Finding Keywords of an Article in a Corpus Using Their Log Word Rank Movements

Yitong SUN[1] and Siew Ann CHEONG[2,3]

[1]National Junior College, 37 Hillcrest Road, Singapore 288913
[2]Division of Physics and Applied Physics, School of Physical and Mathematical Sciences, Nanyang Technological University, Singapore 637371
[3]Complexity Institute, Academic Building North Level 1 Section B Unit No. 7 (ABN-01B-07), 61 Nanyang Drive, Singapore 637335

Abstract

Keyword and keyphrase extraction is an important problem in informational retrieval, natural language processing, and text mining. Accurately identified keywords and keyphrases can serve as the rough summary of a text, which is increasingly important for knowledge management in a world where the numbers of technical and non-technical documents increase exponentially with time. Many methods have been proposed and tested, some relying on corpora of texts to compare against, and others do not, but nearly all rely on first filtering out 'stop words'. These are common in most languages, but in spite of their high occurrence frequencies have no meanings of their own. In this paper, we use insights derived from Zipf's Law to propose keyword and keyphrase identification methods (LWRM1 and LWRM2) based on the difference between the logarithms of the word/bigram ranks in the document and in the corpus (the log rank movement), which do not require filtering of stop words. We compare our two methods against TF-IDF and RAKE, two highly popular keyword/keyphrase extraction methods, and found that the LWRM1 methods agree very well with TF-IDF, although the keywords are discovered in different orders. We also found poor agreement between our methods and the corpus-free RAKE method. After statistical testing using Zipf's Law as the null model, we found strong correlation between the log rank movements of keywords and their statistical significance. We also checked the effects of using a larger corpus, or using a wrong corpus, on the log rank movements, before explaining why our LWRM methods are free from the bias of the TF-IDF method against rare words.

Introduction

The number of scientific publications is growing exponentially, not only in terms of overall numbers [1], but also in specific disciplines (for example, in subfields of biology [2], sports science [3], molecular phylogeny [4], social science research in South Asia [5], STEM [6], physics journals [7]) and research topics (for example, social networks [8], nanopatterned implants for drug delivery [9], noninvasive brain stimulation [10], carbon nanodots research by Chinese scientists [11], microbial antimony biochemistry [12], climate change research [13], Mediterranean forest research [14], microsimulation models in health [15]). It would be wonderful if our knowledge in science also grows exponentially. Unfortunately, this cannot happen until scientists read and critically assess publications by other scientists, and debate to reach consensus. Against this exponentially growing literature we continue to have only 24 hours a day, and 365 days a year. Recognizing this problem, publishers and researchers alike

are developing recommendation engines to help scientists navigate their research fields, but they are fighting a losing battle unless there is a more accurate and efficient way to discover the main ideas behind each paper.

This is an old problem in the fields of text mining, information retrieval, natural language processing, and have been called *keyword extraction* [16–18], *text summarization* [19–23], or *topic modeling* [24–28]. One common solution is to have experts come up with keywords and keyphrases (also called *key terms*). Indeed, in some journals authors are asked to identify a small number of keywords or keyphrases, whereas in other journals, authors must choose a small number of topical codes. Additionally, in life sciences journals the title of a paper functions like a one-sentence summary of the work done. For example, we find titles like "tumor-penetrating peptide fused EGFR single-domain antibody enhances cancer drug penetration into 3D multicellular spheroids and facilitates effective gastric cancer therapy" in the *Journal of Controlled Release*, and "the reproductive number of COVID-19 is higher compared to SARS coronavirus" in the *Journal of Travel Medicine*. With titles like these, we would be able to understand the main conclusion of the papers without reading them (although that is not our goal). Unfortunately, such a practice is not common in other disciplines. In these disciplines, we cannot know what the papers are about, beyond the few keywords or topic codes listed by the authors, unless we analyze their abstracts or their full texts.

In the literature, keyword extraction is defined in the popular textbook on text mining *Text Mining: Applications and Theory* by Berry and Kogan [29] as the "automatic identification of a set of terms that best describe the subject of a document". The *International Encyclopedia of Information and Library Science* edited by Feather and Sturges [30] defines *keyword* to be "a word that succinctly and accurately describes the subject, or an aspect of the subject, discussed in a document". According to Siddiqi and Sharan [16], "appropriate keywords can serve as a highly concise summary of a document, and help us organize documents and retrieve them based on their content". In principle, keyword extraction methods can be classified into two broad categories: (1) those based on statistical approaches, and (2) those based on machine learning approaches.

Most statistical approaches require no training data, are language-independent, and are also domain-independent. These work, as observed by Manning and Schütze in *Foundations of Statistical Natural Language Processing* [31], because

> "words do not occur in just any old order. Languages have constraints on word order. But it is also the case that words in a sentence are not just strung together as a sequence of parts of speech, like beads on a necklace. Instead, words are organized into phrases, grouping of words that are clumped as a unit. One fundamental idea is that certain groupings of words behave as constituents."

One of the earliest work in this direction was by Salton et al. in 1975, who associated the importance of a term with the number of times it appears in the text, i.e. the *term frequency* [32]. Cohen then took the next natural step in 1995, to identify frequent n-grams as keyphrases [33], while Turney used web mining techniques in 2003 to identify frequent n-grams that he called *cohesive features* [34]. Other previous works based on term specificity

included Andrade and Valencia [35], and Jones [36]. These early works attracted more works using increasingly sophisticated statistics. In 2002, Ortuño et al. observed that important words in a text tend to form clusters [37], and proposed the use of the standard deviation of distance between successive occurrences of a word as a parameter to quantify this clustering. In a 2009 follow-up paper, Carpena et al. proposed to treat these clusters of keywords as energy levels of a quantum disordered system [38], whose level spacings are significantly different from a Poisson distribution. In a 2008 paper, Herrera et al. also observed this non-uniform spatial distribution of keywords, and devised an automatic extraction procedure that tests the spatial homogeneity of such words against randomly reshuffled versions of the text [39]. Instead of long-range correlations between a candidate keyword and itself in the text, Matsuo and Ishizuka also successfully identified keywords based on co-occurrences between frequent terms [40], after testing against χ^2 distributions for significance. More recently, we also find the work by Mehri et al., whose method involves ranking words based on their non-extensive entropies, which measure the correlation ranges between their occurrences in a text [41]. In a separate 1975 paper, Salton et al. improved on their method by comparing the term frequency against the number of documents in the corpus that the given term appears in, and this eventually became the most widely used *TF-IDF (term frequency-inverse document frequency) method* of keyword extraction [42].

There is also a parallel literature using network-based approaches for automatic keyword extraction. One of the earliest work in this area is by Mihalcea and Tarau, who proposed the TextRank method [43] to extract keywords and keyphrases from the co-occurrence graph between words. This inspired many graph-based methods using different centrality measures, like those by Palshikar [44], Wang et al. [45], Litvak and Last [46], Boudin [47], Lahiri et al. [48], Litvak et al. [49], Abilhoa and de Castro [50]. In fact, the highly popular Rapid Automatic Keyword Extraction (RAKE) method proposed by Rose et al. [51] is also centrality-based, since it uses the ratio of the degree of a word over the frequency of the word as the criterion for identifying keywords. We also find network approaches based on neighborhoods and communities, like those by Wan and Xiao [52], and Grineva et al. [53]. More references can be found in the review by Sonawane and Kulkarni [54], and there is also a flurry of more recent works [55–65].

Machine learning approaches to keyword extraction are also popular. These can be unsupervised, or supervised studies incorporating linguistic knowledge. The earliest study considered an unsupervised machine learning study was by Steier and Belew [66], who used mutual information statistics to extract two-word keyphrases. Another early study was by Krulwich and Burkey, who used heuristics like italicization, the presence of phrases in section headers, and the use of acronyms to extract keyphrases from documents [67]. At around the same time, Muñoz proposed a clustering algorithm based on Adaptive Resonance Theory (ART) to discover two-word keyphrases [68]. Following this first period in the mid-1990s, we find papers using unsupervised approaches only during two other periods. The first, between 2000 and 2005, included Barker and Cornacchia, who proposed a different heuristic system for choosing noun phrases from a document as keyphrases [69]. Using the Kulback-Liebler divergence measures on phrases in multiple language models, Tomokiyo and Hurst developed a single ranking score to identify keyphrases [70]. Extracting noun phrases from a document, Bracewell et al. clustered terms having the same noun term, ranked these clusters based on the term and noun phrase frequencies, and selected top rank clusters as keyphrases for the

document [71]. The next period was from 2009 to 2012, where we again find the use of clustering techniques by two different groups, Liu et al. to extract keyphrases from meeting transcripts [72], and Liu et al. extracting keyphrases that cover the document semantically [73]. For the purpose of ranking keywords, Gazendam et al. extracted them with the help of a thesaurus [74]. Enhancing the traditional vector-space model by a graph-based syntactic representation of a document, Litvake et al. proposed the DegExt method for keyphrase extraction [75]. Finally, Bao and Deng incorporated unary, binary, and ternary grammar characteristics of the Chinese language, to extract keywords from a restricted subset of common nouns, modifiers, noun phrase, and verb phase [76].

Compared to unsupervised approaches to finding keywords and keyphrases, supervised approaches are more popular. Publishing in 2000, Peter Turney is acknowledged as the first to formulate keyphrase extraction as a supervised learning problem [77], even though Frank et al.'s paper on KEA (keyphrase extraction algorithm) preceded it by a year [78]. These early works were followed by Medelyan and Witten, who improved on the KEA algorithm by using semantic information on terms and phrases extracted from a domain-specific thesaurus [79], and other supervised learning studies making use of part-of-speech (POS) tags [80,81], using Bayesian classification [82], using conditional random field modeling [83], and building lexical chains [84,85]. More recently, we find supervised learning papers on support vector machines [86], lexical chains and semantics [87–89], and conditional random field [90].

Given the numerous and tested methods available, why do we need another keyword extraction algorithm? While some methods rely on a corpus, and others do not, nearly every method in the literature first filter out stop words, because these occur with higher frequencies than meaningful words that can potentially become keywords. An important factor that determines the performance of such methods is thus the list of stop words. As we thought about this problem of stop words, we asked how is it that human beings are so good at recognizing keywords, and are not affected by stop words. If we can detect statistical correlations (like network properties, correlations, mutual information, …) in the absence of a familiar corpus, then humans should be just as good at finding keywords of a text in an unknown language. Unfortunately, this does not seem to be true. Clearly, all of us were educated over an extended period, and thus come equipped with several built-in corpora. These vary from individual to individual, but need not be identical to generate roughly the same keywords. From this perspective, we realize that the key quality of a keyword is 'surprise'. That is, a keyword is a word that we do not expect to see in a text, or do not expect to see so many instances of.

In this paper, we introduce a simple method based on the log word rank movement that does not require initial filtering of stop words, and an extension that does the same for bigrams. We compare our methods against TF-IDF (corpus-based) and RAKE (not corpus-based), and show the expected agreement with TF-IDF in terms of the keywords discovered, but also explain why these keywords are not discovered in the same order. We show further that the bigram-based method can discover keyphrases made up of words that are not keywords. To show that the keywords and keyphrases discovered are meaningful, we test these against simple null models to demonstrate that they are discovered in more or less the same order as their statistical significance. We then compare two corpora, to show that the performance of the method improves with the size of the corpus, and with the specialization of the corpus.

