Natural Language Processing (2024), 1–23
doi:10.1017/nlp.2024.59

ARTICLE

Unsupervised extraction of local and global keywords from a single text

Lida Aleksanyan and Armen Allahverdyan

Alikhanyan National Laboratory, Yerevan, Armenia
Corresponding author: Armen Allahverdyan; Email: armen.allahverdyan@gmail.com

(Received 26 August 2023; revised 10 June 2024; accepted 3 November 2024)

Abstract

We propose an unsupervised, corpus-independent method to extract keywords from a single text. It is based on the spatial distribution of words and the response of this distribution to a random permutation of words. Our method has three advantages over existing unsupervised methods (such as YAKE). First, it is significantly more effective at extracting keywords from long texts in terms of precision and recall. Second, it allows inference of two types of keywords: local and global. Third, it extracts basic topics from texts. Additionally, our method is language-independent and applies to short texts. The results are obtained via human annotators with previous knowledge of texts from our database of classical literary works. The agreement between annotators is moderate to substantial. Our results are supported via human-independent arguments based on the average length of extracted content words and on the average number of nouns in extracted words. We discuss relations of keywords with higher-order textual features and reveal a connection between keywords and chapter divisions.

Keywords: Information extraction; Information retrieval; Machine learning

1. Introduction

Keyword extraction from texts is important for information retrieval and NLP tasks (document searching within a larger database, document indexing, feature extraction, and automatic summarization) (Schütze *et al.* 2008; Firoozeh *et al.* 2020; Alami *et al.* 2020). As an analytical tool, keywords reflect the meaning of a text and help to extract its topics. Hence, keywords and their extraction schemes are also employed in discourse analysis (Bondi and Scott 2010). Our focus will be on this analytical aspect of keywords.

Keyword extraction is challenging, as the state-of-the-art results demonstrate (Kaur and Gupta 2010; Hasan and Ng 2014; Siddiqi and Sharan 2015). A possible explanation is the lack of a sufficiently comprehensive definition of the keyword concept. Keywords are generally non-polysemic nouns (i.e., nouns that do not have many sufficiently different meanings) related to text topics (Bondi and Scott 2010).

Computer science approaches to keyword extraction fall into three main groups; see section 2. First, there are supervised methods, which usually require a large training set to learn which keywords should be found and where to find them (Turney 2003; Gollapalli, Li, and Yang 2017;Song and Hu 2003; Devlin *et al.* 2018). The second group involves unsupervised methods that demand a corpus of texts for contrastive learning (Sparck Jones 1972; Robertson 2004; Ando and Zhang 2005; Scott and Tribble 2006). The third group involves methods that apply to a single text, that is, a text that does not belong to any corpus. Methods from this group rely on statistical (Luhn, 1958; Matsuo and Ishizuka 2004; Rose *et al.* 2010; Campos *et al.* 2018), or graph-theoretical features of a word in a text (Mihalcea and Tarau 2004; Wan and Xiao 2008; Bougouin, Boudin, and Daille 2013; Florescu and Caragea 2017); see section 2. The oldest method from this group is LUHN which selects sufficiently frequent content words of a text (Luhn, 1958). One of the latest state-of-the-art methods from the third group is YAKE (Campos *et al.* 2018, 2020).

Here, we focus on keyword extraction from literary works without supervision and corpus. One purpose of this task is to extract topical groups of keywords. We concentrate on well-known literary works because the confirmation of their keywords and topics should be available to practically anyone with a general education. (Sometimes, manual keyword extraction and validation require specialist expertise.) Another reason to work with literary works is that regular keyword extraction schemes can be applied to discourse analysis (Bondi and Scott 2010).

Our keyword extraction method belongs to the third group and is based on the specific spatial distribution of keywords in the text; see section 3. Systematic studies of the spatial distribution of words were initiated by Zipf (1945) and continued by Yngve (1956) and Herdan (1966); see section 2. Ortuño *et al.* (2002) and Herrera and Pury (2008) suggested to employ the spatial distribution for detecting the keywords; see section 2. This suggestion was taken up by Carretero-Campos *et al.* (2013), Mehri and Darooneh (2011), Mehri *et al.* (2015), and Zhou and Slater (2003).

However, several questions were open with these proposals. What are the best indicators for keywords based on spatial distribution-based methods? How do they compare to existing unsupervised single-text methods? Which keyword-extracting tasks can they help with?

We research these questions and to a large extent answer them. We use the spatial distribution of words for keyword detection, and our unsupervised and corpus-independent method is based on comparing the second (and sixth) moments of this distribution before and after a random permutation of words. By doing so, we capture two types of keywords: global and local. Global keywords are spread through the text and their spatial distribution becomes more homogeneous after a random permutation of words. By contrast, local keywords are found in particular parts of a text and clustered together. After a random permutation, their distribution becomes more homogeneous, and this can be employed for keyword detection. Analyzing several classical texts, we saw that this structural difference between the keywords indeed closely relates to the content of the text; for example, global and local keywords refer to (resp.) the main and secondary characters of the text. Thus, global keywords give the general idea of the text, whereas local keywords focus our attention on parts of the text. We note that the importance of global and local keywords was already understood in linguistics (Scott and Tribble 2006), but no systematic method was proposed there for their detection. Related ideas on different types of keywords appeared in Carpena *et al.* (2016).

Our method provides significantly better precision and recall of keyword extraction than several known methods, including LUHN (Luhn, 1958) and YAKE (Campos *et al.* 2018 2020), KeyBERT (Devlin *et al.* 2018), KEA (Witten *et al.* 1999), and WINGNUS (Nguyen and Luong 2010) (the first three methods are unsupervised and the latter two are supervised). We noted that despite its relative sophistication, for single-word keywords (i.e., not keyphrases) extracted from literary works, YAKE provides results that always approximately coincide with those of LUHN, though it outperforms graph-based methods (Campos *et al.* 2020) (we confirmed this point for texts studied here). Hence, we do not show comparison results with the latter group of methods. We also implemented for our texts another statistics-based method, RAKE (Rose *et al.* 2010), to confirm that YAKE outperforms it.

The advantage of our method is found via human annotators who determine if the extracted words are keywords based on their previous knowledge of classic literature texts in our database. There is moderate to substantial agreement between annotators. Additionally, we gave two indirect, but human-independent indications of the advantage of our method over the above methods. First, words extracted by our method have a longer length (in letters) than English content words on average. Therefore, we can infer indirectly that our method extracts text-specific words because it is known that the length of content words correlates with their average informativeness (Piantadosi, Tily, and Gibson 2012). Second, our method extracts more nouns. This is a proxy for keyword extraction since keywords are mostly nouns (Firoozeh *et al.* 2020).

In contrast to existing methods, our method is able to find topics from the text, that is, annotators were able to identify topical groups from a set of keywords extracted via our method. For the studied texts, serious topic extraction proved to be impossible with all alternative methods considered, including keyword extraction methods, as well as via several NLP topic modeling methods; see section 5.6. Our method is also nearly language-independent, as verified using translations in three languages: English, Russian, and French. It is only for long texts that our keyword extraction method is more efficient. For short texts our method does apply, but its efficiency of keyword extraction is similar to those of LUHN and YAKE. Still, its power in extracting the textual topics remains even for short texts.

To find out the limitations of our method, and to gain an understanding of what a keyword means conceptually, we aimed to relate keywords to the higher-order structures of texts, that is, the fact that literary texts are generally divided into chapters. This was accomplished by developing a method of keyword extraction that is based on chapter division. Even though this method is less efficient than our main method, it is easier to use in practice (for texts that already have many chapters), and it has the potential for further development; see section 6.

The rest of the paper is organized as follows. The next section reviews related work. In section 3, we discuss the main method analyzed in this work. Section 4 shows how the method applies to a classic and well-known text: *Anna Karenina* by L. Tolstoy. Section 5 evaluates our results in various ways. The inter-annotator agreement is also discussed in this section. Section 6 is devoted to the keyword extraction method that employs the fact that a long text is divided over sufficiently many chapters. The final section summarizes the discussion. Here, we emphasize that we considered only single-word keywords, and the extension of our method to extraction of keyphrases is an open problem.

2. Related work

In discussing various keyword extraction methods, one must remember that they are not universally applicable: each task (e.g., information retrieval, information extraction, document classification, and content analysis) requires its own methods. Keyword extraction methods are roughly divided into three groups: supervised, unsupervised but employing a text corpus, and unsupervised methods that apply to a single text. While in the context of the content analysis, we naturally focus on the last group and we shall also briefly review the two other groups.

Supervised methods are discussed in Gollapalli *et al.* (2017), Turney (2003), Song and Hu (2003), and Martinc et al., (2022). For general reviews on such methods, see Kaur and Gupta (2010), Siddiqi and Sharan (2015), Firoozeh *et al.* (2020), and Alami Merrouni *et al.* (2020). The supervision (training) stage normally demands a large training set with $> 10^4$ documents. Hence, such methods are prone to over-fitting and do not seem to be applicable for keyword extraction from a single literary work, though such applications are not excluded in principle and should be studied in the future. Some supervised approaches for keyword extraction employ linguistic-based handcrafted rules (Hulth 2003; Mihalcea and Tarau 2004; Firoozeh *et al.* 2020), which however lack language independence.

Unsupervised approaches include methods from statistics, information theory, and graph-based ranking (Siddiqi and Sharan 2015; Firoozeh *et al.* 2020; Alami Merrouni *et al.* 2020). The most recent review of unsupervised approaches is Nadim, Akopian, and Matamoros (2023). The best-known and widely used statistical approach is perhaps TF-IDF scoring function (Schütze *et al.* 2008; Sparck Jones 1972; Robertson 2004; Ando and Zhang 2005). Ideas that are similar to TF-IDF were independently researched in corpus linguistics (Scott and Tribble 2006; Bondi and Scott 2010). The method assumes that relevant words appear frequently in the given text and rarely in other texts in the corpus. Thus, the TF-IDF function relies on the existence of the corpus, that is, it does not apply to a single text.

Other unsupervised methods do apply to a single text. The first such method was proposed by Luhn (1958). It takes frequent content words as keyword candidates, excludes both high-probable content words and low-probable content words, and selects the rest as keyword candidates (Luhn, 1958). RAKE (Rose *et al.* 2010) and YAKE (Campos *et al.* 2018 2020) are two other methods that employ statistical information and apply it to a single document (without a corpus). In particular, YAKE emerged as the current state-of-the-art keyword extraction algorithm.

In graph-based methods (Mihalcea and Tarau 2004; Wan and Xiao 2008; Florescu and Caragea 2017; Škrlj *et al.* 2019), a text is represented as a graph where nodes are words and relations between words are expressed by edges. Normally, better-connected nodes (e.g., as determined by PageRank algorithm) relate to keywords, though other network features such as betweenness and closeness were also studied in the context of keyword extraction (Brin and Page, 1998; Boudin 2013). These methods mainly differ by the principles used to generate edges between words (Bougouin *et al.* 2013). Graph-based methods need only text information and hence are corpus-independent compared to TF-IDF. They can be semantically driven and agnostic of languages (Duari and Bhatnagar 2019).

