29
2020-09	13	5	22	15	45	28
2020-10	12	10	22	13	45	28
2020-11	12	8	19	14	41	29
2020-12	14	8	22	12	42	28
2021-01	13	7	21	12	40	32
2021-02	13	7	22	11	40	27
2021-03	12	7	21	9	38	31
2021-04	15	5	19	14	44	21
2021-05	15	8	19	15	41	26
2021-06	14	6	19	12	47	28
2021-07	13	5	21	10	42	22
2021-08	11	5	20	10	45	25
2021-09	13	5	19	10	45	25
2021-10	12	7	19	11	41	20
2021-11	13	7	20	9	47	21
2021-12	14	7	21	12	44	32
Mean	12.71	7.08	20.88	13.13	43.58	28.38
Mean%	42.36%	23.61%	41.75%	26.25%	43.58%	28.38%

include the weeks in the previous experiment, but it must be remembered that, apart from the difference in time length, this sample includes tweets from all countries, with no distinction among them, which makes it more difficult to identify specific topics. It must also be borne in mind that the vast majority of tweets are generated in the United States (see Fig. 3.4 in Sect. 3.4). For each month I will be using the top 50 sets, Venn diagrams for single words and tables for multi-words. In all Venn

[13] SW: single words. MW: multi-words.

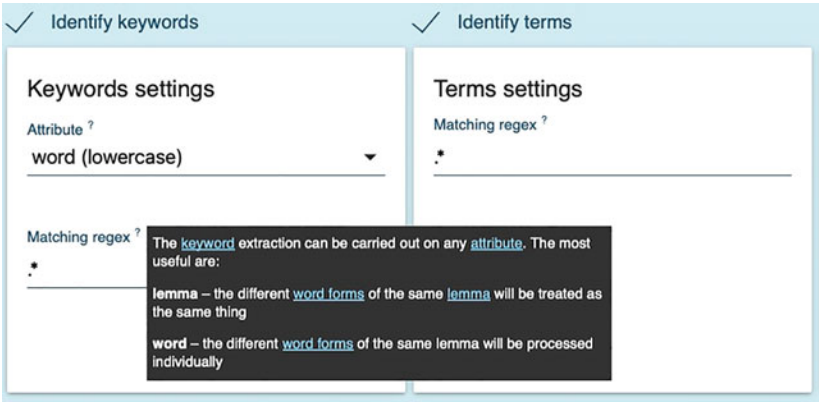

Fig. 4.1 Sketch Engine's attribute selection for keyword extraction

diagrams TextRank's (TR) keywords are displayed on the left and Sketch Engine's on the right.

Figure 4.2 shows the Venn diagram for single-word keywords corresponding to February 2020. The intersection in this case is 44%, which means that almost half the keywords extracted by both systems are the same.

The intersection clearly includes the main words associated with the events in this early stage of the pandemic. The U.S. bias in the corpus can already be seen as the intersection includes references to the American Center for Disease Control and President Donald Trump. It also includes references to the source of the disease ('wuhan', 'china', 'chinese') and other Asian countries ('korea', 'japan'), the early reference to the disease as 'corona', the 'outbreak', and the comparison with a regular 'flu'. As for the differences, TextRank includes a few words that make little sense, as they are too general ('things', 'weeks', 'days', 'years', 'home', 'world'), but the rest are informative and highlight relevant. Sketch Engine's keywords tend to be more specific because the method is based on significant differences in frequencies from a reference corpus, but it also includes irrelevant words, such as 'via' or 'breaking' (both commonly used in Twitter news), 'amid' or 'hong' and 'kong' (the two words in the multi-word unit "Hong Kong").

Here we also find an example of a big problem that Sketch Engine has when dealing with social media text: it is unable to process emojis