Finally, we intentionally compute the rank movements of keywords and key bigrams using a different corpus, to understand how sensitive the method is to the wrong corpus.

Data and Methods

Data

For this study, we used two highly-specialized data sets, and one moderately-specialized data set. The two highly-specialized data sets consist of abstracts from the *ACM Transactions on the Web* and the *ACM Transactions on Software Engineering and Methodology*. These two data sets are not open, but can be requested from the Association of Computing Machinery. The moderately-specialized data set is the open *Reuters-21578 Text Categorization Collection Data Set*, available on the UCI Machine Learning Repository (http://archive.ics.uci.edu/ml/datasets/Reuters-21578+Text+Categorization+Collection). Details of these data sets are shown in Table 1.

Table 1. Datasets used in study.

Corpus	ACM-TWEB	ACM-TSEM	Reuters
Number of texts	224	407	18,103
Number of unique words	6,183	7,148	74,634
Number of tokens	54,635	85,543	2,670,066

All three data sets are clean, and may be used as is after sorting out minor encoding problems.

Zipf's Law and Log Rank Movement

In 1935, Zipf wrote in his book *The Psychobiology of Language* [91] that for a given corpus, the frequency of the $(n+1)$th-ranked word is half that of the nth-ranked word, and this regularity was observed in German, Chinese, Latin, American English, and many other languages. Although there were others before him who also noted this regularity, because of Zipf's efforts popularizing it this came to be known as *Zipf's Law*,

$$f(n) \propto n^{-1},$$

which states that the frequency $f(n)$ of a word is inversely proportional to its rank n.

Naturally, because this is a statistical statement, we do not expect a given word to appear the same number of times in different texts, even if these have the same total number of tokens. At best, if we tabulate the number of times the word appear over a large number of texts, we expect this frequency distribution to be compatible with it being sampled randomly from a corpus with probability $P(n) = A/n$. We can also rank the words in a short text where the given word appears. The rank n' of the word in the short text can be different from n, its rank in the corpus. If the short text is sampled randomly from the corpus, then statistically we would expect $P'(n') = A'/n'$. The normalization constants A' and A are different, because of the different sizes of the short text and the corpus.

If we compare the probabilities by taking the ratio

$$\frac{P'(n')}{P(n)} = \frac{A'}{A} \cdot \frac{n}{n'},$$

and then taking the logarithm,

$$\log_{10} \frac{P'(n')}{P(n)} = \log_{10} \frac{A'}{A} + \log_{10} n - \log_{10} n',$$

we realize that $\log_{10} \frac{A'}{A}$ represents a constant offset for all words, while the log rank movement

$$\Delta_1 = \log_{10} n - \log_{10} n'$$

tells us how closely the given word follow Zipf's Law in the short text. If the rank n' of the word in the short text is consistent with what we expect from the rank n the word has in the corpus, $\Delta_1 \approx 0$. We expect this to be the case for stop words like 'a', 'the', 'we', …. On the other hand, if the given word is *more frequent* in the short text than expected from the corpus, we would have $n' < n$, and thus $\Delta_1 > 0$. Similarly, if the given word is *less frequent* in the short text than expected from the corpus, we would have $n' > n$, and thus $\Delta_1 < 0$.

We argue that humans are good at judging whether a given word is relatively enriched, by comparing the frequency of the word against the frequencies of other words known to be nearly equally frequent in the corpus. In this way, we can probably detect two-fold enrichment, when we expect to see two instances of the word because other words known to be equally frequent in the corpus appeared twice on average, but counted four instances of the word instead in the short text. We will definitely not miss a ten-fold enrichment, with $\Delta_1 = 1.0$, where we expect to see one instance, but counted 10 instances instead! As many before us have argued, enriched words are keywords linked to the topic(s) of the short text. We will ignore depleted words, as they are not keywords, but provide information on which topics are 'orthogonal' to that represented by the enriched words. The use of the log rank movement $\Delta_1 = \log_{10} n - \log_{10} n'$ can therefore be used to identify keywords, *without* first having to filter out stop words. Let us call this method LWRM1.

Now, although we do not think humans can sense long-range and complex correlations, we find it believable that they can make quantitative estimates of short-range correlations in short n-grams. In particular, for bigrams we would be able to judge how likely or unlikely two words can appear next to each other through familiarity with the corpus. For a given bigram, if m is its rank in the corpus, and m' is its rank in the short text, the log rank movement

$$\Delta_2 = \log_{10} m - \log_{10} m'$$

should allow us to identify enriched bigrams as keyphrases. Let us call this method LWRM2.

Results

Zipf's Laws for Words and Bigrams

We demonstrate the utility of our log rank movement methods by applying them to the *ACM Transactions on the Web (ACM-TWEB)* dataset (6,183 distinct words, 54,635 tokens). We

consider this a highly-technical corpus where most of its common words are rarely used in daily life. First, let us check if words and bigrams obey Zipf's Law(s), by plotting their frequencies against their ranks in Figure 1. As we can see, the fits of both frequency-rank plots to power laws are decent, with expected deviations for the most frequent words/bigrams, as well as for the least frequent words/bigrams. Linear regressions over the smoothest $10 \leq n \leq 1000$ region gave an exponent of $\alpha_1 = -0.898472 \pm 0.000057$ for words, and an exponent of $\alpha_2 = -0.686412 \pm 0.000052$. These are significantly different from $\alpha = -1$ for the standard Zipf's Law, but are compatible with contemporary forms $f(n) \propto n^{-\alpha}$ for Zipf's Law [92]. More importantly, α_2 is significantly smaller than α_1. If bigrams are constructed by randomly sampling pairs of words from the distribution of words, the exponent should remain close to α_1. There must therefore be strong correlations between words that make up actual bigrams, for the exponent to have drop by so much.

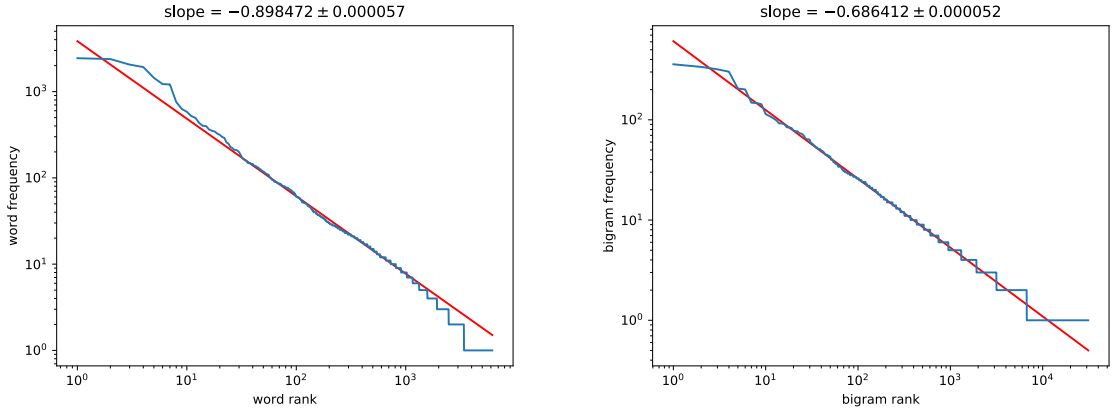


Figure 1. Frequency-rank plots of (left) words and (right) bigrams from the *ACM Transactions on the Web* corpus of abstracts. In these plots, we also show the best fits to power laws, for ranks $10 \leq n \leq 1000$.

Case Study of Log-Rank-Movement Keywords, Key Bigrams, and Stop Words

Before we test how well the log-rank-movement methods compare against existing methods (TF-IDF and RAKE), and how good the methods are at identifying keywords and key bigrams, let us examine the keywords and key bigrams identified for a specific abstract in the ACM-TWEB corpus. We start by checking if the log rank movements of stop words are close to zero as we expected. Indeed, for stop words that appear more than once in the abstract highlighted in Table 2, many of them have $\Delta_1 \approx 0$, like 'with' (0.09), 'from' (0.04), 'The' (0.02), 'and' (0.00), 'We' (-0.02). There are also stop words, like 'for' (0.52), 'are' (0.28), which have slightly positive Δ_1s, and those, like 'as' (-0.28), 'the' (-0.30), 'we' (-0.39), 'a' (-0.50), 'that' (-0.51), 'of' (-0.70), which have slightly negative Δ_1s. Indeed, in this case study no initial filtering of stop words was necessary.

Table 2. Keywords identified by the two log rank movement methods, compared to those identified by RAKE and TF-IDF. In this table, only words and bigrams appearing more than once in the abstract are included. We implemented the TF-IDF algorithm ourselves, while the RAKE keyphrases are identified using the Rake class (which uses stop words for English from NLTK, and all punctuation characters) in the rake_nltk Python module. After instantiating the Rake class `R = Rake(2,2)`, we use the function `R.extract_keywords_from_text(…)`.

LWRM1	LWRM2	RAKE	TF-IDF
sponsored: 2.85, nonsponsored: 2.07, links: 1.59, relevance: 1.48, e-commerce: 1.40, analyzed: 1.35, %: 1.25, relevant: 1.18, engines: 1.17, campaigns: 1.14, major: 1.12, average: 1.03, business: 0.93, queries: 0.92, ratings: 0.86, search: 0.70, …	‘sponsored search’: 2.84, sponsored and’: 2.79, ‘sponsored links’: 2.75, ‘of sponsored’: 2.74, ‘business model’: 2.64, ‘and nonsponsored’: 2.64, ‘for sponsored’: 2.57, ‘e-commerce queries’: 2.54, ‘links for’: 2.52, ‘relevance ratings’: 2.51, ‘nonsponsored links’: 2.45, ‘analyzed the’: 2.41, ‘search campaigns’: 2.39, ‘ratings for’: 2.35, ‘links are’: 2.34, ‘%)’: 2.30, ‘the relevance’: 2.01, ‘search engines’: 1.88, ‘links from’: 1.84, ‘for Web’: 1.79, ‘Web search’: 1.73, …	'yearly revenue', 'various viewpoints', 'statistically higher', 'sponsored search', 'sponsored links', 'specific queries', 'results show', 'relevant choices', 'relevance ratings', 'relevance measures', 'related issues', 'qualitatively analyzed', 'primary basis', ‘nonsponsored links', 'generates billions', 'distant third', 'commerce queries', …	sponsored links search queries relevance engines We relevant Web campaigns information analyzed ratings average major e-commerce nonsponsored model business The

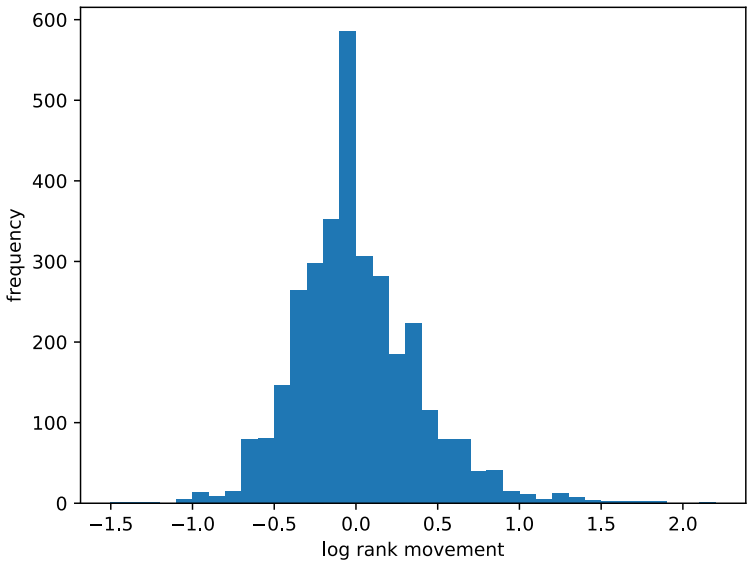


Figure 3. Histogram of the log rank movements of a list of 179 stop words comprising articles, pronouns, common verbs like 'am', 'is', 'are', …, and prepositions.