KeyBERT is another unsupervised method of keyword extraction (Devlin *et al.* 2018). It inherits the pretrained BERT model (Bidirectional Encoder Representations from Transformers) developed by Google that understands the context of words in a sentence by considering the words that come before and after it. BERT is large language model that was trained on a large text corpus (including the entire English Wikipedia and the BookCorpus dataset) to learn language representations. Three recent keyword extraction methods that employ language models are Schopf *et al.* (2022), Tsvetkov and Kipnis (2023), and Liang *et al.* (2021).

Zipf and Fowler initiated systematic studies of the spatial distribution (or gap distribution) of words in texts (Zipf, 1945). Yngve (1956) and Herdan (1966) noted that the gap distribution of words is far from random and that this fact can be employed in quantitative linguistics. A pertinent open question is how to characterize this randomness (Brainerd, 1976; Zörnig, 1984 2010; Carpena *et al.* 2016). Ortuño *et al.* (2002) specifically applied the spatial distribution of words for detecting keywords in a single text, that is, without training and without a corpus. In (Ortuño *et al.* 2002), the variance of the spatial distribution is used for finding clustered words that are related to keywords. Later works (Herrera and Pury 2008; Mehri and Darooneh 2011; Carretero-Campos *et al.* 2013; Mehri *et al.* 2015) suggest several modifications that appear to improve the results. Herrera and Pury (2008) proposed to combine Shannon's information measure with the spatial distribution and studied the keyword distribution of *The Origin of Species* by Charles Darwin. Information-theoretic measures were also tried in Carretero-Campos *et al.* (2013), Mehri and Darooneh (2011), and Mehri *et al.* (2015). An alternative metric for keyword extraction was proposed by Zhou and Slater (2003). However, this variety of methods employing spatial distribution was not applied to a sufficiently large database. Also, no systematic comparison was attempted with the existing methods of keyword extraction from a single text. It was also unclear to which specific keyword-extracting tasks these methods apply. These issues are researched below.

3. Method

Below we discuss our method for keyword extraction (sections 3.2, 3.3) and describe implementation details; see sections 3.4 and 3.5. Section 3.1 introduces ideas on the example of spatial frequency, which shows interesting behavior but does not result in productive keyword indicators.

3.1 Distribution of words: spatial frequency

Our texts were lemmatized and freed from functional words (stop words); see section 3.4 for details. Let $w_{[1]}, \ldots, w_{[\ell]}$ denote all occurrences of a word w along the text. Let ζ_i denotes the number of words (different from w) between $w_{[i]}$ and $w_{[i+1]}$; that is, $\zeta_i + 1 \geq 1$ is the number of space symbols between $w_{[i]}$ and $w_{[i+1]}$. Define the first empirical moment for the distribution of $\zeta_i + 1$ (Yngve, 1956; Deng *et al.* 2021):

$$C_1[w] = \frac{1}{\ell - 1} \sum\nolimits_{i=1}^{\ell-1} (\zeta_i + 1). \tag{1}$$

Eq. (1) is not defined for $\ell = 1$, that is, for words that occur only once; hence, such words are to be excluded from consideration, that is, they will not emerge as keywords.

Note that $C_1[w]$ is the average period of the word w. Hence, the spatial frequency $\tau(w)$ can be defined via (Ortuño *et al.* 2002; Yngve, 1956; Carpena *et al.* 2009; Montemurro and Zanette 2010):

$$\tau[w] \equiv 1/C_1[w]. \tag{2}$$

The smallest value $\frac{1}{N-1}$ of $\tau[w]$ is attained for $\ell = 2$, where w occurs as the first and last word of the text. The largest value $\tau[w] = 1$ is reached when all instances of w occur next to each other (strong clusterization of w).

We compare $\tau[w]$ with the ordinary frequency $f[w]$ of word w:

$$f[w] = N_w/N, \tag{3}$$

where N_w is the number of times w appeared in the text ($N_w = \ell$), while N is the full number of words in the text. Now, $f[w]$ is obviously invariant under any permutation of words in the text.

Note that $(\ell - 1)(C_1[w] - 1)$ equals to the number of words that differ from w and occur between $w_{[1]}$ and $w_{[\ell]}$. Hence, $\tau[w]$ will stay intact at least under any permutation of words in that part of the text which is located between $w_{[1]}$ and $w_{[\ell]}$. This class of permutation is sufficiently big for frequent words (in the sense of (3)), where $w_{[1]}$ $[w_{[\ell]}]$ occurs close to the beginning [end] of the text. Consequently, we expect that a random permutation of all words in the text will leave $\tau[w]$ nearly intact for frequent words: $\tau[w] \approx \tau_{\text{perm}}[w]$. Indeed, we observed such a relation empirically. We also observed that there are many infrequent words for which $\tau[w] \gg \tau_{\text{perm}}[w]$, that is, such words are well clustered (before permutation).

These relations can be made quantitative by noting for frequent words the following implication of the above invariance. Aiming to calculate $\tau_{\text{perm}}[w]$ for a given frequent word w, we can employ the Bernoulli process of random text generation, assuming that w is generated independently from others, with probability of w (not w) equal to $f[w]$ $(1 - f[w])$; see (3). For spatial interval s between the occurrences of w, the Bernoulli process produces the geometric distribution $p(s) = (1 - f[w])^s f[w]$, where for sufficiently long texts we can assume that s changes from 0 to ∞, and $\sum_{s=0}^{\infty} p(s) = 1$. We emphasize that this model is not precise for a random permutation in texts, but it turns out to be sufficient for estimating $\tau_{\text{perm}}[w]$. The mean of this distribution is

$$f[w] \sum\nolimits_{s=0}^{\infty} s(1 - f[w])^s = (1 - f[w])/f[w]. \tag{4}$$

The inverse of (4) estimates $\tau_{\text{perm}}[w]$ for frequent words $\tau_{\text{perm}}[w] \simeq f[w]/(1 - f[w])$. On the other hand, we have $\tau[w] \simeq f[w]/(1 - f[w])$ for frequent words; see Figures 1 and 2. Two of the most famous world literature texts are described in these figures. Figure 1 refers to *Anna Karenina* by L. Tolstoy (the total number of words $N \approx 3.5 \times 10^5$), and Figure 2 refers to *Animal Farm* by G. Orwell ($N \approx 3 \times 10^4$). The length difference between the two texts is reflected in the difference between $\tau[w]$ and $f[w]/(1 - f[w])$. Figure 1 shows that relation:

$$f[w]/(1 - f[w]) \lesssim \tau[w], \tag{5}$$

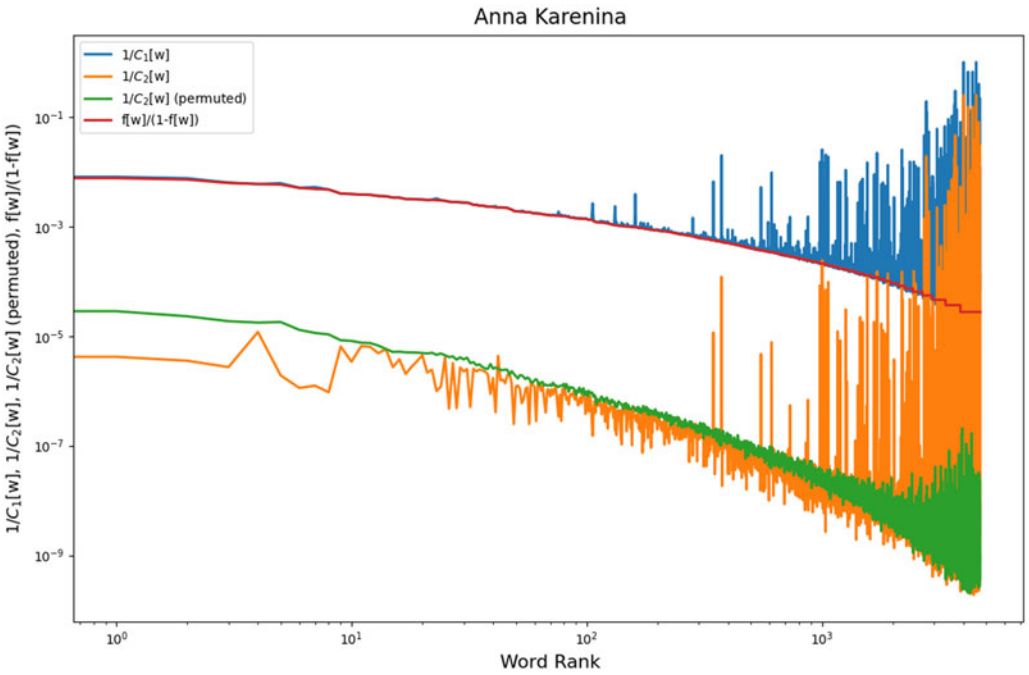

Figure 1. For *Anna Karenina* by L. Tolstoy (Tolstoy 2013), we show space frequency $\tau[w] = 1/C_1[w]$ and $1/C_2[w]$ *versus* word rank for all distinct words w of the text; cf. Eqs. (1, 6). We also show two additional quantities: $1/C_2[w] = 1/C_{2\,\mathrm{perm}}(w)$ after a random permutation of words in the text, and $f[w]/(1-f[w])$, where $f[w]$ is the frequency of w; see Eqs. (5, 3). Ranking of distinct words is done via $f[w]$, that is, the most frequent word got rank 1, *etc*. It is seen that $C_{2\,\mathrm{perm}}[w] < C_2[w]$ holds for frequent words. Both $C_{2\,\mathrm{perm}}[w] < C_2[w]$ and $C_{2\,\mathrm{perm}}[w] > C_2[w]$ hold for less frequent words. Not shown in the figure: a random permutation of the words in the text leaves $\tau[w]$ unaltered for frequent words, while $\tau[w]$ generically increases for less frequent words (clusterization); cf. Eq. (5).

holds for the majority of words with approximate equality for frequent words. In Figure 2, relation (5) holds for frequent words but is violated for some not-frequent words. For both figures, we see that $\tau[w]$ can be significantly larger than $f[w]/(1-f[w])$ for certain non-frequent words, indicating that the distribution of such words is clustered. We checked that there are not many keywords among such words, that is, the magnitude of $\tau[w](1-f[w])/f[w]$ is not a productive indicator for the keywords. More refined quantities are needed to this end.

3.2 The keyword extraction method: the second moment of the spatial distribution

Given Eq. (1), let us define the second moment of the spatial distribution for word w

$$C_2[w] = \frac{1}{\ell - 1}\sum_{i=1}^{\ell-1} (\zeta_i + 1)^2. \tag{6}$$

$C_2[w]$ is not invariant to those word permutations that left invariant $C_1[w]$; cf. the discussion before (5). Figures 1 and 2 show that for sufficiently frequent words w, $C_2[w]$ decreases after a random permutation. Indeed, frequent words are distributed in the text inhomogeneously. A random permutation makes this distribution more homogeneous and hence makes $C_{2\,\mathrm{perm}}[w] < C_2[w]$ for frequent words. For this conclusion, we need the second (or higher-order) moment in (6). Appendix B presents a numerical illustration of this effect and also illustrates that C_1 does not catch it.