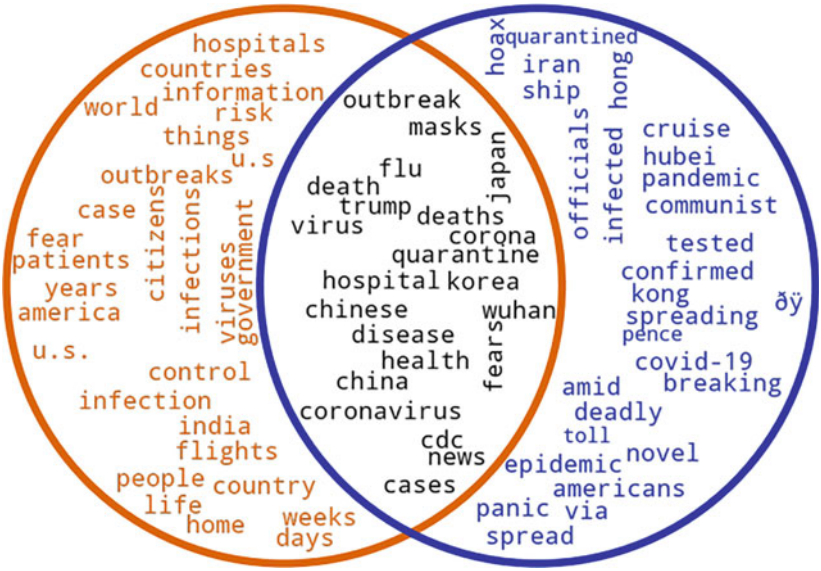

Fig. 4.2 Top 50 single-word keywords for February 2020 (TR left, SE right)

correctly. Even if the corpus is uploaded in correct UTF-8 encoding, the application displays certain Unicode characters instead of the corresponding emoji. The sequence 'ðŸ', specifically corresponds to the *sad emoji*, as evidenced by a concordance search of the 'keyword'. For example, the sentence in (18), extracted from a Sketch Engine concordance corresponds to the tweet shown in (19).

18. <s>One depressing thing about COVID (but perhaps necessary) is finding out ppl you thought were smart are just…not **ðŸ** < /s >
19. {"text": "One depressing thing about COVID (but perhaps necessary) is finding out ppl you thought were smart are just…not ◉", "user": "BillMonty_", "date": "Tue Dec 28 21:34:36 + 0000 2021", "id": "1475943287127171079", "n": 96}

This sequence ('ðŸ') is found ranking high in literally all monthly single-word keyword sets generated by Sketch Engine.

The differences in multi-word keyword extraction are also interesting. Both systems retrieve, in total, 21 two-word compounds where the first word is "coronavirus", which are broken down as follows:

- Retrieved by both systems: 'death', 'infection'', 'outbreak', 'patient', 'spread', 'update', 'vaccine'.
- Only in TextRank: 'cases', 'concerns', 'crisis', 'deaths', 'disease', 'fears', 'impact', 'infections', 'patients', 'quarantine', 'threat', 'quarantine'.
- Only in Sketch Engine: 'epidemic', 'fear', 'response'.

Some of these, however, are cases where both systems actually extracted the same phrases but were not included in the intersection due to Sketch Engine's using lemmas rather than words: both 'coronavirus deaths' and 'coronavirus fears' are included in Sketch Engine in singular because of lemmatization. All of these cases have been marked in bold (Table 4.10).

Table 4.10 Top 50 multi-word keywords for February 2020

Intersection	chinese people, corona virus, coronavirus case, coronavirus infection, coronavirus outbreak, coronavirus patient, coronavirus spread, coronavirus update, coronavirus vaccine, cruise ship, death toll, first case, hong kong, hubei province, new coronavirus, novel coronavirus, public health, south korea, virus outbreak, wuhan coronavirus
Only in TextRank	china coronavirus, china virus, china ■, chinese authorities, chinese officials, communist china, **confirmed cases**, coronavirus cases, coronavirus concerns, coronavirus crisis, **coronavirus deaths**, coronavirus disease, **coronavirus fears**, coronavirus impact, coronavirus infections, coronavirus patients, coronavirus quarantine, coronavirus threat, deadly coronavirus, **face masks**, **health officials**, infected people, mainland china, medical supplies, **new cases**, north korea, social media, wuhan china, wuhan city, wuhan virus
Only in Sketch Engine	case of coronavirus, chinese doctor, chinese government, communist party, **confirmed case**, **coronavirus death**, coronavirus epidemic, **coronavirus fear**, coronavirus response, diamond princess cruise, **face mask**, first coronavirus, **health official**, infectious disease, medical worker, mike pence, mortality rate, **new case**, new virus, other country, president trump, press conference, spread of coronavirus, stock market, supply chain, suspected case, travel ban, trump administration, washington state, world health organization

Some topics are highlighted by keywords in both systems but with some differences. For example, there are words related to the event involving the Diamond Princess cruise ship, which was quarantined off the coast of Japan for two weeks in February 2020, so both sets include 'cruise ship', but only Sketch Engine includes the actual name of the ship ('diamond princess cruise'), which helps identify the specific event.