To check whether this holds for the corpus, we compiled a list of 179 stop words, comprising articles like 'a', 'the', …, pronouns like 'I', 'we', 'they', …, common verbs like 'am', 'is', 'are', …, prepositions like 'at', 'for', 'of', …, and extracted their log rank movements from abstracts in the ACM-TWEB corpus. As we find from the distribution shown in Figure 3, the log rank movements of these stop words are concentrated about $\Delta_1 = 0$, although some go beyond $\Delta_1 = 1.0$ in a small fraction of abstracts. We believe this will not affect our identification of keywords.

Moving on to our comparisons, we see from Table 2 that the LWRM1 keywords and the TF-IDF keywords are largely the same, but the orders of the keywords are different. For example, 'sponsored' is the first keyword for both methods, but while 'nonsponsored' is the second LWRM1 keyword, it is the 17th TF-IDF keyword. Similarly, 'search' is the third TF-IDF keyword, but is the 16th LWRM1 keyword. At this stage, we did not know what is an appropriate threshold to use for Δ_1, and thus can keep fewer or more of the LWRM1 keywords. The same problem exists for the TF-IDF method. If we keep fewer words, we have few matches, but if we keep more words, we may have more matches. To be systematic, we measure the proportion of matches as a function of the number of keywords kept. From Figure 2, we see that the proportion p of matches start high (because the first words coincide), then fluctuate between 0.5 and 0.6, before climbing to a maximum of 0.82, and falling off again. This maximum corresponds to 16 matching words.

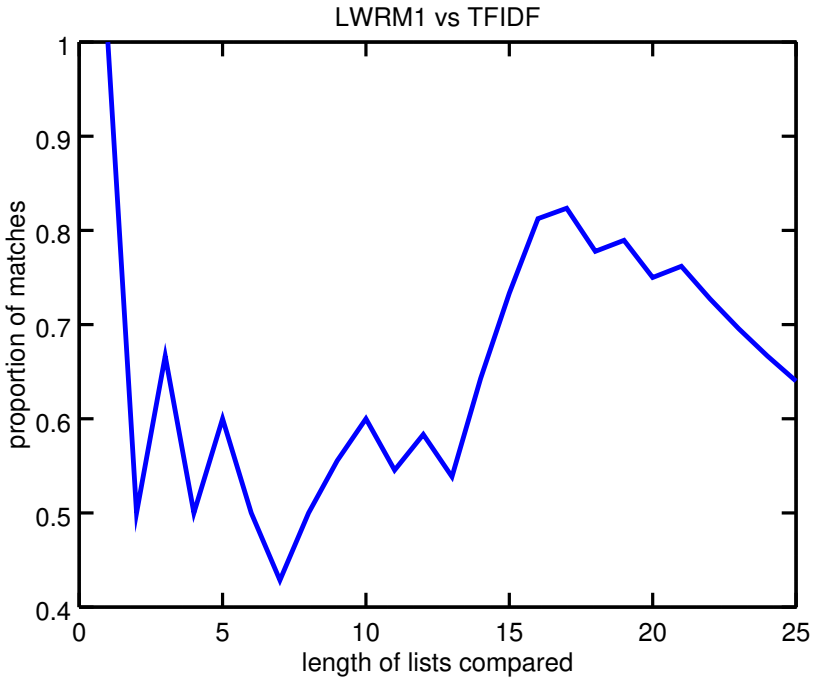


Figure 2. Proportion of matching keywords between LWRM1 and TF-IDF, plotted against the number of keywords kept.

Later in the Discussion section we will explain why the LWRM1 keywords agree so well with the TF-IDF keywords, if we ignore the orders they are discovered. But is the order of the keywords important? If it is we should compute the correlations between the two lists. In fact, the Pearson correlation between the two lists is $C = 0.467$, while the Kendall tau correlation between them is $\tau = 0.333$. Both correlations are not high. In the Discussion section we will also explain why this order is not important, because of a bias in the TF-IDF method.

We also compared the performances of our two methods, LWRM1 and LWRM2. We do so in two ways. For a word-based comparison, we again compare two lists of the same length from LWRM1 and LWRM2. For LWRM2, we decompose the list of bigrams into a set of words. For example, {'sponsored', 'nonsponsored', 'links', 'relevance', 'e-commerce'} are the first five LWRM1 keywords, whereas the first five LWRM2 key bigrams are {'sponsored search', 'sponsored and', 'sponsored links', 'of sponsored', 'business model'}. This corresponds to the set of words {'sponsored', 'search', 'and', 'links', 'of', 'business', 'model'}. Therefore, for these length-5 lists, we find two matches, 'sponsored' and 'links', corresponding to $p = 0.4$. If we now go to lists of length 10, we have {'sponsored', 'nonsponsored', 'links', 'relevance', 'e-commerce', 'analyzed', '%', 'relevant', 'engines', 'campaigns'} from LWRM1, and {'sponsored search', 'sponsored and', 'sponsored links', 'of sponsored', 'business model', 'and nonsponsored', 'for sponsored', 'e-commerce queries', 'links for', 'relevance ratings'} from LWRM2, which gives the set of words {'sponsored', 'search', 'and', 'links', 'of', 'business', 'model', 'nonsponsored', 'for', 'e-commerce', 'queries', 'relevance', 'ratings'}. For these length-10 lists, we find five matches, corresponding to $p = 0.5$. By varying the number of keywords and key bigrams kept, we can find the maximum proportion of matches $p_{\max}$.

For a bigram-based comparison, we say that a given bigram has a match of 0.5 with a list of keywords if one of its two constituent words can be found in the list, and that it has a match

of 1.0 with the list of keywords if both its constituent words can be found in the list. For example, if we compare the first 5 keywords {'sponsored', 'nonsponsored', 'links', 'relevance', 'e-commerce'} from LWRM1, and the first 5 key bigrams {‘sponsored search’, ‘sponsored and’, ‘sponsored links’, ‘of sponsored’, ‘business model’} from LWRM2, we find the cumulative matches at the bigram level shown in Table 3. Here, let us also highlight ‘business model’, which is the fifth most important LWRM2 bigram, because ‘business’ and ‘model’ are individually not LWRM1 keywords (although they are ranked reasonably high by TF-IDF).

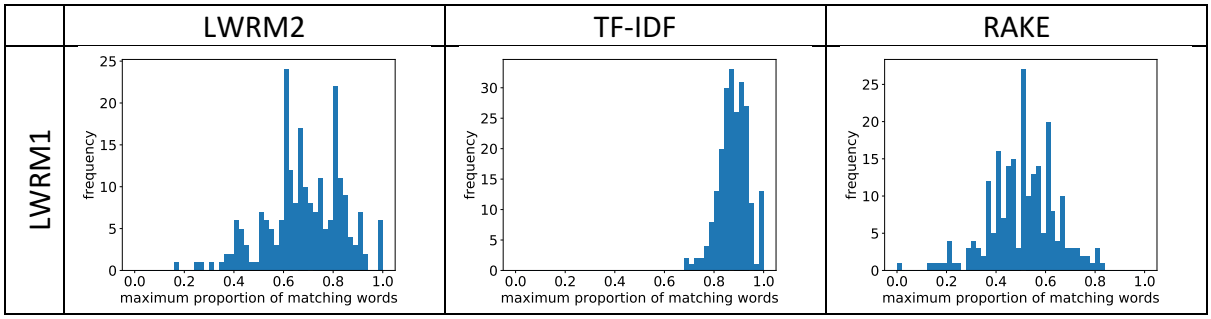
Table 3. Cumulative matches for bigram-based comparison between LWRM1 and LWRM2.

Length	1	2	3	4	5
LWRM2	‘sponsored search’	‘sponsored and’	‘sponsored links’	‘of sponsored’	‘business model’
LWRM1	'sponsored'	'nonsponsored'	'links'	'relevance'	'e-commerce'
Cumulative Matches	0.5	0.5 + 0.5 = 1.0	0.5 + 0.5 + 1.0 = 2.0	0.5 + 0.5 + 1.0 + 0.5 = 2.5	0.5 + 0.5 + 1.0 + 0.5 + 0.0 = 2.5

Instead of doing a word-based or bigram-based comparison between the LWRM1 keywords and the RAKE key bigrams, a word-based or bigram-based comparison between the TF-IDF keywords and the LWRM2 key bigrams, and a word-based or bigram-based comparison between LWRM2 and RAKE, we move on to do systematic comparisons at the corpus level.

Corpus-Wide Comparison Between Methods

Performing pairwise word-based comparison between LWRM1, LWRM2, RAKE, and TF-IDF over the ACM-TWEB corpus to get $\{p_{\max}\}$ for the 224 abstracts, we obtain the histograms shown in Figure 3. From this figure, we see that there is very good agreement between LWRM1 and TF-IDF, because the distribution is between $0.7 < p_{\max} < 1.0$, meaning that for all abstracts more than 70% of the LWRM1 keywords match those of TF-IDF. For the word-based comparison between LWRM2 and LWRM1, we find that for most abstracts, $p_{\max} > 0.5$. This is also true for the word-based comparison between LWRM2 and TF-IDF, except for the absence of larger $p_{\max}$. In contrast, agreement between RAKE and LWRM1/TF-IDF is poor, as the distribution of $p_{\max}$ is centered around $p_{\max} = 0.5$. Finally, when we do word-based comparison between LWRM2 and RAKE, the distribution of $p_{\max}$ is largely below $p_{\max} = 0.4$. This tells us that there is very poor agreement between these two methods.