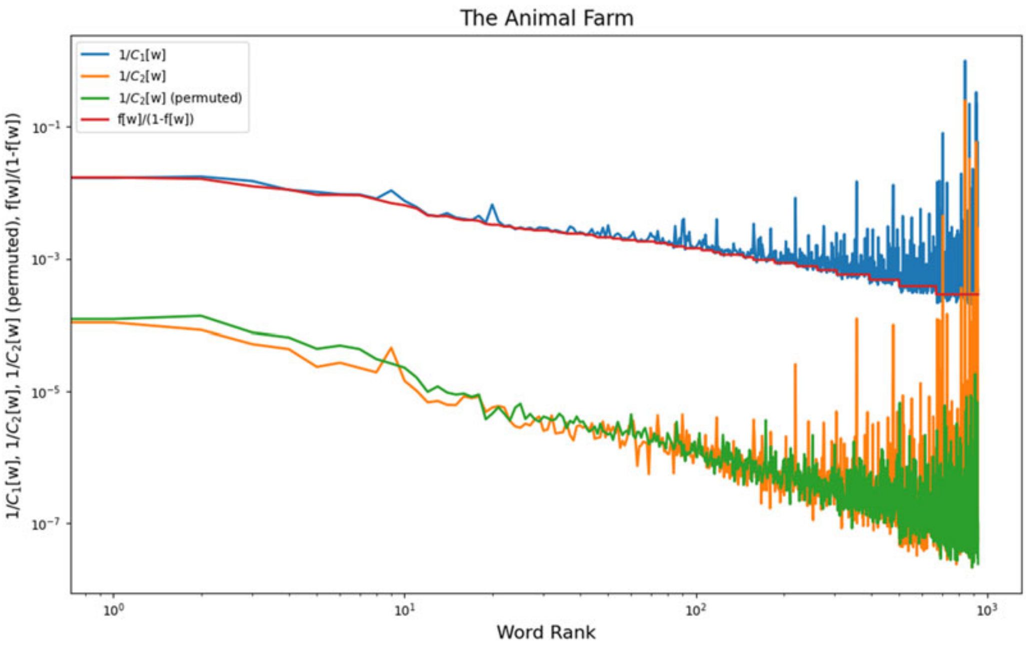

Figure 2. For *Animal Farm* (AF) by G. Orwell, we show the same quantities as for *Anna Karenina* (AK) in Figure 1 (also the same notations). AK is 11.6 times longer than AF; see Table 1. Some differences between these texts are as follows. Inequality $C_{2\,\mathrm{perm}}(w) < C_2(w)$ holds for a lesser number of frequent words in AF compared with AK. Domain $C_{2\,\mathrm{perm}}(w) < C_2(w)$ and $C_{2\,\mathrm{perm}}(w) > C_2(w)$ are well separated in AK, and not so well separated in AF. For AF, relation (5) can be violated for some infrequent words.

The situation changes for less frequent words: now it is possible that for some words non-frequent words we get $C_{2\,\mathrm{perm}}[w] > C_2[w]$; see Appendix B for examples. Those words are clustered in the original text, while after a random permutation, their distribution is more homogeneous; see Figures 1 and 2. For a long text *Anna Karenina*, the words where $C_{2\,\mathrm{perm}}[w]$ is noticeably larger than $C_2[w]$ appear at rank ≈ 300 (the rank is decided by frequency (3)); see Figure 1. There is no such a sharp threshold value for a shorter text *Animal Farm*, as Figure 2 shows. Using Eq. (6), we define

$$A[w] = \frac{C_{2\,\mathrm{perm}}[w]}{C_2[w]}, \tag{7}$$

where $C_{2\,\mathrm{perm}}[w]$ is calculated via Eq. (6) but after a random permutation of all words of the text.

When checking the values of $A[w]$ for all distinct words of several texts, our annotators concluded that sufficiently small and sufficiently large values of $A[w]$ in Eq. (7):

$$A[w] \leq \frac{1}{5}, \tag{8}$$

$$A[w] \geq 5, \tag{9}$$

can be employed for deducing certain keywords of the text. Eq. (8) extracts global keywords of the text, that is, keywords that go through the whole text. Eq. (9) refers to local keywords, that is, those that appear in specific places of the text. In Figure 1, they are seen as local maxima of $1/C_2[w]$. Local keywords are naturally located in the domain of infrequent words.

Taking in Eq. (8) a smaller threshold values

$$\frac{1}{5} \leq A[w] \leq \frac{1}{3}, \tag{10}$$

Table 1. Analyzed long texts: *Anna Karenina*, *War and Peace, part I*, and *War and Peace, part II* by L. Tolstoy; *Master and Margarita* by M. Bulgakov; *Twelve Chairs* by I. Ilf and E. Petrov; *The Glass Bead Game* by H. Hesse; *Crime and Punishment* by F. Dostoevsky. Shorter texts: *The Heart of Dog* by M. Bulgakov; *Animal Farm* by G. Orwell. *Alchemist* by P. Coelho. Next to each text, we indicate the number of words in it, stop words included.
For long texts we extracted for each text the same number of ≈ 300 potential keywords via each method: our method (implemented via Eqs. (8, 9, 10)), LUHN and YAKE. The numbers below are percentages, that is, $15.6 = 15.6\%$. For each text, the first percentage shows the values of precision (Prec.), that is, the fraction of keywords which were identified as keywords by human annotators. The second percentage shows recall (Rec.): the fraction of keywords that the methods were able to extract compared to ground-truth keywords; see (25). The third percentage shows the F1 score; see (26).
For short texts, we extracted via each method ~ 100 words. Our method was implemented via Eq. (13); only the precision is shown. For longer texts, our method provides sizable advantages compared with LUHN and YAKE. For shorter texts the three methods are comparable (the values for recall are not shown)

Method	LUHN			YAKE			Our method		
Text	Prec.	Rec.	F1	Prec.	Rec.	F1	Prec.	Rec.	F1
Anna Karenina (349762)	15.6	26.5	19.6	15.6	25.9	19.5	55.6	91.2	69.1
War and Peace, part I (142254)	22.7	39.1	29.0	22.7	39.1	29.0	42.9	81.7	56.3
War and Peace, part II (128146)	28.1	45.1	34.5	26.0	45.1	33.8	51.0	69.7	58.9
Master and Margarita (145286)	18.5	27.5	22.5	18.2	30.4	22.6	55.9	92.6	69.8
Twelve Chairs (102485)	20.5	36.7	25.9	20.5	38.3	26.3	38.3	67.5	50.2
The Glass Bead Game (192311)	17.7	36.5	24.1	19.3	39.7	25.7	33.9	67.9	45.3
Crime and Punishment (203505)	16.3	39.3	34.0	18.5	42.6	26.3	29.6	81.9	43.9
The Heart of Dog (34950)	26.2			24.3			26.2		
Animal Farm (30037)	49.6			47.3			45.8		
Alchemist (39004)	32.5			33.0			29.5		

leads to selecting a group of lower-frequency global keywords. Below we shall refer to Eq. (8) and Eq. (10) as (resp.) strong and weak cases.

Relations of Eq. (8) and Eq. (9) with (resp.) global and local keywords make intuitive sense. As we checked in detail, spaces between global keywords assume a broad range of values. This distribution becomes more uniform after the random permutation; hence, the the second moment decreases; cf. Eq. (8). Local keywords refer to infrequent words, are localized in a limited range of text and are clustered. Hence, a random permutation increases the second moment; cf. Eq. (9).

Let us comment on the choice of parameters in Eqs. (8, 10). This choice was taken as empirically adequate for *Anna Karenina*, that is, it led to extracting sufficiently many local and global keywords. (Other choices led to fewer global and/or local keywords.) After that, it was applied for all long texts [see Table 1] and led to adequate results.

As our method relies on random permutations, our results are formally dependent on the realization of these permutations. (Random permutations of words were generated via Python's numpy library; see Appendix A.) Such a dependence is weak: we noted that only a few keywords change from one realization to another. However, we cannot avoid random permutations; see section 7 for further discussion.

3.3 Modification of the method for shorter texts

Criteria (8, 9) based on $A(w)$ from Eq. (7) are not sufficiently powerful for discriminating between the keywords and ordinary words in sufficiently short texts; for example, in *Animal Farm* depicted on Figure 2. We found two modifications of the method that apply to short texts. The first option is to look at local maxima and minima of $A(w)$. The second, better option is to modify the order of the moment in Eq. (6). Instead of the second moment in Eq. (6), we employed the sixth moment

$$C_6[w] = \frac{1}{\ell - 1} \sum\nolimits_{i=1}^{\ell-1} (\zeta_i + 1)^6. \tag{11}$$

This modification leads to an indicator (13), which is more susceptible to inhomogeneity and clustering. Now $A_6[w]$ is defined analogously to Eq. (7), but via Eq. (11),

$$A_6[w] = \frac{C_{6\,\mathrm{perm}}[w]}{C_6[w]}, \tag{12}$$

and for extracting keywords we can apply [cf. Eqs. (8, 9)]:

$$A_6[w] \le \frac{1}{3}, \quad A_6[w] \ge 3. \tag{13}$$

The utility of Eqs. (12, 13) was determined for the short text *Animal Farm* and then applied for all other short texts; see Table 1.

3.4 Lemmatization of texts

English texts were preprocessed using WordNetLemmatizer imported from nltk.stem; see Appendix A. This library looks for lemmas of words from the WordNet database. The lemmatization uses corpus for excluding stop words (functional words) and WordNet corpus to produce lemmas. WordNetLemmatizer identifies the intended part of speech and meaning of a word in a sentence, as well as within the larger context surrounding that sentence, such as neighboring sentences or even an entire text. We applied this lemmatization algorithm on nouns, adjectives, verbs, and adverbs to get maximal clean up of the text. Any stemming procedure will be inappropriate for our purposes of extracting keywords, since stemming may mix different parts of speech.

For inflected languages (e.g., Russian), the lemmatization rules are more complex. For French and Russian texts, we used (resp.) lemmatizers LEFFF and pymystem3; see Appendix A.

3.5 Implementation of LUHN and YAKE

Here, we briefly discuss how we implemented Luhn's method (LUHN) for keyword extraction (Luhn, 1958). The method starts with ranking the distinct words of the text according to their frequency (more frequent words got a larger rank): $\{f_r\}_{r=1}^{n}$, where f_r is the word frequency and r is its rank. Next, one cuts off the high-frequency and low-frequency words and selects the remaining words $\{f_r\}_{r=r_{\min}}^{r_{\max}}$ as candidates for keywords. Hence, the method amounts to selecting the above cutoffs $r_{\min}$ and $r_{\max}$: the high-frequency words are to be omitted because there are many stop words there that normally are not considered as keywords. Low-frequency words are to be omitted since they are not relevant to the semantics of the text. Once we already skipped functional words from our texts, we did not apply the high-frequency cutoff, that is, we take $r_{\min} = 1$. For the low-frequency threshold, we employed a hypothesis that the essence of Luhn's method is related to Zipf's law, that is, to the law that fits the rank-frequency curve of distinct words of a text to a power law (Zipf, 1945); Allahverdyan, Deng, and Wang 2013). It is known that the power-law fitting works approximately till the rank r_{10} so that for $r \ge r_{10}$ the number of words having the same frequency f_r is 10 or larger (Allahverdyan *et al.* 2013). For $r > r_{10}$, the rank-frequency curve starts to show steps that cannot be fitted to a single power-law curve, that is, the proper Zipf's law becomes ill defined for $r > r_{10}$ (Allahverdyan *et al.* 2013). Hence, we selected the rank $r_{\max} = r_{10}$. This choice shows reasonable results in practice.