Finally, we can see how TextRank manages emojis correctly treats them just like any other word ('china 🇨🇳').

Figure 4.3 displays the Venn diagram corresponding to the month of August 2020. First, we find several more examples of Sketch Engine taking as keywords several unreadable characters: 'ðŸ', 'à', 'â', 'u'. We also find the same issues affecting both or either of the extraction systems (i.e. more frequent terms in TextRank, rarer words in Sketch Engine, Twitter-specific words in the latter). Also, in addition to the general words referring to the disease, the intersection includes some words in reference to important repercussions of the pandemic in some countries, specifically India, in relation to official exams ('students', 'exam', 'exams', 'tests'). TextRank does not give us any more clues regarding this issue, but Sketch Engine does: both 'jee' and 'neet' refer to official exams: JEE (Joint Entrance Examination) is India's entrance exam to several engineering degrees, which took place on September 1, 2020, and NEET is India's medical admission test (National Eligibility cum Entrance Test). During August there were doubts that these important exams could actually be conducted and which measures would apply if they were.

The reason that these keywords are picked up by the reference-corpus method and not the graph method is, again, frequency, which is relatively low in the focus corpus but high relative to the reference corpus. TextRank does retrieve these two words, but they are ranked low (200th position for 'jee' and 205th for 'neet') (Table 4.11).

As for multi-word keywords, here we find again many cases (14, marked in bold) that should be part of the intersection, as they are plurals that have been lemmatized by Sketch Engine. On the other hand, it is surprising to see how both systems include Donald Trump, but only TextRank includes Dr. Fauci (which Sketch Engine ranks in position 614).

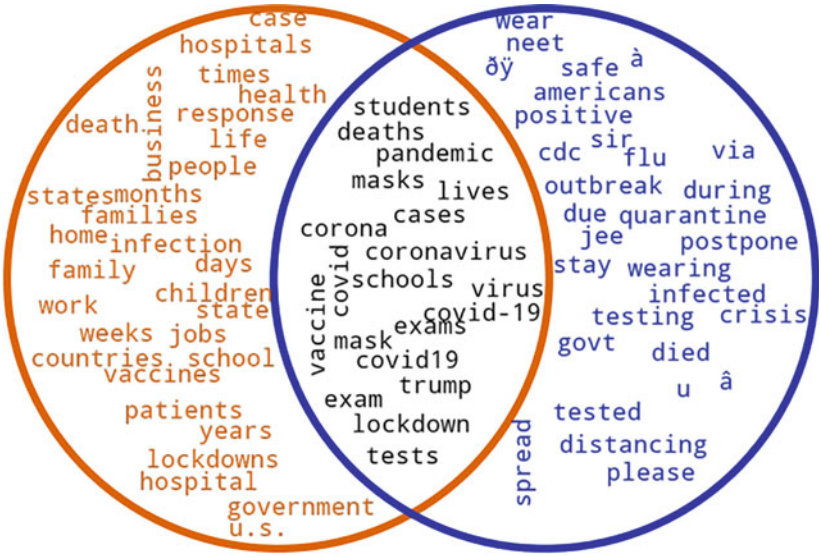

Fig. 4.3 Top 50 single-word keywords for August 2020 (TR left, SE right)

Table 4.11 Top 50 multi-word keywords for August 2020

Intersection	aged care, corona virus, coronavirus pandemic, coronavirus relief, coronavirus vaccine, covid relief, covid-19 pandemic, death toll, donald trump, global pandemic, herd immunity, mental health, president trump, public health, social distance, social distancing
Only in TextRank	**active cases**, black people, **care homes**, climate change, **confirmed cases,** corona cases, coronavirus cases, **coronavirus deaths**, coronavirus infections, **covid cases**, **covid deaths**, covid pandemic, **covid patients**, covid times, **covid19 cases**, covid19 pandemic, dr. fauci, election day, **face masks**, health care, high risk, **loved ones**, **new cases**, next week, next year, **nursing homes**, physical distancing, **positive cases**, public transport, small businesses, social media, social security, students life, young people
Only in Sketch Engine	**active case**, **care home**, china virus, **confirmed case**, **coronavirus case**, **coronavirus death**, coronavirus outbreak, **covid case**, **covid death**, **covid patient**, covid test, covid vaccine, **covid-19 case**, covid-19 death, covid-19 patient, covid-19 test, covid-19 vaccine, death rate, **face mask**, joe biden, **loved one**, middle of a pandemic, **new case**, new coronavirus, new zealand, **nursing home**, pandemic response, pandemic situation, **positive case**, second wave, stay home, trump administration, wearing mask, white house