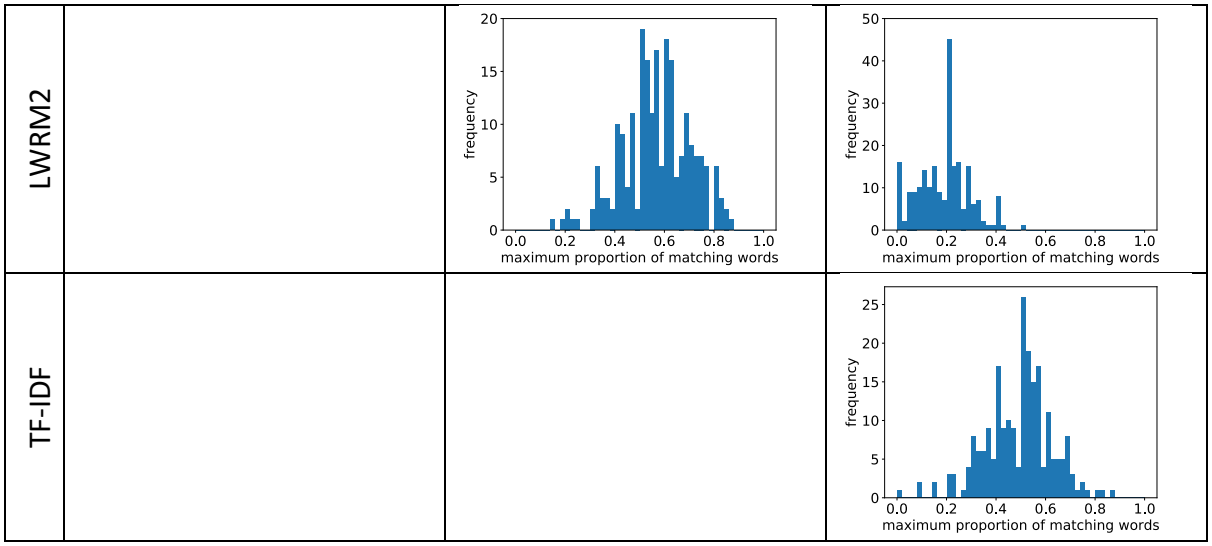


Figure 3. Histograms of $p_{\max}$ for the pairwise, word-based comparison between LWRM1, LWRM2, TF-IDF, and RAKE of the ACM-TWEB data set.

Next, after performing pairwise bigram-based comparison between LWRM1, LWRM2, RAKE, and TF-IDF over the TWEB corpus, we obtain the histograms shown in Figure 4. Here we find reasonable agreement between LWRM1/TF-IDF and LWRM2 (peak $p_{\max} \approx 0.4$), poor agreement between the two word-based methods and RAKE (peak $p_{\max} \approx 0.3$), and very poor agreement between LWRM2 and RAKE (peak $p_{\max} \approx 0.1$) at the bigram level. This is understandable, as we can already tell from Table 2 that the keyphrases identified by RAKE without reference to any corpus are less accurate.

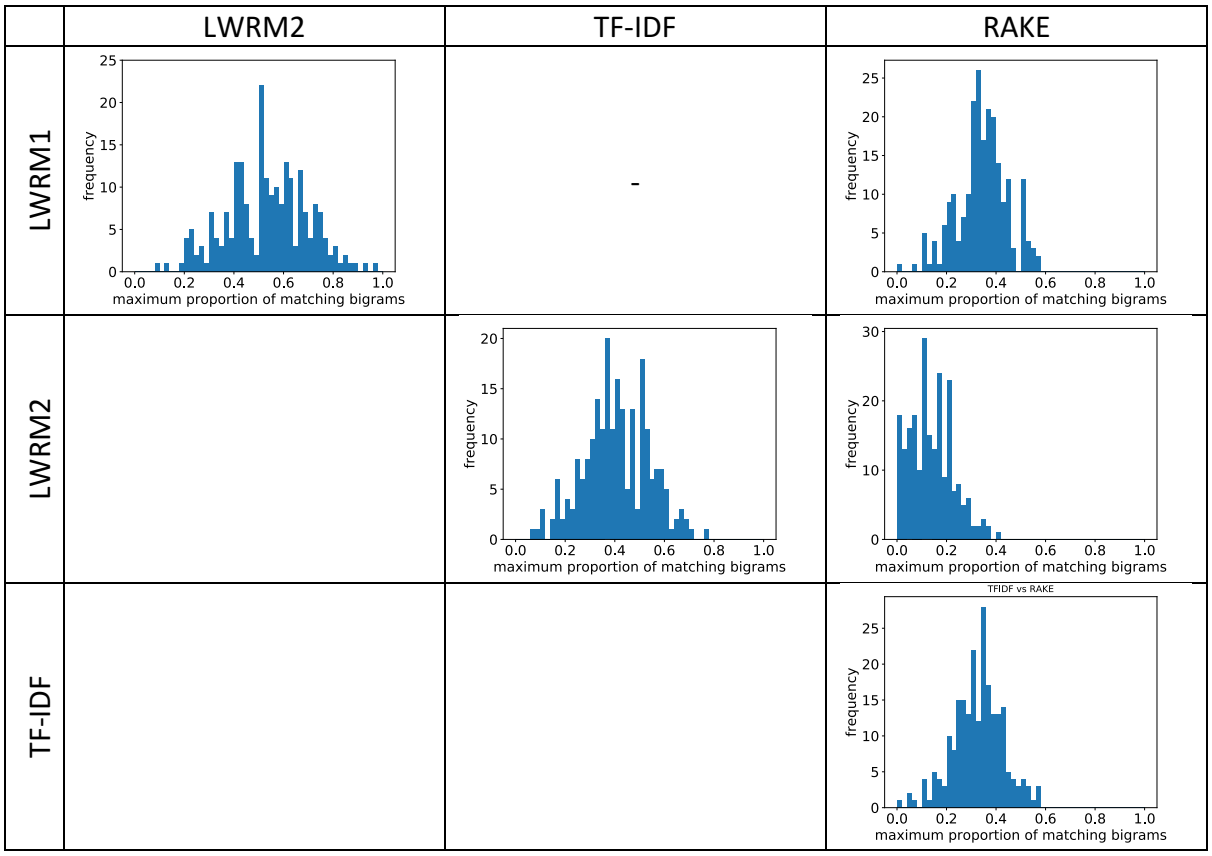


Figure 4. Histograms of $p_{\max}$ for the pairwise, bigram-based comparison between LWRM1, LWRM2, TF-IDF, and RAKE of the ACM-TWEB data set.

Test of Statistical Significance

In the Data and Methods section, we described how the numbers of times ordinary words appear in a short document (on the order of 100 distinct words) should be consistent with these words being sampled randomly from a corpus (on the order of 10,000 to 100,000 distinct words) that obeys Zipf's law. Using this as a null model, we can test the statistical significance of the M distinct words in an abstract with N tokens. Suppose we sample N tokens according to $P(n) = A/n$, where n is the rank of a distinct word in the corpus, we can then count the numbers of times $\widehat{N}_1$, $\widehat{N}_2$, …, $\widehat{N}_M$ the M distinct words appear in the N sampled tokens. Repeating this random sampling 10^4 times, we end up with histograms of frequencies $\{\widehat{N}_i\}$ for the M distinct words in the corpus. We then compare the actual numbers of times $N_1, N_2, \ldots, N_M$ these words appear in the abstract, such that $N_1 + N_2 + \cdots + N_M = N$, against the M histograms. The basic idea here is that small log word rank movements can occur by chance, but large ones must have occurred by choice, and are therefore associated with keywords.

Again, let us see how this significance testing works for the 1,427-token ACM-TWEB abstract whose keywords are shown in Table 2. Sampling 10^4 sets of 1,427 tokens without replacement from the 54,635-token corpus, we find the null-model frequencies shown in Supplementary Figure S1 for the top 16 LWRM1 keywords. For these keywords, we find the top keywords like 'sponsored' ($N_{obs} = 13, \Delta_1 = 2.85, p < 10^{-4}$), 'nonsponsored' ($N_{obs} = 2, \Delta_1 = 2.07, p = 0.0005$), 'links' ($N_{obs} = 8, \Delta_1 = 1.59, p < 10^{-4}$), 'relevance' ($N_{obs} = 4, \Delta_1 = 1.48, p = 0.0014$), 'e-commerce' ($N_{obs} = 2, \Delta_1 = 1.40, p = 0.0326$), 'analyzed' ($N_{obs} = 2, \Delta_1 = 1.35, p = 0.0263$), '%' ($N_{obs} = 4, \Delta_1 = 1.25, p = 0.0039$), 'relevant' ($N_{obs} = 3, \Delta_1 = 1.18, p = 0.0239$), 'engines' ($N_{obs} = 4, \Delta_1 = 1.17, p = 0.0142$) to be statistically significant at $p \leq 0.05$ level of confidence. All of the following keywords, 'campaigns' ($N_{obs} = 2, \Delta_1 = 1.14, p = 0.0558$), 'major' ($N_{obs} = 2, \Delta_1 = 1.12, p = 0.0867$), 'average' ($N_{obs} = 2, \Delta_1 = 1.03, p = 0.1012$), 'business' ($N_{obs} = 2, \Delta_1 = 0.93, p = 0.1732$), 'queries' ($N_{obs} = 5, \Delta_1 = 0.92, p = 0.0455$), 'ratings' ($N_{obs} = 2, \Delta_1 = 0.86, p = 0.1582$), 'search' ($N_{obs} = 9, \Delta_1 = 0.70, p = 0.0780$), were statistically insignificant ($p > 0.05$), except for 'queries' ($N_{obs} = 5, \Delta_1 = 0.92, p = 0.0455$).

As we can see for this ACM-TWEB abstract, there is strong correlation between log word rank movement and statistical significance. However, the correlation is not perfect: some keywords with smaller log word rank movements are statistically significant, while some keywords with larger log word rank movements are not statistically significant. To use the log word rank movement for keyword identification in practice, we must set a threshold and keep keywords above this threshold. Depending on the threshold, we might end up retaining keywords that are not statistically significant, or discarding keywords that are. To see how well the LWRM1 method fare in identifying statistically significant keywords, we go through all words (not just suspected keywords) that appear twice or more in the 224 abstracts, classify them as significant or insignificant for three different confidence levels, and then plot the histogram of the log word rank movements of significant words, as well as that for insignificant words.

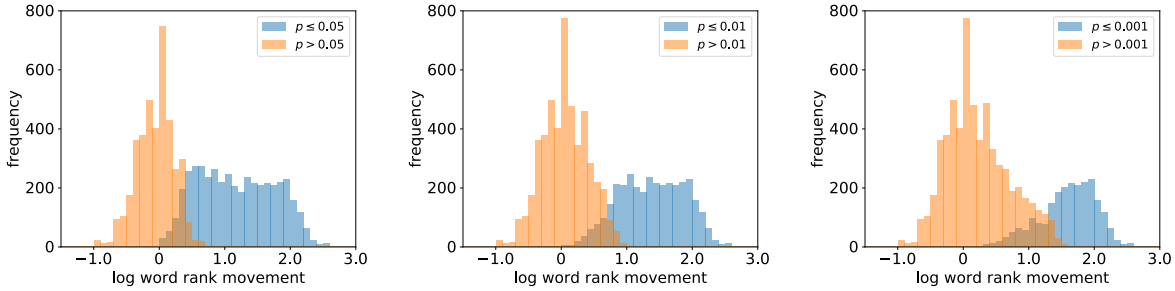


Figure 6. Distributions of log word rank movements of significant keywords (blue) versus those of insignificant keywords (orange), for the ACM-TWEB data set and $p = 0.05$ (left), $p = 0.01$ (center), $p = 0.001$ (right) levels of confidence.