Table 2. The values of Cohen's kappa (14) for the agreement in keyword extraction tasks between two annotators for three different texts; see section 5.2. The keyword extraction employed the method discussed in Table 1 and section 3.2. Results for global and local keywords are shown separately. It is seen that the agreement is better for global keywords. A possible explanation is that the annotators do not focus on text details

Text/type of keyword	global	local
Animal Farm	0.68	0.40
Alchemist	0.83	0.53
Master and Margarita	0.78	0.65

Both LUHN and our method are based on a unique idea with a straightforward implementation. In contrast, YAKE incorporates various tools and ideas along with numerous empirical formulas (Campos *et al.* 2018 2020) (the same, but with a lesser degree holds for RAKE (Rose *et al.* 2010)). In particular, YAKE includes textual context of candidate keywords, sentence structure, sliding windows for keyword selection, statistics of n-grams, nontrivial (and multi-parametric) scoring process, *etc*. We worked with the version of YAKE that was implemented via a Python package; see Appendix A. We employed YAKE both with and without preprocessing of texts. In the second case, YAKE was better at extracting capitalized proper nouns (such words are frequently keywords). Otherwise, its performance did not change much. This advantage of YAKE is due to a specific tool implemented in it: YAKE looks for capitalized words which do not appear in the beginning of a sentence. Such a tool is easy to implement in any keyword searching method, but we avoided doing that, since we are interested in checking ideas behind Eqs. (6–13). Therefore, we mostly discuss YAKE's outcomes after preprocessing.

4. Keywords extracted from *Anna Karenina*

The above keyword extraction method was applied to several texts of classic literature; see Table 1. (Our data for texts from Table 1 is freely available at https://github.com/LidaAleksanyan/keywords_extraction_data/tree/master, while our codes are available at https://github.com/LidaAleksanyan/spatial_keyword_extraction.) Among them, we choose one of the most known works of classic literature, *Anna Karenina* by L. Tolstoy, and analyze in detail the implications of our method in extracting and interpreting its keywords. The evaluation of extracted keywords was done by annotators with expert knowledge of classic Russian literature and specifically works by Tolstoy. The agreement between annotators is moderate to substantial; see Table 2.

4.1 Comparison with known methods of keyword extraction and language independence

Using *Anna Karenina* (Tolstoy 2013), we compared our approach discussed in section 3.2 with two well-known methods that also apply to a single text (i.e., do not require corpus): LUHN and YAKE; see section 3.5.

Two hundred eighty-two words were extracted via each method, and the keywords were identified. Tables 1 and 3 show that for three languages (English, Russian, and French) our method is better in terms of both precision and recall; see Appendix C for a reminder of these concepts. The relatively poor performance of YAKE and LUHN can be explained by their focus on relatively short content words that are not likely to be keywords. We quantified this by calculating the mean number of letters in each set of 282 words. For our method, LUHN and YAKE the mean is (resp.) 6.95, 5.43, and 5.5; cf. the fact that the average number of letters in English content word is 6.47 (for stop word it is 3.13) (Miller, Newman, and Friedman, 1958).

Table 3. Comparison of three different keyword extraction methods for English, Russian, and French versions of *Anna Karenina*. Percentages for keywords indicate the precision [cf. Table 1], while ‘nouns’ means the percentage of nouns in candidate words that were not identified as keywords. For all cases, our method fares better than LUHN and YAKE

Method	English keywords	English nouns	Russian keywords	Russian nouns	French keywords	French nouns
LUHN	15.6 %	54 %	14.1%	51.1%	19.2%	62.3 %
YAKE	15.6 %	55 %	14.8%	49.2%	18%	60 %
Our method	55.6 %	82 %	55%	86.2%	50.7%	77.3%

The three methods have scores for words. In LUHN and our method, the score coincides with the word frequency. For YAKE, the scores are described by Campos *et al.* (2018 2020). However, for LUHN and YAKE, the score did not correlate with the feature of being keyword. For our method it certainly did, that is, by selecting only high-score words we can significantly enlarge the percentage of keywords compared to what is seen in Table 1. These two facts (low density of keywords plus no correlation with their score) make it impossible to extract topical groups of keywords via LUHN and YAKE; cf. the discussion after Eq. (9).

Another comparison criterion between the three methods is the amount of nouns in words that were not identified as keywords. Indeed, once keywords are predominantly nouns, a method that extracts more nouns (e.g., more nouns in candidate words that were not identified as keywords) has an advantage. In this respect, our method fares better than both LUHN and YAKE; see Table 3.

Table 3 also addresses the language independence of the three methods that were studied in three versions (English, Russian, and French) of *Anna Karenina*. Our method performs comparably for English and Russian, which are morphologically quite distinct languages. For French the performance is worse, but overall still comparable with English and Russian. Altogether, our method applies to different languages. This confirms an intuitive expectation that spatial structure features embedded into Eqs. (6–13) are largely language-independent.

4.2 Topical groups extracted via extracted keywords

Anna Karenina features more than a dozen major characters and many lesser characters. Annotators separated keywords into nine topical groups: (1) proper names of major characters; (2) proper names of secondary characters; (3) animal names; (4) trains and railway; (5) hunting; (6) rural life and agriculture; (7) local governance (zemstvo); (8) nobility life and habits; and (9) religion; see Table 4.

The names of these characters are keywords because they inform us about the character’s gender (’anna’ *vs.* ’vronsky’), age (’alexandrovitch’ *vs.* ’seryozha’) and the social strata; for example, ’tit’ *versus* ’levin’. Proper nouns provide additional information due to name symbolism employed by Tolstoy; for example, ’anna’=’grace’ ’alexey’=’reflector’ ’levin’=’leo’ is the *alter ego* of Tolstoy (Gustafson 2014).

All the main character names came out from our method as strong global keywords holding condition $A[w] \leq \frac{1}{5}$ in Eq. (8): ’levin’, ’anna’, ’vronsky’, ’kitty’, ’alexey’, ’stepan’, ’dolly’, ’sergey’; see Table 4 for details. Many pertinent lesser characters came out as local keywords, as determined via Eq. (9); for example, ’vassenka’, ’golenishtchev’, ’varvara’; see Table 4. Important characters that are not the main actors came out as weak global keywords, for example, ’seryozha’, ’yashvin’, and ’sviazhsky’.

The novel is also known for its animal characters that play an important role in Tolstoy’s symbolism (Gustafson 2014). Our method extracted as local keywords the four main animal characters: ’froufrou’, ’gladiator’ ’laska’, and ’krak’. Trains are a motif throughout the novel (they symbolize the modernization of Russia), with several major plot points taking place either on

Table 4. Words of *Anna Karenina* extracted via our method. For global keywords strong and weak cases mean (resp.) that the words w were chosen according to $A(w) \leq \frac{1}{5}$ and $\frac{1}{5} \leq A(w) \leq \frac{1}{3}$; cf. Eqs. (8, 10). Local keywords were chosen according to $A(w) \geq 5$; see Eq. (9). For each column, the words were arranged according to their frequency Eq. (3). Keyword classes are denoted by upper indices; see details in the text. The last group $^{(10)}$ denotes words that were identified as keywords but did not belong to any of the above groups. Words without the upper index were not identified as keywords

Global keywords strong cases	Global keywords weak cases	Local keywords
levin$^{(1)}$, anna$^{(1)}$, vronsky$^{(1)}$, kitty$^{(1)}$, alexey$^{(1)}$, stepan$^{(1)}$, alexandrovitch$^{(1)}$, arkadyevitch$^{(1)}$, dolly$^{(1)}$, sergey$^{(1)}$, ivanovitch$^{(1)}$, peasant$^{(6)}$, darya$^{(1)}$, alexandrovna$^{(1)}$, varenka$^{(1)}$, lidia$^{(1)}$, death, ivanovna$^{(1)}$, laborer$^{(6)}$, mow$^{(6)}$, district$^{(7)}$, stahl$^{(1)}$, bailiff$^{(5)}$, gun$^{(5)}$, snipe$^{(5)}$, plough$^{(6)}$, rain, lesson$^{(10)}$, lord$^{(9)}$, acre$^{(6)}$, platform$^{(4)}$, natalia$^{(1)}$, built, rich, overlook, river, crime$^{(10)}$, rail$^{(6)}$, relate, throb, contrast, puzzle, cheat$^{(10)}$, oppress, irrational$^{(10)}$	love, princess$^{(8)}$, brother, carriage$^{(4)}$, horse$^{(8)}$, prince$^{(8)}$, doctor$^{(8)}$, countess$^{(8)}$, madame$^{(8)}$, sviazhsky$^{(1)}$, land$^{(6)}$, seryozha$^{(1)}$, konstantin$^{(1)}$, picture, oblonsky$^{(1)}$, nikolay$^{(1)}$, agafea$^{(2)}$, katavasov$^{(2)}$, grass$^{(6)}$, yashvin$^{(1)}$, shoot$^{(5)}$, mihalovna$^{(2)}$, officer$^{(8)}$, box, marshal$^{(7)}$, mare$^{(6)}$, priest$^{(9)}$, tree$^{(6)}$, forest$^{(6)}$, laska$^{(3)}$, law$^{(10)}$, landowner$^{(6)}$, realize, scythe$^{(6)}$, telegram$^{(8)}$, meadow$^{(6)}$, bedroom$^{(8)}$, argument, sledge, nobleman$^{(8)}$, paint, article$^{(8)}$, professor$^{(8)}$, scream, sky, trap, birch$^{(6)}$, cow$^{(6)}$, debt$^{(10)}$, rent, punish, sow$^{(6)}$, annushka$^{(2)}$, lightly, sportsman$^{(8)}$, myakaya$^{(2)}$, invalid, smart, parent, vividly, maman$^{(8)}$, institution$^{(7)}$, stable, distance, salary$^{(10)}$, educate, firm, skirt, mahotin$^{(2)}$, reconciliation, yellow, plump, childrens, tatar$^{(2)}$, outer, steward$^{(8)}$, cousin, loathsome, sharp, splash, armchair$^{(8)}$, understands, coarse, quicken, grace, delicious, director$^{(8)}$, unseen, selfpossession, cheese, rate, physically, timidity, tucked, reassure, sunday, compartment, frost, minister$^{(8)}$, won, king, repent, clock, wage, shock, uncertain, deliver, cream, silently, monday, captain$^{(8)}$, shaft$^{(6)}$, matrona$^{(8)}$, strictly, original	vassenka$^{(2)}$, golenishtchev$^{(2)}$, election$^{(7)}$, skate$^{(10)}$, varvara$^{(2)}$, pyotr$^{(2)}$, lizaveta$^{(2)}$, landau$^{(2)}$, petrovna$^{(2)}$, gladiator$^{(3)}$, metrov$^{(2)}$, tit$^{(2)}$, vote$^{(7)}$, froufrou$^{(3)}$, ryabinin$^{(2)}$, volunteer$^{(8)}$, nevyedovsky$^{(2)}$, duel$^{(8)}$, scandal$^{(8)}$, tribe$^{(10)}$, snetkov$^{(2)}$, lukitch$^{(2)}$, mower$^{(6)}$, deacon$^{(9)}$, native, korsunsky$^{(2)}$, hospital, remote, mazurka$^{(8)}$, pilate$^{(10)}$, sappho$^{(10)}$, villa$^{(8)}$, rival, reed$^{(6)}$, bridegroom$^{(8)}$, krak$^{(3)}$, merkalova$^{(2)}$, vorkuev$^{(2)}$, photograph$^{(8)}$, yegor$^{(2)}$, mitya$^{(2)}$, kapitonitch$^{(2)}$, architect$^{(8)}$, intensely, elect$^{(7)}$, golenishtchevs$^{(2)}$, pa$^{(8)}$, birthday, trousseau$^{(8)}$, transition, chalk, potato$^{(6)}$, kritsky$^{(2)}$, ergushovo$^{(6)}$, katya$^{(2)}$, weep, sympathetic, repair, mais$^{(8)}$, seryozhas$^{(2)}$, ballroom$^{(8)}$, classical, vozdvizhenskoe$^{(6)}$, technique, bedchamber$^{(8)}$, opium$^{(8)}$, penetrate, tchirikov$^{(2)}$, rider, palazzo$^{(8)}$, crown$^{(8)}$, remove, miracle, intolerable, turk$^{(2)}$, ballot$^{(7)}$, custom, nevsky$^{(8)}$, adultery$^{(8)}$, ditch, musical