Only Sketch Engine includes Joe Biden, but in TextRank it is in 57th position. Also, the two systems seem to have some advantages and disadvantages over the other; TextRank seems to pick up on the social and economic impact of the pandemic ('small businesses', 'social security', 'students life'), whereas Sketch Engine includes some important keywords for this stage of the pandemic, such as 'second wave' and 'stay home'.

Finally, for the month corresponding to week 85 (September 2021), TextRank seems to better extract the keywords specific to this time period. This can be seen in the top single-word items (Fig. 4.4), as it includes terms like 'delta', 'booster', 'pfizer', and 'immunity', all of which are ranked lower in TextRank's list (in position 54, 224, 150, and 79, respectively).

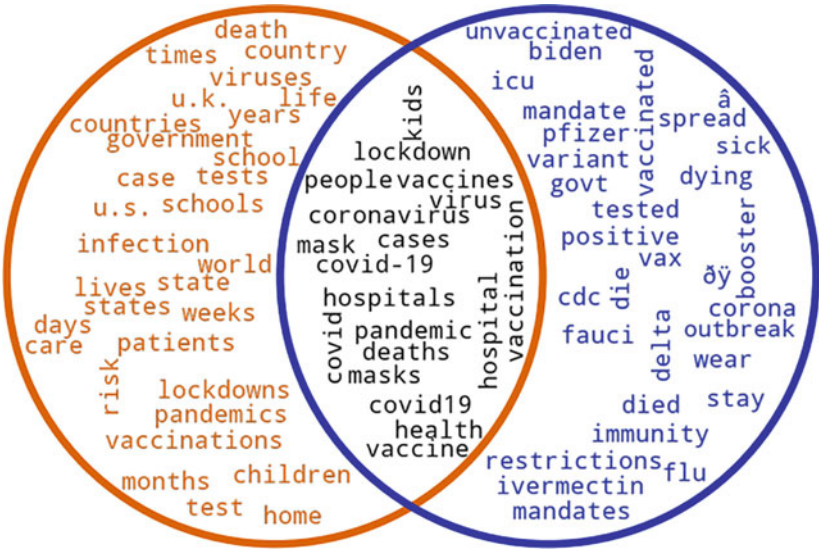

Fig. 4.4 Top 50 single-word keywords for September 2021 (TR left, SE right)

Table 4.12 Top 50 multi-word keywords for September 2021

Intersection	covid pandemic, covid test, covid vaccination, covid vaccine, health care, long covid, public health, unvaccinated people, vaccinated people, vaccine mandate
Only in TextRank	**active cases**, catching covid, contracting covid, coronavirus vaccines, covid cases, covid deaths, covid hospitalizations, covid infections, covid mandates, covid misinformation, covid numbers, covid passports, covid patients, covid protocols, covid relief, covid restrictions, covid rules, covid testing, covid tests, covid vaccinations, covid vaccines, covid visualizations, **covid-19 vaccines**, dr. fauci, face masks, health care workers, health workers, healthy people, **mask mandates**, new cases, new covid cases, next week, next year, severe covid, unvaccinated covid patients, vaccine hesitancy, vaccine mandates, **vaccine passports**, vaccine requirements, young people
Only in Sketch Engine	**active case**, booster shot, care worker, covid case, covid death, covid infection, covid jab, covid passport, covid patient, covid restriction, covid shot, covid-19 case, covid-19 death, covid-19 pandemic, covid-19 patient, covid-19 vaccination, **covid-19 vaccine**, death rate, death toll, delta variant, global pandemic, healthcare worker, icu bed, immune system, joe rogan, last year, long term, loved one, many people, **mask mandate**, mental health, natural immunity, new case, panic buying, side effect, social distancing, vaccination rate, vaccine dose, **vaccine passport**, wearing mask

Similarly, in multi-word keywords, although both systems include the stage-specific term 'long covid', only Sketch Engine offers others, such as 'booster shot', 'delta variant', and 'mental health' (Table 4.12).