From Figure 6, we see that for $p = 0.05$, the log word rank movement distributions of 4,443 significant and 3,937 insignificant keywords overlap between $0 < \Delta_1 < 0.7$, and the best threshold to discriminate between significant and insignificant keywords at the corpus level is $\Delta_1 = 0.4$. When the confidence level is changed to $p = 0.01$, the number of significant keywords decreased to 3,406, while the number of insignificant keywords increased to 4,974. The two distributions overlap between $0 < \Delta_1 < 1.0$, and the discrimination threshold must be increased to $\Delta_1 = 0.7$. Finally, when we become stricter and set $p = 0.001$, we find only 2,281 significant keywords, against 6,099 insignificant keywords. The two distributions now overlap between $0.3 < \Delta_1 < 1.6$, with an optimum discrimination threshold of $\Delta_1 = 1.3$. Ultimately, we do not need to be so strict, and can afford to include a few statistically insignificant keywords. Therefore, for this 54,635-token ACM-TWEB corpus, any choice of $0.5 < \Delta_1 < 1.0$ would be reasonable.

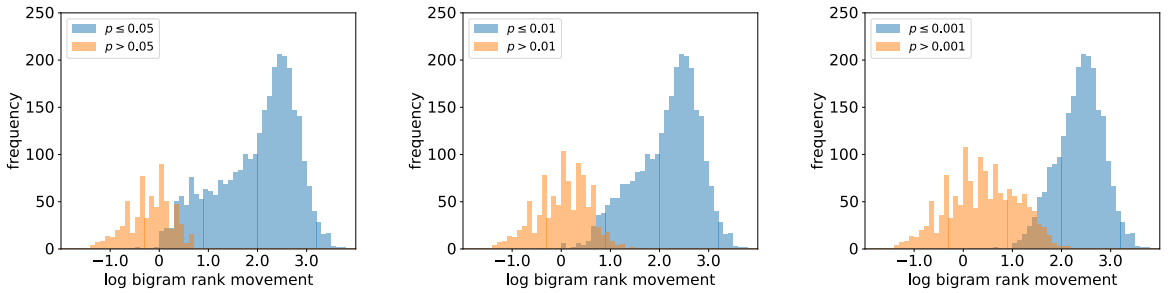


Figure 7. Distributions of log bigram rank movements of significant bigrams (blue) versus those of insignificant bigrams (orange), for the ACM-TWEB data set and $p = 0.05$ (left), $p = 0.01$ (center), $p = 0.001$ (right) levels of confidence.

We next do statistical testing for the bigrams. The ACM-TWEB corpus contains 54,411 bigrams, of which 31,185 are distinct. We admitted only 3,692 distinct bigrams that appear twice or more to be key bigrams. For $p = 0.05$, 3,035 of these are significant, while 657 are insignificant. As we can see from Figure 7, the distributions of significant and insignificant log bigram rank movements overlap between $0 < \Delta_2 < 0.6$, and the two distributions are best discriminated using a threshold of $\Delta_2 = 0.3$. For $p = 0.01$, we find 2,718 significant bigrams and 974 insignificant bigrams. The two distributions now overlap between $0 < \Delta_2 < 1.2$, and the optimum threshold is $\Delta_2 = 0.6$. Finally, for $p = 0.001$, we find 2,248 significant bigrams and 1,444 insignificant bigrams. The two distributions overlap between $1.0 < \Delta_2 < 2.1$, and the optimum threshold is $\Delta_2 = 1.5$. At the same level of confidence, we find that the overlap

between the significant and insignificant distributions is slightly smaller for bigrams than for words. As with words, because the overlap grows as we go to smaller p, we should not go for the strictest level of confidence in identifying key bigrams.

Up to this point, we have demonstrated the strong correlation between the log rank movement and the statistical significance of keywords and key bigrams, and thus the utility of this quantity for identifying keywords and keyphrases. In the next two subsections, we explore further properties of the log rank movement, specifically how it behaves for a larger corpus, or when the log rank movement for words in a text is computed using the 'wrong' corpus.

Comparison Between Corpora of Different Sizes

To see how much more or less effective the log rank movement is at identifying keywords and keyphrases for a larger corpus, we turn to the Reuters data set. This consists of 18,103 news items, containing 74,634 unique words occurring a total of 2,670,066 times. The 18,103 news items also comprise 627,524 unique bigrams, occurring 2,651,964 times. Compared to the ACM-TWEB data set, the Reuters news data set is much larger, but at the same time somewhat less technical.

We begin by checking the relationships between frequency and rank for words and bigrams in the Reuters data set. As shown in Figure 8, the frequency-rank plots of words and bigrams are better 'fitted' by exponentially-truncated power laws of the form $f(n) = An^{-\alpha}\exp(-n/n_0)$, also known as *Menzerath-Altmann's law* in quantitative linguistics [93–100]. As with the TWEB data set, the exponent $\alpha_2 \approx 0.75$ for bigrams is smaller than the exponent $\alpha_1 \approx 0.95$.

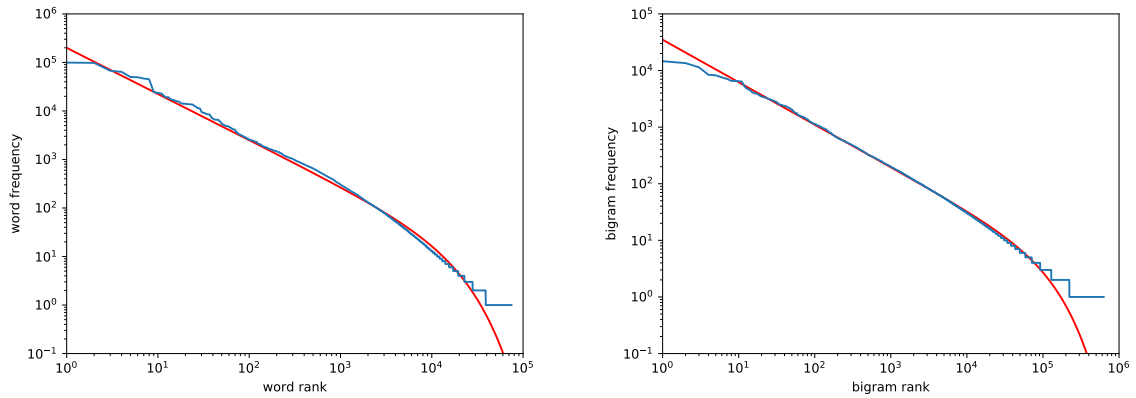


Figure 8. Frequency-rank plots of (left) words and (right) bigrams from the Reuters corpus. In these plots, we also show approximate fits to exponentially-truncated power laws, $F = An^B\exp(-n/C)$. For words, we find that $A = 2\times10^5$, $B = -0.95$, and $C = 1.5\times10^4$, whereas for bigrams, we find $A = 3.5\times10^4$, $B = -0.75$, and $C = 1.3\times10^5$.

Next, we examined how well the LWRM methods work compared to TF-IDF and RAKE. For this corpus-based comparison at the word-based and bigram-based levels, only 5,956 news items were usable. The remaining news item were too short for us to compute the maximum matching probability. Comparing Figure 9 with Figure 3, we find the same good agreements between LWRM1, LWRM2, and TF-IDF, poor agreements between LWRM1 and TF-IDF with RAKE, and very poor agreement between LWRM2 and RAKE. When we compare Figure 10

and Figure 4, we again find good agreement between LWRM1 and LWRM2, TF-IDF and generally poor agreement between LWRM1, LWRM2, TF-IDF and RAKE. For both comparisons in general, the distributions of $p_{\max}$ appeared narrower for the larger Reuters News corpus than for the smaller ACM-TWEB corpus. However, when we tested the inter-quartile ranges shown in Table 4, this observation did not appear to be statistically significant.

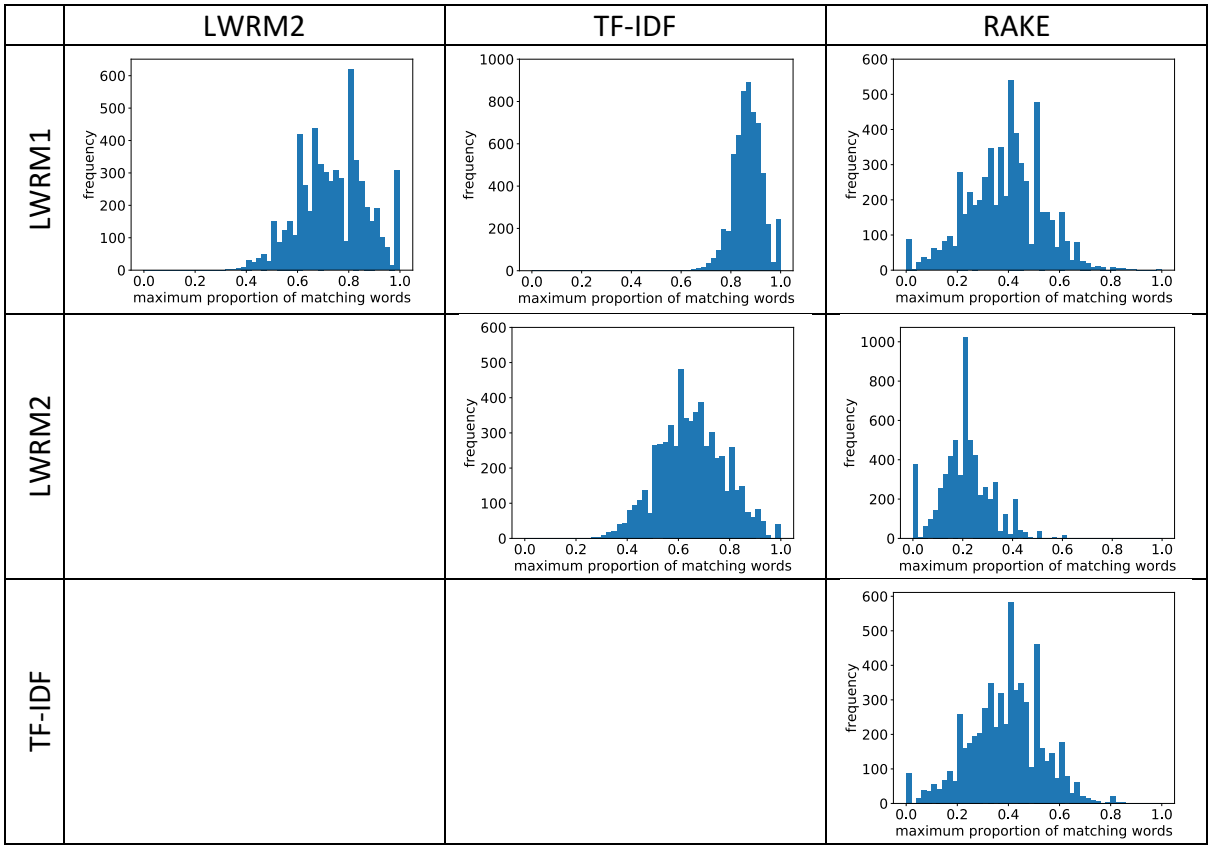
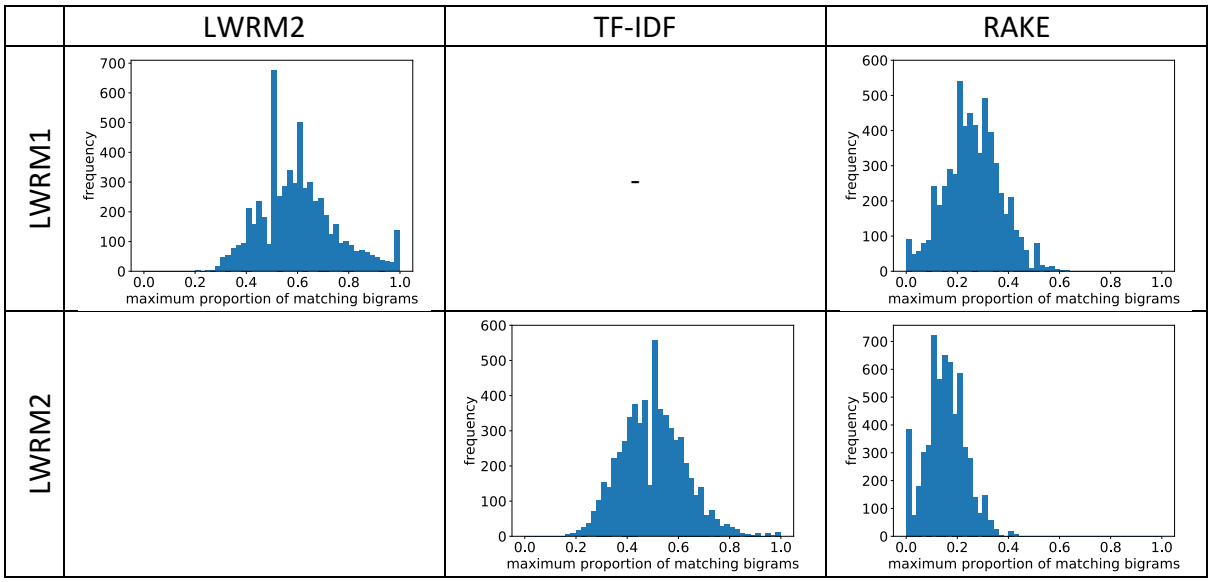