passenger trains or at stations in Russia (Tolstoy 2013; Gustafson 2014). Our method extracted among the global keywords ’carriage’, ’platform’, and ’rail’. Hunting scenes are important in the novel depicting the life of Russian nobility. Accordingly, our method extracted keywords related to that activity: ’snipe’, ’gun’, and ’shoot’. Two major social topics considered in the novel are local democratic governance (Zemstvo) and the agricultural life of by then mostly rural Russia. For the first, we extracted keywords: ’district’, ’bailiff’, ’election’ *etc*. And for the second: ’mow’, ’lord’, ’acre’, *etc*. A large set of keywords are provided by Russian nobility’s living and manners, including their titles, professions, and habits; see Table 4. Religion and Christian faith is an important subject of the novel. In this context, we noted keyword ’Lord’, ’priest’, and ’deacon’; see Table 4. Finally, a few words stayed out of these topical groups but was identified as keywords: ’lesson’, ’crime’, ’cheat’, ’salary’, ’irrational’, ’law’, ’skate’, and ’tribe’.

5. Evaluation

5.1 Precision and recall

Results obtained for *Anna Karenina* are confirmed for several other texts; cf. Table 1. We extracted for each text the same number of ≈ 300 potential keywords via three methods: our method

Table 5. Here we discuss topical groups extracted from a short text. *Heart of Dog* by M. Bulgakov is a known satirical novella that shows the post-revolutionary Moscow (first half of the 1920s) under social changes, the emergence of new elites of Stalin’s era, and science-driven eugenic ideas of the intelligentsia. Eventually, the novella is about the life of a homeless dog Sharik (a standard name for an unpedigreed dog in Russia) picked up for medical and social experiments. The majority of keywords below were not even extracted via LUHN and/or YAKE

Topical group	Keywords
Canine features	’dog’, ’sharikov’, ’salami’, ’sharikovs’, ’bite’, ’cracow’, ’scald’, ’animal’, ’sharik’, ’cat’, ’bitten’, ’canine’, ’phewphew’, ’cur’, ’claw’, ’bitch’, ’mange’, ’shaggy’, ’phew’, ’paw’, ’bark’, ’wild’, ’biting’, ’oooo’
Medical terms	’skull’, ’camphor’, ’weight’, ’temperature’, ’method’, ’stitch’, ’pulse’, ’organism’, ’injection’, ’laboratory’, ’sore’, ’needle’, ’scissors’, hospital’, ’respiratory’, ’gauze’, ’adrenaline’, ’clinic’, ’doze’, ‘heal’, ’transplant’, ’phonograph’, ’hypothesis’, ’organism’, ’nostril’, ’injection’, ’subdepartment’, ’department’, ’laboratory’, ’hospital’, ’rejuvenation’, ’throat’, ’scholar’, ’brow’, ’cheek’, ’lip’, ’strip’, ’experiment’, ’forehead’, ’hormone’, ’breast’, ’science’, ’hypophysis’, ’brain’
Revolution	’proletariat’, ’terror’, ’kautsky’, ’council’, ’bourgeois’, ’proletarian’, ’revolution’, ’war’, ’worker’, ’engels’, ’pest’, ’revolver’, ’social’, ’chairman’, ’committee’
Moscow	’moscow’, ’kalabukhov’, ’blizzard’, ’bolshoi’, ’mosselprom’, ’nikitins’, ’prechistenka’, ’swearword’

(implemented via Eqs. (8, 9, 10)), LUHN and YAKE. The precise number of extracted words depends on the text.

As seen from Table 1, for long texts (with the length roughly comparable with *Anna Karenina*), our method outperforms both LUHN and YAKE in terms of the precision, that is, the relative number of extracted keywords which is defined as the number of keywords extracted via the given method (for each text) divided over the total number of words proposed by the method as potential keywords. The advantage of our method is also seen in terms of recall, which is the number of keywords extracted via the given method (for each text) divided over the full number of keywords announced by an annotator for the text. (The definitions of precision and recall are reminded in Appendix C; in particular, the above results were found via Eqs. (25).) Importantly, for YAKE and LUHN the values of recall are lower than 0.5, while for our method they are sizably larger than 0.5 meaning that our method extracted the majority of potential keywords; see Table 1.

In this context, we distinguish between long and short texts; cf. Table 1. For short texts, our method needs modifications that are described above. After these modifications, our method implemented via Eq. (13) produces for short texts nearly the same results as LUHN and YAKE; see Table 1. For short texts, we extracted via each method the same number of ~ 100 words. However, our method still has an important advantage, since it allows us to extract topical groups of short texts nearly in the same way as for long texts; see Table 5 where we analyze topical groups of *The Heart of Dog* by M. Bulgakov. We emphasize that this feature is absent for LUHN and YAKE.

5.2 Inter-anontator agreement

The performance of any keyword extraction method is evaluated by annotators. First, annotators should be provided with guidelines on the extraction process; for example, characters are keywords and pay more attention to nouns and less to verbs and adjectives, *etc*. Second, two (or more) annotators are independently given the set of keywords extracted by our algorithm from the same set of texts, and they mark words that they consider as keywords. So each annotator will get at the end a list of keywords *versus* non-keyword. Annotators are influenced by various subjective factors: background, prior knowledge, taste, *etc*. However, the situation will not be subjective if different annotators produce similar results. To quantify the agreement between annotators, we employed Cohen’s kappa κ; see Table 2. This statistical measure assesses inter-annotator agreement when working on categorical data in linguistics, psychology, and information retrieval;

see (Cook 2005) for review. It accounts for chance agreement and provides a more robust evaluation of agreement than the simple percentage. Cohen's κ reads

$$\kappa = \frac{p_o - p_e}{1 - p_e}, -1 \geq \kappa \geq 1, \tag{14}$$

$$p_o = p(A = \mathrm{k}, B = \mathrm{k}) + p(A = \mathrm{nk}, B = \mathrm{nk}), \tag{15}$$

$$p_e = p(A = \mathrm{k})p(B = \mathrm{k}) + p(A = \mathrm{nk})p(B = \mathrm{nk}), \tag{16}$$

where $p(A = \mathrm{k}, B = \mathrm{k})$ is the joint probability for annotators A and B to identify keyword, $p(A = \mathrm{nk}, B = \mathrm{nk})$ is the same for non-keywords (denoted by nk), $p(A = \mathrm{k})$ is the marginal probability, *etc*. Hence, p_o is the agreement probability, while p_e is the probability to agree by chance. Now $\kappa \to p_o$ for $p_e \to 0$, while for $p_e \lesssim 1$ even a relatively small, but positive difference $p_o - p_e$ is sufficient for $\kappa \to 1$.

Numerical interpretation of κ is as follows (Cook 2005). No agreement: $\kappa < 0$. Slight agreement: $0.2 > \kappa > 0$. Fair agreement: $0.4 > \kappa > 0.2$. Moderate agreement: $0.6 > \kappa > 0.4$. Substantial agreement: $0.8 > \kappa > 0.6$. Following these steps we got an agreement, which is between moderate and substantial both for short and long texts; see Table 2. The agreement is better for global keywords, as expected.

5.3 Comparison with RAKE

The method we developed was compared with two unsupervised keyword extraction methods that are among the best, LUHN and YAKE. As an additional comparison, let us take a look at RAKE (Rapid Automatic Keyword Extraction), another unsupervised method (Rose *et al.* 2010). Its standard implementation in https://pypi.org/project/python-rake/ requires that punctuation signs and stop words are conserved in the text. It returns keyphrases that we partitioned into separate words. Here are the first (i.e., the highest score) 50 words extracted by RAKE for *Anna Karenina* (English version): *finesses, ces, toutes, par-dessus, passer, influence, mot, nihilist, le, moral, disons, plaisir, h FC;bsch, recht, auch, ich, brisé, est, en, moule, le, monde, du, merveilles, sept, les, jusqu, tomber, va, poulet, ce, blague, une, est, ça, tout, devoir, votre, oubliez, vous, et, ausrechnen, sich, lässt, das, terre-á-terre, excessivement, mais, amour, le, filez, vous*. These words are mostly French and German (not English), and they are certainly far from being keywords.

5.4 Comparison with KeyBERT

In section 2, we mentioned KeyBERT, an unsupervised method that uses BERT to convert the input text and potential keywords into high-dimensional vectors (embeddings) Devlin *et al.* (2018). These embeddings capture the semantic meaning of the words and phrases. BERT is a transformer deep neural network with at least 110 million parameters, which looks for the context of words by considering the entire sentence in a bidirectional way. KeyBERT generates a list of candidate keywords and keyphrases from the input text. For each candidate keyword/keyphrase, KeyBERT calculates the similarity between the embedding of the candidate and the embedding of the entire input text. High similarity means a higher score for a potential keyword.

We applied KeyBERT to *Anna Karenina*. To apply it efficiently, we extracted two-word keyphrases (which could then be split into different keywords if necessary). Below we present the 100 highest-score results of these applications together with their scores. Only nine single keywords were actually extracted over 200 words: *anna, karenina, marriage, annushka, madame, peasant, karenins,* and *sviazhsky*; that is, the performance is much lower than for our method. The fact of two-word keyphrases provides additional information, but this information is of a specific type: it associates *anna* with words such as *husband, wife, marriage, household, love, married,* *sincerely, woman, madame, fashionable, courteously, unpardonable, emotionalism, lady, etc.* This provides some information about what *anna* does or is involved in. We recall from our extensive discussion in section 4 that *Anna Karenina* is certainly not solely about Anna Karenina but presents an epochal representation of Russian life at the end of nineteenth century.