4.4 Keyword Extraction Using Word Embeddings

Transformers-based Large Language Models (LLMs) have proved to be incredibly useful in many tasks, not just language generation, including, of course, keyword extraction. This is because the word embeddings that are used to create LLMs do capture the semantics of the words and phrases that make up a text, as well as the text as a whole. The KeyBERT (Grootendorst 2020) keyword extraction tool used in the last experiment of this chapter is based on this very basic principle, as it calculates keyness by measuring the similarity between the individual words and phrases of a text and the text itself; this is a keyword extraction method that was first proposed by Sharma and Li (2019). Other keyword extractors based on word embeddings are available, such as EmbedRank (Bennani-Smires et al. 2018).

As is common in word embeddings, the metric it uses is cosine similarity. The approach taken is fairly simple: words or n-grams with a higher cosine similarity to that of the text as a whole will rank higher than those more distant. This is a simple, yet powerful keyword extraction method that is easy to implement and can be customized by using different embeddings, probably the most determining factor.

4.4.1 Experiment: Comparing Keywords from Two Countries Using KeyBERT

In the following experiment, I use KeyBERT[14] to extract keywords and keyphrases in the range 2 to 3 from the geotagged version of the CCTC in order to compare two countries: Australia and India. I use the 25% sample of each of these countries. The Australian subcorpus is made up of 69,685 tweets (about 1.7 million words); the Indian sample is larger, at 170,974 tweets (about 4.39 million words).

KeyBERT takes a number of parameters that can have a strong impact on the results. The most important is obviously the language model that is used to compute the similarity between words/phrases and the whole document. In this experiment I use the default model (*all-MiniLM-L6-v2*), a sentence-transformers model that is compact in size yet powerful for many applications.[15] There are two very useful parameters that KeyBERT can take aimed at diversifying results. This is in order to avoid sets of different but very similar keywords and, especially, keyphrases, as we have seen in previous sets ('coronavirus', 'corona virus', 'covid shot', 'covid jab'). Again, the tool leverages the power of word embeddings and cosine similarity to obtain a measurement of the similarities between the results obtained and discard those that display a high level of similarity. There are two such parameters: Max Sum Distance (which was set to *true*) and Maximal Marginal Relevance, which was set to 0.7 (high diversity).

[14] KeyBERT is distributed as a Python package. Instructions on installation and use can be found at https://maartengr.github.io/KeyBERT/index.html.

[15] A thorough description of this language model can be found at https://huggingface.co/sentence-transformers/all-MiniLM-L6-v2.

KeyBERT also allows the specification of the length of n-grams that we wish to extract. In order to extract them independently, two runs are necessary, one to extract single-word keywords (*n*-gram range 1–1) and another for multi-word keywords (range 2–3) (Table 4.13).

Eleven of the top 20 single-word keywords are hashtags, although KeyBERT drops the hash sign. Furthermore, although many of them make sense as keywords ('covid19vaccination', 'positivevibes', 'lockdownmelbourne', 'savehospitality'), others appear to be rather irrelevant; for example, both 'mugsareqldracing' and 'scottymissingagain' occur exactly once—Examples (20) and (21)—in the corpus and bear no relationship to any relevant topic.

Table 4.13 KeyBERT results for July 2021 Australia[16]

Keywords	*Score*	*Keyphrases*	*Score*
covid19vaccination	0.565	australia covid deaths	0.639
mugsareqldracing	0.339	disgusting assembletheguillotines auspol	0.395
bleak	0.271	healthcare biosecurity citizens	0.342
caringbah	0.267	sydney buck naked	0.304
gladyscovidspreaders	0.253	just like blm	0.297
positivevibes	0.245	pandemic snack time	0.297
wuhanvirus	0.239	doherty warned patients	0.291
lockdownmelbourne	0.234	comments skynews pretending	0.284
jfc	0.234	vaccine takes long	0.282
notsafeforwork	0.233	morrison undermined	0.282
coffees	0.233	virus tax return	0.279
sarscov2	0.231	hopefully ease victoria	0.274
xenophobic	0.218	health recorded zero	0.266
coffs	0.211	reconsiders use astrazeneca	0.254
antibodies	0.205	cases uk july	0.253
stateoforigin2021	0.201	gladys corruption idea	0.251
scottymissingagain	0.2	great news general	0.251
trigger	0.199	wollongong2022 rename	0.249
wildlife	0.198	clots national drug	0.248
savehospitality	0.196	visit dutton takes	0.242

[16] The book's repository includes the full lists of keywords and keyphrases for all months corresponding to the Australian, South African, and Indian subcorpora.