Figure 9. Histograms of $p_{\max}$ for the pairwise, word-based comparison between LWRM1, LWRM2, TF-IDF, and RAKE for the Reuters data set.



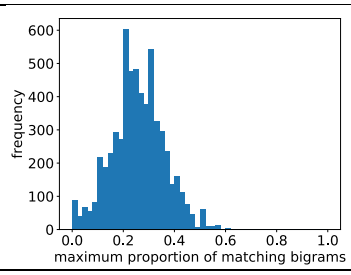
TF-IDF			

Figure 10. Histograms of $p_{\max}$ for the pairwise, bigram-based comparison between LWRM1, LWRM2, TF-IDF, and RAKE for the Reuters data set.

Table 4. Inter-quartile ranges of $p_{\max}$ for pairwise comparison between LWRM1, LWRM2, TF-IDF, and RAKE for the Reuters data set. In each cell, the number above is for word-based comparison, while the number below is for bigram-based comparison. The inter-quartile ranges for the ACM-TWEB data set is included in parentheses for comparison.

	LWRM2	TF-IDF	RAKE
LWRM1	0.194 (0.200) 0.183 (0.223)	0.073 (0.075) -	0.200 (0.178) 0.146 (0.110)
LWRM2		0.174 (0.187) 0.173 (0.177)	0.119 (0.137) 0.100 (0.131)
TF-IDF			0.185 (0.171) 0.129 (0.125)

Finally, we checked the correlation between the log rank movement and the statistical significance of keywords. For $p = 0.05$, the log word rank movement distributions of 69,743 significant and 139,369 insignificant keywords overlap between $0.3 < \Delta_1 < 1.0$, and the best threshold to discriminate between significant and insignificant keywords at the corpus level is $\Delta_1 = 0.8$. When $p = 0.01$, we found 58,893 significant keywords, and 150,219 insignificant keywords. The two distributions overlap between $0.4 < \Delta_1 < 1.4$, and the discrimination threshold must be increased to $\Delta_1 = 0.9$. Finally, when $p = 0.001$, we found only 47,308 significant keywords, against 161,804 insignificant keywords. The two distributions now overlap between $0.6 < \Delta_1 < 1.6$, with an optimum discrimination threshold of $\Delta_1 = 1.1$. Ultimately, for this 2,670,066-token Reuters corpus, we can choose any threshold within $0.8 < \Delta_1 < 1.1$ to find reliable keywords.

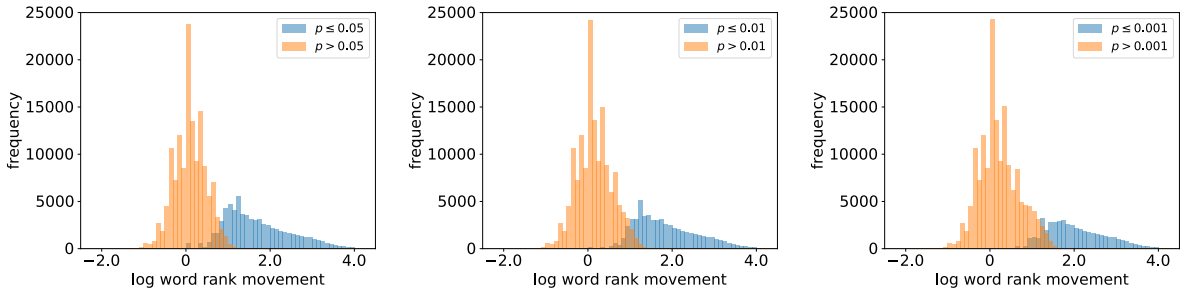
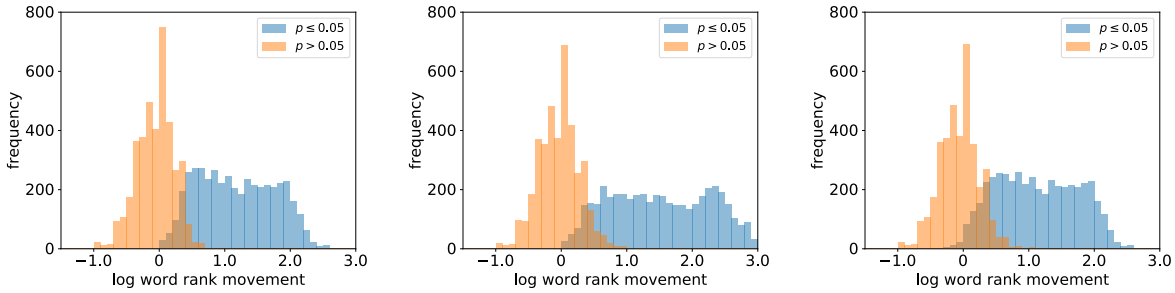


Figure 11. Distributions of log word rank movements of significant keywords (blue) versus those of insignificant keywords (orange), for the Reuters data set and $p = 0.05$ (left), $p = 0.01$ (center), $p = 0.001$ (right) levels of confidence.

Comparing the two corpora, we found that for the same level of confidence, the range of overlap between the log word rank movements of significant and insignificant keywords did not become smaller with the size of the corpus. In fact, as the size of the corpus is increased, the discrimination threshold also increased. At this point, it is not clear whether this is due to the size of the corpus, or due to the less technical nature of the Reuters corpus. Ultimately, the maximum log word rank movement is 2.5 for the ACM-TWEB corpus, but 4.0 for the Reuters corpus. Therefore, we are likely to find more statistically significant keywords for a larger corpus, even if the individual documents are roughly the same lengths.

Effect of Using the Wrong Corpus

Lastly, we investigated what happens to the log word rank movement score, if we used the wrong corpus to find keywords. To begin, let us note that there are two ways we can use a wrong corpus. The first is to use the wrong corpus (ACM-TSEM) to compute the log word rank movements in a text, but thereafter use the correct corpus (ACM-TWEB) to do statistical testing. Doing this for the ACM-TWEB abstract shown in Table 2, we found the set of statistically significant keywords at the $p \leq 0.05$ level of confidence included 'sponsored' ($N_{obs} = 13, \Delta_1 = 3.85$), 'links' ($N_{obs} = 8, \Delta_1 = 2.61$), 'nonsponsored' ($N_{obs} = 2, \Delta_1 = 2.44$), 'e-commerce' ($N_{obs} = 2, \Delta_1 = 2.42$), 'engines' ($N_{obs} = 4, \Delta_1 = 2.32$) 'relevance' ($N_{obs} = 4, \Delta_1 = 2.22$), 'queries' ($N_{obs} = 5, \Delta_1 = 1.81$), '%' ($N_{obs} = 4, \Delta_1 = 1.78$), 'relevant' ($N_{obs} = 3, \Delta_1 = 1.36$) and 'analyzed' ($N_{obs} = 2, \Delta_1 = 1.13$), while the set of statistically insignificant keywords included 'ratings' ($N_{obs} = 2, \Delta_1 = 2.36$), 'campaigns' ($N_{obs} = 2, \Delta_1 = 2.31$), 'search' ($N_{obs} = 9, \Delta_1 = 1.55$), 'major' ($N_{obs} = 2, \Delta_1 = 1.47$), 'average' ($N_{obs} = 2, \Delta_1 = 1.29$), 'business' ($N_{obs} = 2, \Delta_1 = 1.19$). As we can see, the statistically significant keywords 'ratings' and 'campaigns' have log word rank movements ($\Delta_1 = 2.36$ and $\Delta_1 = 2.31$ respectively) higher than most statistically significant keywords (false positives), while the statistically significant keywords 'relevant' and 'analyzed' have log word rank movements ($\Delta_1 = 1.36$ and $\Delta_1 = 1.13$ respectively) lower than most statistically insignificant keywords (false negatives). We therefore expect the false positive and false negative rates to increase in general when we compute the log word rank movements using the wrong corpus. This demonstrated the loss of sensitivity when we used the wrong corpus for computing log word rank movements.