Hence, when evaluated over long texts, KeyBERT performed worse than our method in terms of precision, recall, and topic extraction. We exemplified these facts on *Anna Karenina*, but they hold for several other long texts we checked.

('anna husband', 0.4899); ('anna karenina', 0.489); ('anna wife', 0.4878); ('wife anna', 0.4833); ('household anna', 0.4813); ('karenina anna', 0.4755); ('husband anna', 0.4676); ('karenina marriage', 0.4615); ('love anna', 0.4607); ('anna involuntarily', 0.4606); ('superfluous anna', 0.4595); ('karenina husband', 0.4585); ('anna sincerely', 0.4533); ('married anna', 0.4499); ('care anna', 0.4464); ('anna married', 0.4455); ('anna annushka', 0.4446); ('anna commonplace', 0.4443); ('anna unnaturalness', 0.4424); ('arrange anna', 0.4422); ('anna strangely', 0.4417); ('anna love', 0.4415); ('anna irritates', 0.4413); ('anna peasant', 0.4406); ('anna courteously', 0.4406); ('woman anna', 0.4404); ('anna unpardonably', 0.4403); ('anna distress', 0.4401); ('dear anna', 0.4399); ('anna wonderfully', 0.4395); ('anna unmistakably', 0.4387); ('anna fashionable', 0.4382); ('madame karenina', 0.438); ('anna dear', 0.4379); ('anna lovely', 0.4375); ('anna emotionalism', 0.437); ('anna woman', 0.4367); ('annushka anna', 0.4363); ('distinctly anna', 0.4359); ('intensely anna', 0.4353); ('living anna', 0.4348); ('anna fascinate', 0.4344); ('perceive anna', 0.4344); ('recognize anna', 0.4343); ('anna rarely', 0.4343); ('maid anna', 0.4342); ('remarkable anna', 0.434); ('dress anna', 0.4326); ('irritate anna', 0.4323); ('karenina leo', 0.4302); ('person anna', 0.43); ('anna lady', 0.429); ('anna sister', 0.4281); ('repeat anna', 0.4276); ('anna occupation', 0.4276); ('anna meant', 0.4274); ('unusual anna', 0.4271); ('expression anna', 0.4268); ('anna karenin', 0.4264); ('karenins household', 0.426); ('turn anna', 0.4258); ('anna write', 0.4255); ('peasant anna', 0.4253); ('sincerely anna', 0.4253); ('unbecoming anna', 0.425); ('anna madame', 0.4245); ('occupation anna', 0.4242); ('discern anna', 0.4242); ('anna kindly', 0.4239); ('anna care', 0.4229); ('anna mentally', 0.4227); ('anna constantly', 0.4223); ('altogether anna', 0.4222); ('word anna', 0.4209); ('couple anna', 0.4209); ('special anna', 0.4208); ('anna interpose', 0.4204); ('anna query', 0.4203); ('anna dreamily', 0.4202); ('uttered anna', 0.4202); ('delight anna', 0.4199); ('indicate anna', 0.4196); ('description anna', 0.4195); ('anna anna', 0.4194); ('anna sviazhsky', 0.4187); ('complain anna', 0.4184); ('exceptional anna', 0.4182); ('anna irritable', 0.4175); ('anna properly', 0.4174); ('meet anna', 0.417); ('anna indifferently', 0.4167); ('anna manner', 0.4165); ('anna resolutely', 0.4164); ('anna memorable', 0.4163); ('anna wholly', 0.4161); ('anna sisterinlaw', 0.4158); ('anna special', 0.4158); ('articulate anna', 0.4153); ('explain anna', 0.4152); ('anna recognize', 0.4151)

5.5 Comparison with supervised methods

We compare our results with two supervised methods: KEA (Witten *et al.* 1999) and WINGNUS (Nguyen and Luong 2010). These two methods were selected because they are relatively new, their software is free, and their structure is well documented in literature. The implementation of both cases were taken from pke python library: https://github.com/boudinfl/pke/tree/master?tab=readme-ov-file. Both methods were trained on the semeval2010 dataset available from https://aclanthology.org/S10-1004/. This dataset amounts to 284 abstracts from scientific articles included in this dataset, which were annotated by both authors and independent annotators.

The performance of KEA and WINGNUS was checked on long texts from Table 1. Their performance is generally lower than for our method. Here are two examples framed in terms of precision. For *Anna Karenina* KEA and WINGNUS led to 12.3% and 10%, respectively. For *Master and Margarita,* these numbers are 15.3% and 8.3%, respectively; cf. Table 1.

5.6 Comparison with topic modeling methods

Above we described how human annotators can deduce topical groups of *Anna Karenina* using keywords extracted by our method. The same task—known as topic modeling—is achieved algorithmically (i.e., without human intervention) by several known NLP methods (Pedregosa *et al.* 2011). We focused on three known topic modeling methods (all of them were applied without supervising), and below we compare their efficiency with our results. Before proceeding, let us note that topic modeling is generally applied to an entire corpus of texts, not just a single one. However, there are reasons to believe it can also be applied to a sufficiently long text.

Non-Negative Matrix Factorization (NNMF) is an unsupervised method of topic modeling that applies to a single text; see Appendix A. Its implementation for *Anna Karenina* produced the following 10 potential topics (their number is a hyperparameter of the model).

T-1: *levin, vronsky, anna, kitty, alexey, alexandrovitch, arkadyevitch, stepan, room, wife.*

T-2: *walked, beauty, maid, gentleman, people, frou, complete, carriage, completely, coming*

T-3: *living, sviazhsky, meaning, mare, kitty, book, natural, listening, friends, suppose*

T-4: *read, work, ivanovitch, coat, drove, agriculture, lack, hearing, matters, living*

T-5: *serpuhovskoy, received, young, desire, pass, asleep, set, action, clerk, stay*

T-6: *doctor, stahl, coming, today, passion, porter, silence, movement, object, levin*

T-7: *tanya, remember, game, deal, live, mamma, walking, bare, easy, hurriedly*

T-8: *noticed, possibility, christian, dmitrievitch, feelings, fall, forget, success, stopped, suffer*

T-9: *early, covered, recognized, angrily, connection, expression, figure, breathing, nice, friend*

T-10: *scythe, nobility, elections, minutes, promised, extreme, afraid, decided, ordered, lifting*

Only one of them (T-1) was reliable and approximately coincided with the first topical group (proper names of major characters) discussed above; see Table 4. This is not surprising, since this topical group contains the most frequent content words of the text. Other topics extracted by NNMF turn out to be meaningless, that is, they do not correspond to any topical group of the text.

We applied two alternative topic modeling methods: Truncated SVD and LDA (Latent Dirichlet Allocation); see Appendix A. Truncated SVD applied to a single text. LDA was attempted both for a single text directly and after training on a group of ≈ 100 long texts. Both methods produced similar results: several topics were discovered, but all those topics were closely related to each other. Eventually, this situation amounts to only one sensible topic: the proper names of major characters. Let us illustrate this situation on the main topic discovered by LDA for *Anna Karenina*: *levin, vronsky, anna, arkadyevitch, alexey, kitty, hand, stepan, long, alexandrovitch*. In variations of this topic, *hand* can be changed to *wife*, or *brother*, *etc*. Likewise, LDA extracted (effectively) a single topic for other long texts, for example, *Master and Margarita*. This topic is based on the following words: *margarita, ivan, hand, procurator, began, asked, woland, pilate, master, koroviev*. Again, besides *began* and *asked*, these are the proper names of the main characters.

In sum, NNMF, LDA, and truncated LDA produced consistent results for the studied long texts: they extracted a single topic based on the proper names of the main characters. This cannot be considered as a productive result.

5.7 Comparison with the standard database

We created all of the above databases in order to evaluate the method we employed. The standard databases should also be used in such cases. Hence, we analyzed a database of 100 theses [see https://github.com/LIAAD/KeywordExtractor-Datasets?tab=readme-ov-file#theses]. The average length of texts from this database is roughly comparable with the length of short texts from Table 1. Each of these texts was annotated manually and got 10 keywords. We also implemented LUHN, YAKE, and our method, each extracting 10 keywords from every database text. The evaluation of each method was based on the standard F1 score, which is the harmonic mean of the precision and recall; see (23) in Appendix C. The evaluation results are as follows. For eight database texts, all three methods produced zero F1 score. For the remaining 92 texts, LUHN wins (i.e., its F1 score is largest among three) in 73 cases, YAKE wins in 68 cases, and our method is the winner in 62 cases from 92, that is, the performance of all three methods is comparable. (The concrete values of the F1 score can be looked up from https://github.com/LidaAleksanyan/keywords_extraction_data/tree/master) This is an expected conclusion for two independent reasons. First, the average length of the texts from the database is comparable with that of short texts from Table 1, where we also see roughly equal performance of all three methods. Second, the extracted number of keywords is small (i.e., 10). Even for longer texts, we would not see a well-defined advantage of our method for such a small number of extracted keywords. Unfortunately, standard databases do not have sufficiently many extracted keywords per text, even when the text is complex and structured.

6. Keyword extraction and distribution of words over chapters

Long texts are frequently divided into sufficiently many chapters. It is an interesting question whether this fact can be employed as an independent criterion for extracting keywords. To search for such criteria, let us introduce the following basic quantities. Given a word w and chapters $c = 1, .., N_{\text{chap}}$, we define $m_w(c) \geq 0$ as the number of times w appeared in chapter c. Likewise, let $V_w(s)$ be the number of chapters, where w appeared $s \geq 0$ times; that is, $\sum_{s \geq s_0} V_w(s)$ is the number of chapters, where w appears at least s_0 times. We have

$$\sum\nolimits_{c=1}^{N_{\text{chap}}} m_w(c) = N_w, \tag{17}$$

$$\sum\nolimits_{s \geq 0} s V_w(s) = N_w, \tag{18}$$

where N_w is the number of times w appears in the text; cf. Eq. (3). Hence, when taking a random occurrence of word w, we shall see w appearing in chapter c with probability $m_w(c)/N_w$. Likewise, $sV_w(s)/N_w$ is the probability w will appear in a chapter, where w is encountered s times.

It appears that quantities deduced from $m_w(c)/N_w$ do not lead to useful predictions concerning keywords. In particular, this concerns the entropy $-\sum_{c=1}^{N_{\text{chap}}} \frac{m_w(c)}{N_w} \ln \frac{m_w(c)}{N_w}$ and correlation function $\sum_{c_1,c_2=1}^{N_{\text{chap}}} |c_1 - c_2| m_w(c_1) m_w(c_2)$ together with some of its generalizations. In contrast, the following mean [cf. Eq. (6) with Eq. (19)]:

$$\sum\nolimits_{s \geq 0} \frac{s^2 V_w(s)}{N_w}, \tag{19}$$

related to $sV_w(s)/N_w$ predicts sufficiently many global keywords; see Table 6. Similar results are found upon using the entropy $-\sum_{s \geq 0} \frac{sV_w(s)}{N_w} \ln \frac{sV_w(s)}{N_w}$ instead of Eq. (19). This formula is calculated for each word and then words with the largest value of Eq. (19) are selected. For *Anna Karenina*, at least the first 35–36 words selected in this way are keywords. Minor exclusions are seen in Table 6, which also shows that this method is much better than YAKE both in quantity and quality of keyword extraction. The advantage of this chapter-based method is that it does not depend on random permutations. Hence, it will be easier in practical implementations. The drawbacks are seen above: it depends on the existence of sufficiently many chapters (hence, it certainly does not apply to texts with a few or no chapters), and it addresses only some of the keywords.