20. {“country_code”: “AU”, “timestamp”: “Fri Jul 02 10:14:36+0000 2021”, “user”: “mugspunting”, “id”: “1410904734341287937”, “text”: “Anyone else’s #Lockdown look a bit like this? Thanks for the new sponsorship… haven’t workout out any details yet but we’ll get there. #MugsAREqldracing”}
21. {“country_code”: “AU”, “timestamp”: “Tue Jul 20 04:39:58+0000 2021”, “user”: “SullivanCate”, “id”: “1417343502996832257”, “text”: “NSW in lockdown. VIC in lockdown. SA in lockdown. #scottymissingagain”}

Keyphrases do seem to convey more of the topics relevant to the events in the country (‘australia covid deaths’, ‘doherty warned patients’, ‘morrison undermined’, ‘gladys corruption idea’). However, we find the same issue related to the selection of hashtags, or sequences of hashtags; for example, the hashtag ‘#assembletheguillotines’ occurs twice in the sample, and the actual sequence ‘#disgusting #assembletheguillotines #auspol’ occurs once, shown in (22).

22. {“country_code”: “AU”, “timestamp”: “Wed Jul 28 00:59:49+0000 2021”, “user”: “amandajanewd”, “id”: “1420187201594290176”, “text”: “@AustralianLabor you ripped my heart out after the last election and now keep trampling on it. You are the literal worst 😡 #Heartbreaking #Disgusting #assembletheguillotines #auspol”}

Finally, the list of keyphrases is quite obviously the result of processing n-grams rather than syntactically coherent groupings, which results in rather awkward phrases, such as ‘hopefully ease victoria’, ‘health recorded zero’, or ‘visit dutton takes’.

After this initial experiment, it seems that this recent method of keyword extraction, although original in its proposal, still needs a lot of refining and improvement to be a good alternative other well-established methods, specifically the reference-corpus and graph-based methods, which in our tests offered the best results.

The other important criterion that needs to be considered is that of performance. Again, the reference-corpus method is extremely fast and lightweight in terms of computing requirements, as all it takes is a list of pre-calculated word frequencies for the reference corpus, and the

frequencies of the focus corpus, which in the case of an indexed, lemmatized corpus management tool has also been pre-calculated. Most other methods we have explored are more costly in terms of computing power, with the exception of some unsupervised methods (RAKE), whose quality leaves much to be desired.

References

Alessi, Glenn Michael, and Alan Partington. 2020. *Modern Diachronic Corpus-Assisted Language Studies: Methodologies for Tracking Language Change Over Recent Time*. Italia: Mattioli, 1885.

Anthony, Laurence. 2023a. AntConc (Version 4.2.0). Tokyo, Japan: Waseda University.

Anthony, Laurence. 2023b. Common Statistics Used in Corpus Linguistics.

Baker, Paul. 2004. Querying Keywords: Questions of Difference, Frequency and Sense in Keywords Analysis. *Journal of English Linguistics* 32: 346–359. https://doi.org/10.1177/0075424204269894.

Baker, Paul. 2006. *Using Corpora in Discourse Analysis*. A&C Black.

Bennani-Smires, Kamil, Claudiu Musat, Andreea Hossmann, Michael Baeriswyl, and Martin Jaggi. 2018. Simple Unsupervised Keyphrase Extraction using Sentence Embeddings. In *Proceedings of the 22nd Conference on Computational Natural Language Learning*, 221–229. Brussels, Belgium: Association for Computational Linguistics. https://doi.org/10.18653/v1/K18-1022.

Boyce, Bert R., Charles T. Meadow, and Donald H. Kraft. 1994. *Measurement in Information Science: An Information Services Perspective*. Library and Information Science (New York, NY). San Diego, California: Academic Press.