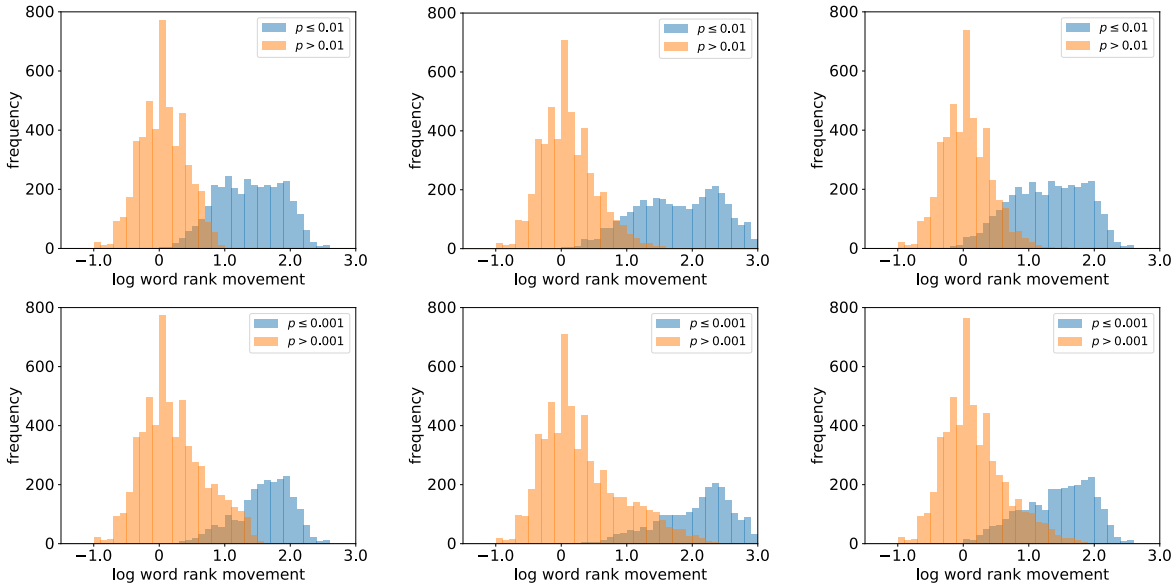


Figure 12. Distributions of log word rank movements of significant keywords (blue) versus those of insignificant keywords (orange), for the ACM-TWEB data set at the $p = 0.05$ (top), $p = 0.01$ (middle), $p = 0.001$ (bottom) levels of confidence. In the left column, we compare the distributions of log word rank movements obtained using the correct corpus, and tested against the correct corpus. In the center column, we compare the distributions of log word rank movements obtained using the wrong corpus (ACM-TSEM), but tested against the correct corpus. In the right column, we compare the distributions of log word rank movements obtained using the correct corpus, but tested against the wrong corpus (ACM-TSEM).

Second, we can use the correct corpus to compute the log work rank movements, but somehow use the wrong corpus to do statistical testing. In Figure 12, we show the distributions of log word rank movements obtained using the wrong corpus (center column), and the distributions of log word rank movements tested using the wrong corpus (right column), compared against the correct distributions of log word rank movements (left column). As we can see, whatever the level of confidence we chose, the overlap between the two distributions increase, whether we use the wrong corpus to compute the log word rank movements, or to perform statistical testing. When we used the wrong corpus to compute the log word rank movements, there was an additional effect: the log word rank movements of significant keywords increased (resulting in a broader distribution), but those of insignificant keywords remained the same.

Discussion

In deriving the log rank movement method for identifying keywords, we explained that a word that is not enriched (i.e. not a keyword) would appear in a short text with probability $P'(n') = A'/n'$, and in the corpus with probability $P(n) = A/n$, where n' and n are the ranks of the word in the short text and corpus respectively. If there are N' tokens in the short text, and N tokens in the corpus containing L short texts, then we expect the word to appear $f' = N'P'(n') = A'N'/n'$ in the short text, and $f = NP(n) = AN/n$ in the corpus. In the simplest TF-IDF method, we can use $f' = A'N'/n'$ as the term frequency TF.

To work out the simplest inverse document frequency $IDF = -\log_{10} l_t/L$, where l_t is the number of short texts in which the given word appears, we can estimate the number of short texts in which the given word *does not appear*. For short text $1 \leq i \leq L$ with N_i' tokens, the probability that the given word does not appear is $[1 - P(n)]^{N_i'}$. This probability varies from text to text if they do not have the same lengths, making it harder to estimate l_t. Therefore, let us use $[1 - P(n)]^{\langle N' \rangle}$ to be the probability that the given word is absent from a short text, where $\langle N' \rangle$ is the average number of tokens in the short texts. With this, we can estimate the number of short texts where the given word is absent to be $L[1 - P(n)]^{\langle N' \rangle}$. Thus, the number of short texts where the given word appears must be $l_t = L\{1 - [1 - P(n)]^{\langle N' \rangle}\}$, and $IDF = -\log_{10} \frac{l_t}{L} = -\log_{10}\{1 - [1 - P(n)]^{\langle N' \rangle}\}$, which is approximately

$$IDF \approx [1 - P(n)]^{\langle N' \rangle} + \frac{1}{2}[1 - P(n)]^{2\langle N' \rangle} + \frac{1}{3}[1 - P(n)]^{3\langle N' \rangle} + \cdots.$$

This is messy, considering how simple $P(n) = A/n$ is. Putting the two contributions together, we find that the TF-IDF score

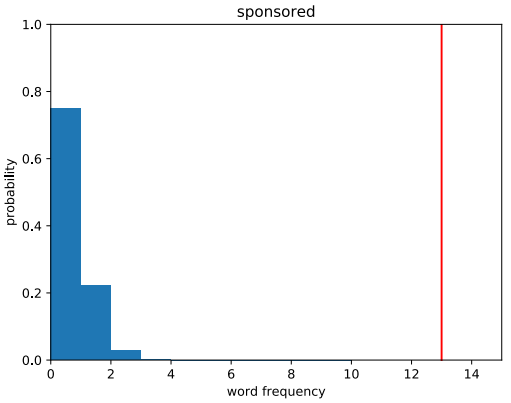
$$TF \cdot IDF = \frac{A'\langle N' \rangle}{n'}\left\{\left[1 - \frac{A}{n}\right]^{\langle N' \rangle} + \frac{1}{2}\left[1 - \frac{A}{n}\right]^{2\langle N' \rangle} + \frac{1}{3}\left[1 - \frac{A}{n}\right]^{3\langle N' \rangle} + \cdots\right\}$$

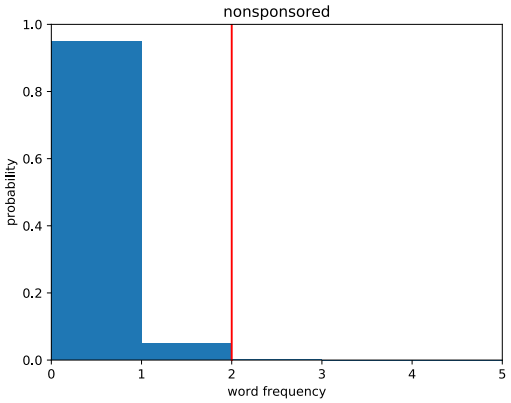
of a word is in general a complicated function of n, its rank in the corpus, and n', its rank in the short text. Instead of $\langle N' \rangle$, we can also write it as N/L.

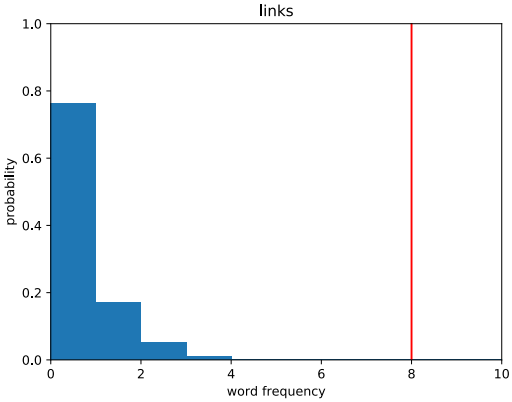
The idea behind the TF-IDF method is very similar to our log rank movement method: look at how frequently a given word or term appears in a given short text, and compare this frequency (or probability) against that expected from how frequently the given word appears in the corpus. This is why the TF-IDF keywords agree so well with our LWRM1 keywords. However, the inventors of the TF-IDF method probably realized that they cannot directly compare $f' = A'N'/n'$ against $f = AN/n$, because the two frequencies are orders of magnitude apart. However, if instead of $IDF = -\log_{10}(l_t/L)$ Salton et al. chose to work with $f/L = AN/Ln = A\langle N' \rangle/n$ the TF-IDF method would be even closer to our LWRM1 method. In any case, in all variants of the TF-IDF method, the score depends on the normalization constants A and A'. Hence, even if n' is the rank we expect from randomly sampling the given word with rank n in the corpus, the TF-IDF score is not zero, or some fixed reference level.

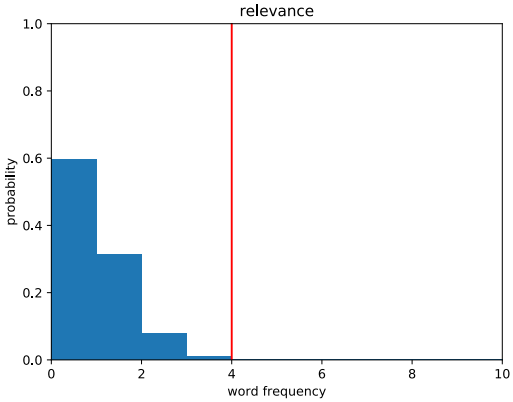
More importantly, for the same level of enrichment (say 10-fold), the TF-IDF score depends on the value of n'. If n' is small, the TF-IDF score would be large. On the other hand, if n' is large, the TF-IDF score would be small. There is therefore a bias in the TF-IDF score against rare words. For a rare word to have the same TF-IDF score as a common word, the level of enrichment of the rare word must be very much larger than that of the common word. In comparison, the LWRM1 score $\Delta_1 = \log_{10} n - \log_{10} n'$ depends only on the level of enrichment n/n', but not on how common or rare the word is in the corpus. This is why we believe the LWRM method is unbiased, and why Δ_1 is so strongly correlated with statistical significance. This is also why the order of TF-IDF keywords is different from that of the LWRM1 keywords.