Table 6. First column: 36 words from *Anna Karenina* that have the highest score of YAKE (Campos *et al.* 2018, 2020). Keywords are indicated by the number of their group; see Table 4. Among 36 words, there are 25 non-keywords. Keywords refer mostly to group $^{(1)}$. Second column: 36 words of *Anna Karenina* extracted via looking at the distribution of words over chapters, that is, at the largest value of Eq. (19). Only 2 words out of 36 are not keywords. Several keyword groups are represented

36 words having largest score of YAKE	36 words having largest values of Eq. (19)
levin$^{(1)}$, anna$^{(1)}$, vronsky$^{(1)}$, alexey$^{(1)}$, kitty$^{(1)}$, stepan$^{(1)}$, hand, alexandrovitch$^{(1)}$, smile, thought, arkadyevitch$^{(1)}$, time, love, face, eye, felt, man, feel, talk, life, answer, day, wife, begin, long, knew, turn, child, sergey$^{(1)}$, husband, work, princess$^{(8)}$, room, ivanovitch$^{(1)}$, people, woman	levin$^{(1)}$, alexey$^{(1)}$, alexandrovitch$^{(1)}$, varenka$^{(2)}$, vronsky$^{(1)}$, kitty$^{(1)}$, doctor$^{(8)}$, stepan$^{(1)}$, scythe$^{(6)}$, anna$^{(1)}$, arkadyevitch$^{(1)}$, marsh$^{(6)}$, countess$^{(8)}$, katavasov$^{(2)}$, priest$^{(9)}$, darya$^{(1)}$, veslovsky$^{(2)}$, alexandrovna$^{(1)}$, seryozha$^{(1)}$, mare$^{(6)}$, sviazhsky$^{(2)}$, mihailov$^{(2)}$, brother, dolly$^{(1)}$, grass$^{(6)}$, sergey$^{(1)}$, princess$^{(8)}$, mow$^{(6)}$, marshal$^{(7)}$, konstantin$^{(2)}$, ivanovitch$^{(2)}$, peasant$^{(6)}$, lidia$^{(1)}$, sick, petritsky$^{(2)}$

7. Discussion and conclusion

We proposed a method for extracting keywords from a single text. The method employs a spatial structure in word distribution. Our unsupervised method extends previous proposals, applies to a single text (i.e., it does not need databases), and demonstrates two pertinent applications: extracting the main topics of the text and separating between local and global keywords. For long texts, our analysis confirms that such a separation is semantically meaningful; see (Scott and Tribble 2006; Carpena *et al.* 2016) for related ideas about various types of keywords. The method was illustrated in several classic literature texts; see Table 1. In researching the performance of our method, we relied on expert evaluation of keywords (that show from moderate to substantial inter-annotator agreement). As expected, the agreement is better for global keywords. An important aspect of our method is that allows to extract topics of text by looking at keywords. In particular, we focused on the analysis of topics of *Anna Karenina* and *Heart of Dog*.

Both in terms of precision and recall, our method outperforms several existing methods for keyword extraction. These methods include LUHN (Luhn, 1958), YAKE (Campos *et al.* 2018 2020) (which outperforms graph-based methods), RAKE (Rose *et al.* 2010), KEA (Witten *et al.* 1999), WINGNUS (Nguyen and Luong 2010), and KeyBERT (Devlin *et al.* 2018). We mostly compare with YAKE and LUHN, because these are our closest competitors. For sufficiently long texts, the advantage of our method compared to LUHN and YAKE is both quantitative (since it extracts $2 - 3$ times more words than LUHN and YAKE), and qualitative, because LUHN and YAKE do not extract the topics, do not distinguish between local and global keywords, and do not have efficient ranking of keywords. For shorter text, only one advantage persists: our method helps to extract topical groups; see Table 1 and Table 5. We show that our method generally extracts more nouns and longer content words than YAKE and LUHN. There are correlations between these features and the features of being a keyword; Piantadosi *et al.* (2012) and Firoozeh *et al.* (2020) provide additional arguments along these lines. Our method is also language-independent, as we checked with several translations of the same text. It shares this advantage with LUHN and YAKE.

We demonstrated that our method of identifying topical groups of texts (where human annotators employ keywords extracted by our method) produced significantly better results than algorithmic methods of topic modeling such as Non-Negative Matrix Factorization, Truncated SVD, and LDA (Latent Dirichlet Allocation); see Appendix A.

We also worked out a method for keyword extraction that uses the fact that a text has sufficiently many chapters. This method is easy to implement and works better than LUHN and YAKE, but it is inferior to the previous one. However, we believe this method does have the potential for further development, for example, for clarifying the conceptual meaning of keywords and relating it with higher-order textual structures.

Our method is targeted to extract local and global keywords. Their spatial distribution moments have different behavior with respect to random permutations; see Eqs. (8–10). For short texts, the difference between local and global keywords is blurred; see Figure 2. Put differently, local and global keywords are not so different from each other. Hence, our method is not efficient (or more precisely nearly as efficient as YAKE and LUHN) in extracting keywords from short texts; see Table 1 . We are currently searching for modifications of our method that can outperform those two methods also for shorter texts. We believe that this should be feasible because our method was able to extract topical groups of a short text; see Table 5.

Future work will include adding n-gram analysis functionality to extract not only single words but also phrases of two or more words from a text. In particular, these keyphrases are important for reflecting the aspects that were not studied in this work, that is, relations of keywords to the style of the text. Brin and Page (1998); Papagiannopoulou and Tsoumakas (2020); Katz (1996) provide useful information on how to find keyphrases from keyword extractions. Another useful idea for extracting keyphrases is to study co-occurrences between candidate keywords; see, for example, Matsuo and Ishizuka (2004) for this technique. A more remote but important application will be to employ keywords for facilitating text compression methods (Allahverdyan and Khachatryan 2023).

The present work leaves open the issue of finding an adequate model for random text so that the implementation of a random permutation (on which our method relies) is not needed; see section 3.2. We were not able to employ theoretical models of a random text proposed by Yngve (1956), Herdan (1966), Herrera and Pury (2008), and Mehri and Darooneh (2011) because these models frequently imply asymptotic limits that are not reached for keywords. Further ideas about modeling the random gap distribution in non-asymptotic situations were proposed by Zörnig (1984, 2010), Carpena *et al.* (2016), and Allahverdyan *et al.* (2013). We plan to address them in the future. Ideally, once the proper random model for the gap distribution is understood, random permutations will not be needed.

Acknowledgments. This work was supported by SCS of Armenia, grant No. 21AG-1C038. It is a pleasure to thank Narek Martirosyan for participating in initial stages of this work. We acknowledge discussions with A. Khachatryan, K. Avetisyan, and Ts. Ghukasyan. We thank S. Tamazyan for helping us with French texts.

References

Alami Merrouni Z., Frikh B. and Ouhbi B. (2020). Automatic keyphrase extraction: a survey and trends. *Journal of Intelligent Information Systems* **54**, 391–424.

Allahverdyan A. and Khachatryan A. (2023). Optimal alphabet for single text compression. *Information Sciences* **621**, 458–473.

Allahverdyan A. E., Deng W. and Wang Q. A. (2013). Explaining zipf's law via a mental lexicon. *Physical Review E* **88**, 062804.

Ando R. K. and Zhang T. (2005). A framework for learning predictive structures from multiple tasks and unlabeled data. *Journal of Machine Learning Research* **6**, 1817–1853.

Bondi M. and Scott M. (2010). *Keyness in Texts*, vol. 41. John Benjamins Publishing.

Boudin F. (2013). A comparison of centrality measures for graph-based keyphrase extraction. In Proceedings of the sixth international joint conference on natural language processing, pp. 834–838.

Bougouin A., Boudin F. and Daille B. (2013). Topicrank: Graph-based topic ranking for keyphrase extraction. In International joint conference on natural language processing (IJCNLP), pp. 543–551.

Brainerd B. (1976). On the markov nature of text. *Linguistics* **176**, 5–30.

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

Campos R., Mangaravite V., Pasquali A., Jorge A., Nunes C. and Jatowt A. (2020). Yake! keyword extraction from single documents using multiple local features. *Information Sciences* **509**, 257–289.

Campos R., Mangaravite V., Pasquali A., Jorge A. M., Nunes C. and Jatowt A. (2018). Yake! collection-independent automatic keyword extractor. In European Conference on Information Retrieval, Springer, pp. 806–810.

Carpena P., Bernaola-Galván P., Hackenberg M., Coronado A. and Oliver J. (2009). Level statistics of words: finding keywords in literary texts and symbolic sequences. *Physical Review E* **79**, 035102.

Carpena P., Bernaola-Galván P. A., Carretero-Campos C. and Coronado A. V. (2016). Probability distribution of intersymbol distances in random symbolic sequences: applications to improving detection of keywords in texts and of amino acid clustering in proteins. *Physical Review E* **94**, 052302.

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

Cook R. J. (2005). Kappa, *Encyclopedia of Biostatistics*. John Wiley & Sons, Ltd. https://doi.org/10.1002/0470011815.b2a04024.

Deng W., Xie R., Deng S. and Allahverdyan A. E. (2021). Two halves of a meaningful text are statistically different. *Journal of Statistical Mechanics: Theory and Experiment* **2021**, 033413.

Devlin J., Chang M.-W., Lee K. and Toutanova K. (2018). Bert: pre-training of deep bidirectional transformers for language understanding. In Proceedings of NaacL-HLT (Volume 1), p. 2.

Duari S. and Bhatnagar V. (2019). Scake: semantic connectivity aware keyword extraction. *Information Sciences* **477**, 100–117.

Firoozeh N., Nazarenko A., Alizon F. and Daille B. (2020). Keyword extraction: issues and methods. *Natural Language Engineering* **26**, 259–291.

Florescu C. and Caragea C. (2017). Positionrank: An unsupervised approach to keyphrase extraction from scholarly documents. In Proceedings of the 55th annual meeting of the association for computational linguistics (volume 1: long papers), pp. 1105–1115.

Gollapalli S. D., Li X.-L. and Yang P. (2017). Incorporating expert knowledge into keyphrase extraction. In Proceedings of the AAAI Conference on Artificial Intelligence (Volume 31, No. 1).

Gustafson R. (2014). *Leo Tolstoy: Resident and Stranger*. Princeton: Princeton University Press.

Hasan K. S. and Ng V. (2014). Automatic keyphrase extraction: A survey of the state of the art. In Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp.1262–1273.

Herdan G. (1966). *The Advanced Theory of Language as Choice and Chance*. Berlin: Springer.

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

Hulth A. (2003). Improved automatic keyword extraction given more linguistic knowledge. In Proceedings of the 2003 conference on Empirical methods in natural language processing, pp. 216–223.

Katz S. M. (1996). Distribution of content words and phrases in text and language modelling. *Natural Language Engineering* **2**, 15–59.