Bondi, Marina. 2010. An Introduction: Perspectives on Keywords and Keyness. In *Keyness in Texts*, ed. Marina Bondi and Mike Scott, 1–18. Studies in Corpus Linguistics. John Benjamins Publishing Company. https://doi.org/10.1075/scl.41.01bon.

Bondi, Marina, and Mike Scott, ed. 2010. *Keyness in Texts*. John Benjamins Publishing Company.

Brezina, Vaclav. 2018. *Statistics in Corpus Linguistics: A Practical Guide*. Cambridge, UK: Cambridge University Press. https://doi.org/10.1017/9781316410899.

Brin, Sergey, and Lawrence Page. 1998. The Anatomy of a Large-Scale Hypertextual Web Search Engine. *Computer Networks* 30: 107–117.

Campos, Ricardo, Vítor Mangaravite, Arian Pasquali, Alípio Mário Jorge, Célia Nunes, and Adam Jatowt. 2018. YAKE! Collection-Independent Automatic Keyword Extractor. *Lecture Notes in Computer Science*. https://doi.org/10.1007/978-3-319-76941-7_80.

Egbert, Jesse, and Doug Biber. 2019. Incorporating Text Dispersion into Keyword Analyses. *Corpora* 14: 77–104. Edinburgh University Press. https://doi.org/10.3366/cor.2019.0162.

El-Beltagy, Samhaa R., and Ahmed Rafea. 2009. KP-Miner: A Keyphrase Extraction System for English and Arabic Documents. *Information Systems* 34: 132–144. https://doi.org/10.1016/j.is.2008.05.002.

Gabrielatos, Costas, Tony McEnery, Peter J Diggle, and Paul Baker. 2012. The Peaks and Troughs of Corpus-Based Contextual Analysis. *International journal of corpus linguistics* 17: 151–175. John Benjamins.

Gabrielatos, Costas. 2018. Keyness analysis: Nature, metrics and techniques. In *Corpus Approaches to Discourse: A Critical Review*, ed. C. Taylor and A. Marchi, 225–258. Oxford: Routledge.

Grootendorst, Maarten. 2020. KeyBERT: Minimal keyword extraction with BERT. *Zenodo*. https://doi.org/10.5281/zenodo.4461265.

Honnibal, Matthew, Ines Montani, Sofie Van Landeghem, and Adriane Boyd. 2020. SpaCy: Industrial-strength natural language processing in python. *Zenodo*. https://doi.org/10.5281/zenodo.1212303.

Hulth, Anette. 2004. Combining Machine Learning and Natural Language Processing for Automatic Keyword Extraction. Stockholm, Sweden: Stockholm University, Faculty of Social Sciences, Department of Computer and Systems Sciences.

Hunt, Daniel, and Kevin Harvey. 2015. Health Communication and Corpus Linguistics: Using Corpus Tools to Analyse Eating Disorder Discourse Online. In *Corpora and Discourse Studies: Integrating Discourse and Corpora*, ed. Paul Baker and Tony McEnery, 134–154. Palgrave Advances in Language and Linguistics. London: Palgrave Macmillan UK. https://doi.org/10.1057/9781137431738_7.

Jakubíček, Miloš, Adam Kilgarriff, Vojtěch Kovář, Pavel Rychlý, and Vít. Suchomel. 2013. The TenTen Corpus Family. In *7th International Corpus Linguistics Conference CL 2013*, 125–127. UK: Lancaster.

Johnson, Sally, and Astrid Ensslin. 2006. Language in the News: Some Reflections on Keyword Analysis Using Wordsmith Tools and the BNC. *Leeds Working Papers in Linguistics and Phonetics* 11.

Kilgarriff, Adam. 2009. Simple Maths for Keywords. In *Proceedings of Corpus Linguistics Conference (CL 2009)*, ed. M. Mahlberg, V. González-Díaz, and C. Smith. University of Liverpool, UK.

Kilgarriff, Adam. 2012. Getting to Know Your Corpus. In *Text, Speech and Dialogue*, ed. Petr Sojka, Aleš Horák, Ivan Kopeček, and Karel Pala, 3–15. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer. https://doi.org/10.1007/978-3-642-32790-2_1.

Kilgarriff, Adam, Vít Baisa, Jan Bušta, Miloš Jakubíček, Vojtěch Kovář, Jan Michelfeit, Pavel Rychlý, and Vít Suchomel. 2014. The Sketch Engine: Ten Years On. *Lexicography* 1: 7–36.

Mahlberg, Michaela. 2007. *Corpus Stylistics: Bridging the Gap Between Linguistic and Literary Studies*.

Marchi, Anna. 2018. Dividing Up the Data: Epistemological, Methodological and Practical Impact of Diachronic Segmentation. In *Corpus Approaches to Discourse*. Routledge.

Matoré, Georges. 1953. *La méthode en lexicologie: domaine français*. M. Didier.

Mihalcea, Rada, and Paul Tarau. 2004. TextRank: Bringing Order into Text. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, 404–411. Barcelona, Spain: Association for Computational Linguistics.

Nathan, Paco. 2016. *PyTextRank, a Python Implementation of TextRank for Phrase Extraction and Summarization of Text Documents*. Derwen.

Nomoto, Tadashi. 2023. Keyword Extraction: A Modern Perspective. *Sn Computer Science* 4: 92. https://doi.org/10.1007/s42979-022-01481-7.

Pedregosa, F., G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, et al. 2011. Scikit-Learn: Machine Learning in Python. *Journal of Machine Learning Research* 12: 2825–2830.

Rose, Stuart, Dave Engel, Nick Cramer, and Wendy Cowley. 2010. Automatic Keyword Extraction from Individual Documents. In *Text Mining*, 1–20. Wiley. https://doi.org/10.1002/9780470689646.ch1.

Scott, Mike. 1996. *WordSmith Tools*. Oxford: Oxford University Press.

Scott, Mike. 1997. PC Analysis of Key Words—And Key Key Words. *System* 25: 233–245. https://doi.org/10.1016/S0346-251X(97)00011-0.

Scott, Mike. 2010. Problems in Investigating Keyness, or Clearing the Undergrowth and Marking Out Trails…. In *Keyness in Texts*, ed. Marina Bondi and Mike Scott, 43–57. Amsterdam/Philadelphia: John Benjamins Publishing Company.

Scott, Mike. 2022. *WordSmith Tools*. Stroud: Lexical Analysis Software.

Scott, Mike, and Christopher Tribble. 2006. *Textual Patterns. Key Words and Corpus Analysis in Language Education*. Studies in Corpus Linguistics 22. Amsterdam/Philadelphia: John Benjamins Publishing Company.

Sharma, Prafull, and Yingbo Li. 2019. *Self-Supervised Contextual Keyword and Keyphrase Retrieval with Self-Labelling*. Preprints. https://doi.org/10.20944/preprints201908.0073.v1.

Sinclair, John McHardy. 1991. *Corpus, Concordance, Collocation*. Oxford: Oxford University Press.

Spärck Jones, Karen. 1972. A Statistical Interpretation of Term Specificity and Its Application in Retrieval. *Journal of Documentation* 28: 11–21.

Stubbs, Michael. 2010. Three Concepts of Keywords. In *Keyness in Texts*, ed. Marina Bondi and Mike Scott, 21–42. Amsterdam/Philadelphia: John Benjamins Publishing Company.

Sun, Chengyu, Liang Hu, Shuai Li, Tuohang Li, Hongtu Li, and Ling Chi. 2020. A Review of Unsupervised Keyphrase Extraction Methods Using Within-Collection Resources. *Symmetry* 12. Multidisciplinary Digital Publishing Institute: 1864. https://doi.org/10.3390/sym12111864.

Teubert, Wolfgang. 1989. Politische Vexierwörter. In *Politische Semantik: Bedeutungsanalytische und Sprachkritische Beiträge zur politischen Sprachverwendung*, ed. Josef Klein, 51–68. Wiesbaden: VS Verlag für Sozialwissenschaften. https://doi.org/10.1007/978-3-322-91068-4_2.

Turney, Peter D. 2000. Learning Algorithms for Keyphrase Extraction. *Information Retrieval* 2: 303–336. https://doi.org/10.1023/A:1009976227802.

Williams, Raymond. 1976. *Keywords: A Vocabulary of Culture and Society*. USA: Oxford University Press.

Witten, Ian H., Gordon W. Paynter, Eibe Frank, Carl Gutwin, and Craig G. Nevill-Manning. 1999. KEA: Practical Automatic Keyphrase Extraction. arXiv. https://doi.org/10.48550/arXiv.cs/9902007.