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

Liang X., Wu S., Li M. and Li Z. (2021). Unsupervised keyphrase extraction by jointly modeling local and global context. arXiv preprint arXiv: 2109.07293.

Luhn H. P. (1958). The automatic creation of literature abstracts. *IBM Journal of Research and Development* **2**, 159–165.

Martinc M., Škrlj B. and Pollak S. (2022). TNT-KID: Transformer-based neural tagger for keyword identification. *Natural Language Engineering* **28**, 409–448.

Matsuo Y. and Ishizuka M. (2004). Keyword extraction from a single document using word co-occurrence statistical information. *International Journal on Artificial Intelligence Tools* **13**, 157–169.

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

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

Mihalcea R. and Tarau P. (2004). Textrank: Bringing order into text. In Proceedings of the 2004 conference on empirical methods in natural language processing, pp. 404–411.

Miller G. A., Newman E. B. and Friedman E. A. (1958). Length-frequency statistics for written english. *Information and Control* **1**, 370–389.

Montemurro M. A. and Zanette D. H. (2010). Towards the quantification of the semantic information encoded in written language. *Advances in Complex Systems* **13**, 135–153.

Nadim M., Akopian D. and Matamoros A. (2023). A comparative assessment of unsupervised keyword extraction tools **11**, 144778–144798. https://doi.org/10.1109/ACCESS.2023.3344032.

Nguyen T. D. and Luong M.-T. (2010). Wingnus: Keyphrase extraction utilizing document logical structure. In Proceedings of the 5th international workshop on semantic evaluation, pp. 166–169.

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

Papagiannopoulou E. and Tsoumakas G. (2020). A review of keyphrase extraction. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* **10**, e1339.

Pedregosa F., Varoquaux G., Gramfort A., Michel V., Thirion B., Grisel O., Blondel M., Prettenhofer P., Weiss R., Dubourg V., Vanderplas J., Passos A., Cournapeau D., Brucher M., Perrot M. and Duchesnay E. (2011). Scikit-learn: machine learning in python. *Journal of Machine Learning Research* **12**, 2825–2830.

Piantadosi S. T., Tily H. and Gibson E. (2012). The communicative function of ambiguity in language. *Cognition* **122**, 280–291.

Robertson S. (2004). Understanding inverse document frequency: on theoretical arguments for IDF. *Journal of Documentation* **60**(5), 503–520.

Rose S., Engel D., Cramer N. and Cowley W. (2010). *Automatic keyword extraction from individual documents*. (eds), *Text Mining: Applications and Theory*, chapter 1, pp. 1–20. https://doi.org/10.1002/9780470689646.ch1.

Schopf T., Klimek S. and Matthes F. (2022). Patternrank: leveraging pretrained language models and part of speech for unsupervised keyphrase extraction. https://arxiv.org/abs/2210.05245.

Schütze H., Manning C. D. and Raghavan P. (2008). *Introduction to Information Retrieval*, vol. **39**. Cambridge: Cambridge University Press.

Scott M. and Tribble C. (2006). *Textual Patterns: Key Words and Corpus Analysis in Language Education*, vol. **22**. John Benjamins Publishing.

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

Škrlj B., Repar A. and Pollak S. (2019). Rakun: Rank-based keyword extraction via unsupervised learning and meta vertex aggregation. In Statistical Language and Speech Processing: 7th International Conference, SLSP 2019, Ljubljana, Slovenia: Springer, vol **Proceedings 7**, pp. 311–323, October 14-16, 2019

Song M., Song I.-Y. and Hu X. (2003). Kpspotter: a flexible information gain-based keyphrase extraction system. In Proceedings of WIDM2003

Sparck Jones K. (1972). A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation* **28**(1), 11–21.

Tolstoy L. (2013). *Anna Karenina*. CreateSpace Independent Publishing. translated by C. Garnett. ISBN 10: 1461195195 / ISBN 13: 9781461195191. https://www.gutenberg.org/files/1399/1399-h/1399-h.htm.

Tsvetkov A. and Kipnis A. (2023). Entropyrank: unsupervised keyphrase extraction via side-information optimization for language model-based text compression. In *ICML 2023 Workshop Neural Compression: From Information Theory to Applications*. https://openreview.net/pdf?id=WCTtOfIhsJ.

Turney P. D. (2003). Coherent keyphrase extraction via web mining. In In Proceedings of IJCAI-0, pp. 434–439.

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

Witten I. H., Paynter G. W., Frank E., Gutwin C. and Nevill-Manning C. G. (1999). Kea: Practical automatic keyphrase extraction. In Proceedings of the fourth ACM conference on Digital libraries, pp. 254–255.

Yngve V. (1956). Gap analysis and syntax. *IRE Transactions on Information Theory* **2**, 106–112.

Zhou H. and Slater G. W. (2003). A metric to search for relevant words. *Physica A: Statistical Mechanics and its Applications* **329**, 309–327.

Zipf G. K. (1945). The repetition of words, time-perspective, and semantic balance. *The Journal of General Psychology* **32**, 127–148.

Zörnig P. (1984). The distribution of the distance between like elements in a sequence I. *Glottometrika* **6**, 1–15.

Zörnig P. (2010). Statistical simulation and the distribution of distances between identical elements in a random sequence. *Computational Statistics & Data Analysis* **54**, 2317–2327.

Appendix A. Software employed

– Our data for texts from Table 1 is freely available at https://github.com/LidaAleksanyan/keywords_extraction_data/tree/master. Our codes are available at https://github.com/LidaAleksanyan/spatial_keyword_extraction.

– Natural Language Toolkit (NLTK) is available at https://github.com/nltk/nltk.

– We used the following Python package with YAKE algorithm implementation: https://pypi.org/project/yake/.

– We used the following Python package for KeyBERT: https://pypi.org/project/keybert/

– For French and Russian texts we used (resp.) French LEFFF Lemmatizer https://github.com/ClaudeCoulombe/FrenchLefffLemmatizer, and A Python wrapper of the Yandex Mystem 3.1 morphological analyzer pymystem3 https://github.com/nlpub/pymystem3.

– Non-Negative Matrix Factorization (NNMF) was employed via https://scikit-learn.org/stable/modules/generated/sklearn.decomposition.NMF.html.

– LDA (Latent Dirichlet Allocation) was employed via https://radimrehurek.com/gensim/models/ldamodel.html.

– Truncated SVD was employed via https://scikit-learn.org/stable/modules/generated/sklearn.decomposition.TruncatedSVD.html.

– The software for random permutations was taken from numpy.random.permutation: https://numpy.org/doc/stable/reference/random/generated/numpy.random.permutation.html.

– The implementations of KEA and WINGNUS were taken from https://github.com/boudinfl/pke/tree/master?tab=readme-ov-file.

– The semeval2010 dataset is available from https://aclanthology.org/S10-1004/.

Appendix B. Numerical illustrations of Eq. (6)

Here, we illustrate on simple numerical examples how $C_2[w]$ defined via Eq. (6) can both increase or decrease after the distribution of words is made more homogeneous. We also illustrate that $C_1[w]$ [cf. Eq. (1)] does not distinguish between homogeneous and inhomogeneous distributions. Take in Eq. (6) the following parameters: $N=10$ and $\ell=4$. Consider the following three distributions of words and the respective values of $C_2[w]$ ($\bar{w}$ means any word different from w):

$$w\,\bar{w}\,\bar{w}\,\bar{w}\,\bar{w}\,\bar{w}\,\bar{w}\,w\,w\,w, \qquad C_2[w]=17, \qquad C_1[w]=3, \tag{20}$$

$$\bar{w}\,\bar{w}\,\bar{w}\,\bar{w}\,\bar{w}\,\bar{w}\,w\,w\,w\,w, \qquad C_2[w]=1, \qquad C_1[w]=1, \tag{21}$$

$$w\,\bar{w}\,\bar{w}\,w\,\bar{w}\,\bar{w}\,w\,\bar{w}\,\bar{w}\,w, \qquad C_2[w]=9, \qquad C_1[w]=3. \tag{22}$$

Eq. (20) shows an inhomogeneous distribution of w, where w appears both at the beginning of the text and its end. It has a large value of C_2. Eq. (21) is a strongly clustered distribution of w with the minimal value of C_2. Eq. (22) shows the homogeneous distribution of w, its value of C_2 is intermediate between the above two. It is seen that C_1 does not distinguish between (20) and (22).

Appendix C. Precision versus recall

For short texts, where the number of extracted and compared keywords is around 10–20, one employs the following standard definitions of precision and recall. For a given a text T, an annotator A extracts set of keywords $w(T,A)$. Next, an automated method M extracts a set $w(T,M)$ of candidate keywords. Let $N[w]$ be the number of elements in set w. Now define precision (Pre) and recall (Rec) for the present case as:

$$\mathrm{Pre}(T,M,A)=\frac{N[w(T,A)\cap w(T,M)]}{N[w(T,M)]}, \qquad \mathrm{Rec}(T,M,A)=\frac{N[w(T,A)\cap w(T,M)]}{N[w(T,A)]}. \tag{23}$$

Note that these definitions come from more general concepts, where the precision is defined as (relevant retrieved instances)/(all retrieved instances), while the recall is (relevant retrieved instances)/(all relevant instances). In application of these more general concepts to (23), it is assumed that all words extracted by human annotator are by definition keywords.

However, definitions in (23) need to be modified for longer texts, where the automated methods offer many (up to 100–200) keywords, possibly joined in topical groups. We do not expect that human annotators (even those knowing the text beforehand) can extract *apriori* such a large amount of keywords more or less precisely. Hence, for long texts the experiments with annotators were carried out differently. Annotator A studied all words extracted for a given text T by all the involved S methods $\{M_k\}_{k=1}^{S}$, and only after that completed his/her final list of words $\bar{w}(T,A)$. This set is larger (or equal) than the union of all keywords:

$$\cup_{k=1}^{S}\bar{w}(T,M_k,A)\subset\bar{w}(T,A), \tag{24}$$

where $\bar{w}(T, M_k, A)$ is the set of keywords identified by A in a list of potential keywords extracted from T using method M_k, and where $\bar{w}(T, A)$ can contain keywords that are not in $\cup_{k=1}^{S} w(T, M_k, A)$, though it can happen that $w(T, A) = \cup_{k=1}^{S} \bar{w}(T, M_k, A)$.

Now the analogs of (23) are defined (for a given annotator A, and a given precision method M) as:

$$\overline{\text{Pre}}(T, M, A) = \frac{N[\bar{w}(T, M, A)]}{N[w(T, M)]}, \qquad \overline{\text{Rec}}(T, M, A) = \frac{N[\bar{w}(T, M, A)]}{N[\bar{w}(T, A)]}. \tag{25}$$

Finally, remind that once recall and precision change between 0 and 1, a balanced way to account for both is to calculate their harmonic mean, which is standardly known as F1 score:

$$F1 = 2\,\frac{\overline{\text{Pre}}(T, M, A)\overline{\text{Rec}}(T, M, A)}{\overline{\text{Pre}}(T, M, A) + \overline{\text{Rec}}(T, M, A)}. \tag{26}$$

Cite this article: Aleksanyan L and Allahverdyan A. Unsupervised extraction of local and global keywords from a single text. *Natural Language Processing* https://doi.org/10.1017/nlp.2024.59