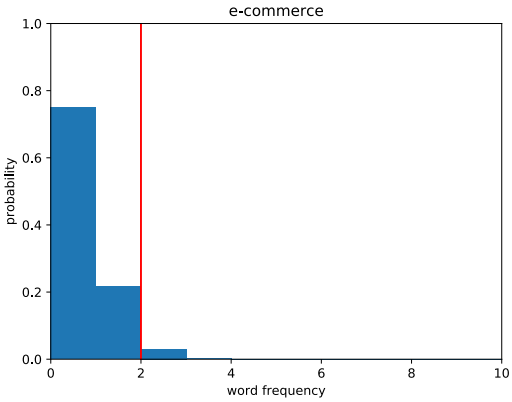
Supporting Information

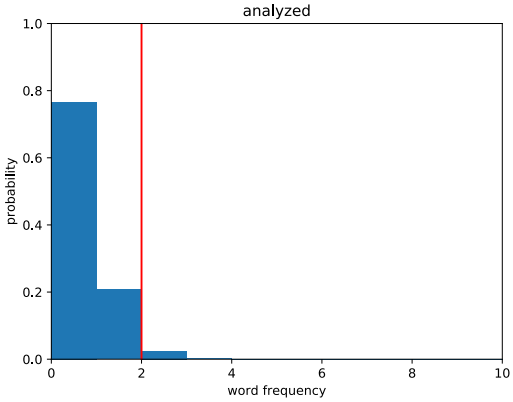

sponsored
probability
word frequency

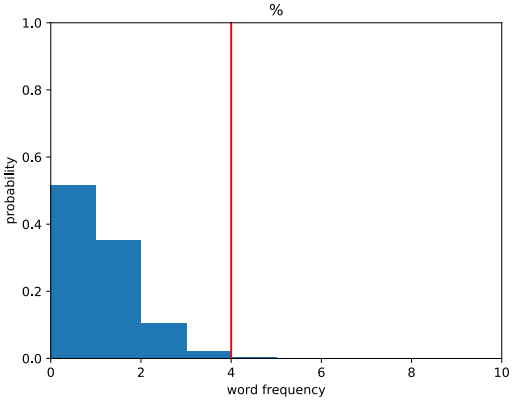

nonsponsored
probability
word frequency

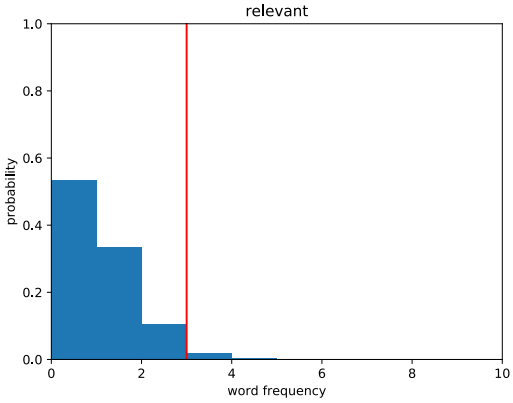

links
probability
word frequency

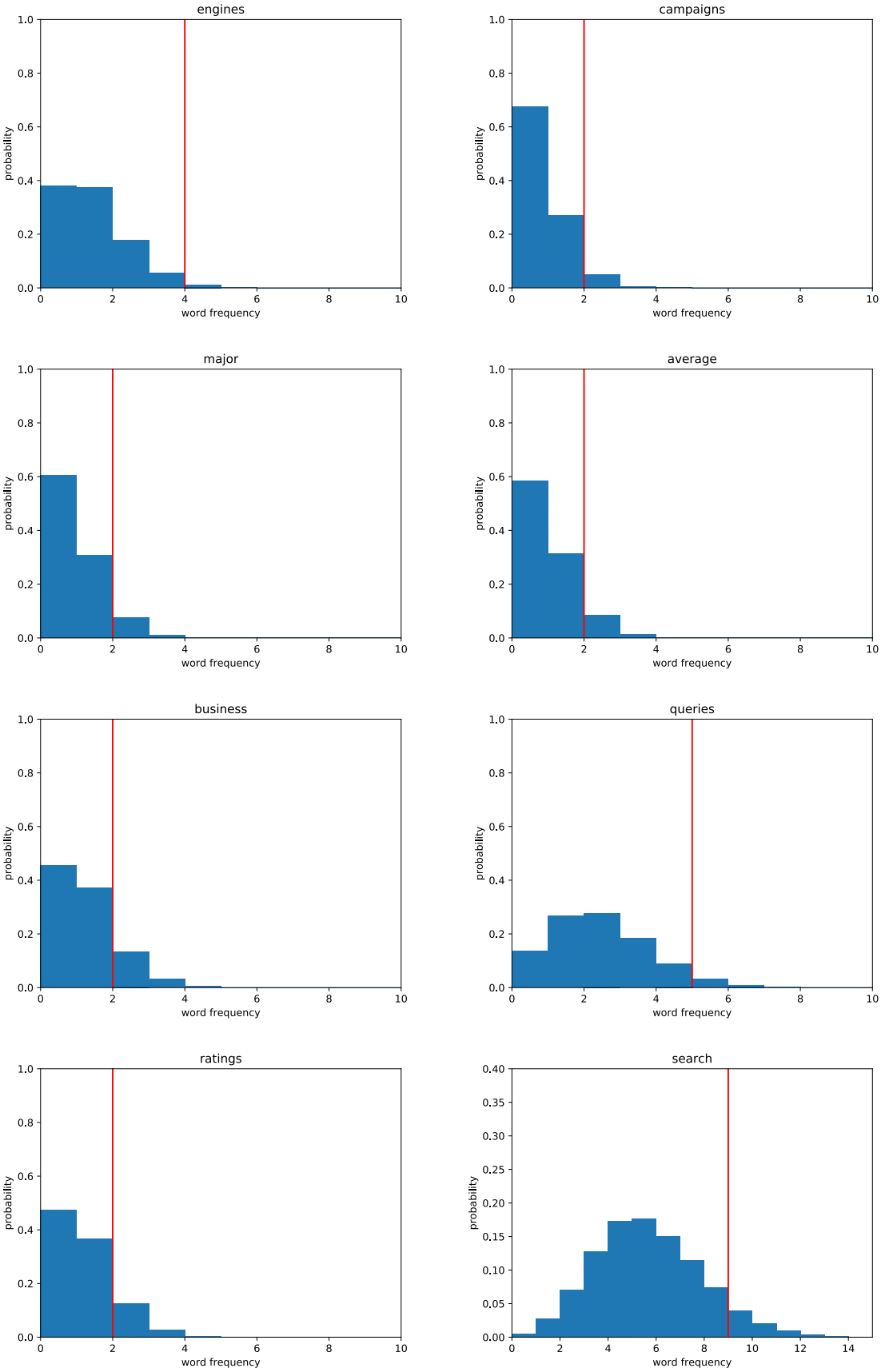

relevance
probability
word frequency


e-commerce
probability
word frequency


analyzed
probability
word frequency


%
probability
word frequency


relevant
probability
word frequency



Supplementary Figure S1. Histograms of null-model frequencies (blue) and observed frequencies (red) for the top 16 LWRM1 keywords. In these histograms, the p-value is

computed using $p = P(N \geq N_{obs})$, which is the probability that the null-model word frequency N is larger or equal to the observed word frequency N_{obs}.

Supplementary Data Files

The raw ACM-TRANS-TWEB and ACM-TRANS-TSEM xml files are private, and must be requested from the Association of Computing Machinery (ACM).

The *Reuters-21578 Text Categorization Collection Data Set* is open, and is available for download on the UCI Machine Learning Repository (http://archive.ics.uci.edu/ml/datasets/Reuters-21578+Text+Categorization+Collection). Researchers who are interested in reproducing our results must first extract the corpora of texts, and save them as `AllNews.npy` so that they can be read by the Python scripts we share.

The open data sets listed below are derived from the raw data sets, and can be downloaded from the Nanyang Technological University DR-NTU (Data) repository using the links provided.

ACM-TWEB	
`OrderedWords.npy`	https://doi.org/10.21979/N9/GZAHAL
`OrderedDicts.npy`	https://doi.org/10.21979/N9/SA1EWV
`AllWords.npy`	https://doi.org/10.21979/N9/QAX2PY
`MasterDict.npy`	https://doi.org/10.21979/N9/YWYGLB
`SortedMasterDict.npy`	https://doi.org/10.21979/N9/N3FURG
`TWEBOrderedBigrams.npy`	https://doi.org/10.21979/N9/4GWA8V
`TWEBOrderedBigramDicts.npy`	https://doi.org/10.21979/N9/BJSHIR
`TWEBAllBigrams.npy`	https://doi.org/10.21979/N9/R6FNVH
`TWEBMasterBigramDict.npy`	https://doi.org/10.21979/N9/B4HSJO
`TWEBSortedMasterBigramDict.npy`	https://doi.org/10.21979/N9/MOZMTP
`TWEBOrderedLogWordRankMovements.npy`	https://doi.org/10.21979/N9/LWLE13
`TWEBOrderedLogBigramRankMovements.npy`	https://doi.org/10.21979/N9/SIN4OV
`shortstopwords.npy`	https://doi.org/10.21979/N9/4MUVJN
`TWEBSSWLWRM.npy`	https://doi.org/10.21979/N9/DADD1A
`TWEB-LWRM1&LWRM2-wordBased.npy`	https://doi.org/10.21979/N9/A2UYYM
`TWEB-LWRM1&RAKE-wordBased.npy`	
`TWEB-LWRM2&RAKE-wordBased.npy`	
`TWEB-LWRM2&TFIDF-wordBased.npy`	
`TWEB-TFIDF&RAKE-wordBased.npy`	
`TWEB-LWRM1&LWRM2-bigramBased.npy`	https://doi.org/10.21979/N9/CVTUIQ
`TWEB-LWRM1&LWRM2-bigramValue.npy`	
`TWEB-LWRM1&RAKE-bigramBased.npy`	
`TWEB-LWRM1&RAKE-bigramValue.npy`	
`TWEB-LWRM2&RAKE-bigramBased.npy`	
`TWEB-LWRM2&RAKE-bigramValue.npy`	
`TWEB-LWRM2&TFIDF-bigramBased.npy`	
`TWEB-LWRM2&TFIDF-bigramValue.npy`	
`TWEB-TFIDF&RAKE-bigramBased.npy`	
`TWEB-TFIDF&RAKE-bigramValue.npy`	
`wrongCorpus.npy`	https://doi.org/10.21979/N9/OHMS4W
Reuters	
`ReutersOrderedWords.npy`	https://doi.org/10.21979/N9/L09RSR
`ReutersOrderedDicts.npy`	

ReutersAllWords.npy	
ReutersMasterDict.npy	
ReutersSortedMasterDict.npy	
ReutersOrderedBigrams.npy	https://doi.org/10.21979/N9/RT6QFP
ReutersOrderedBigramDicts.npy	
ReutersAllBigrams.npy	
ReutersMasterBigramDict.npy	
ReutersSortedMasterBigramDict.npy	
ReutersOrderedLogWordRankMovements.npy	https://doi.org/10.21979/N9/S6XPII
ReutersOrderedLogBigramRankMovements.npy	
News-LWRM1&LWRM2-wordBased.npy	https://doi.org/10.21979/N9/1YOH1P
News-LWRM1&RAKE-wordBased.npy	
News-LWRM1&TFIDF-wordBased.npy	
News-LWRM2&RAKE-wordBased.npy	
News-LWRM2&TFIDF-wordBased.npy	
News-RAKE&TFIDF-wordBased.npy	
News-LWRM1&LWRM2-bigramBased.npy	https://doi.org/10.21979/N9/YMVWYW
News-LWRM1&LWRM2-bigramValue.npy	
News-LWRM1&RAKE-bigramBased.npy	
News-LWRM1&RAKE-bigramValue.npy	
News-LWRM2&RAKE-bigramBased.npy	
News-LWRM2&RAKE-bigramValue.npy	
News-LWRM2&TFIDF-bigramBased.npy	
News-LWRM2&TFIDF-bigramValue.npy	
News-RAKE&TFIDF-bigramBased.npy	
News-RAKE&TFIDF-bigramValue.npy	
ReutersFKW00.npy	https://doi.org/10.21979/N9/EC76OJ
ReutersFKW01.npy	
ReutersFKW02.npy	
ReutersFKW03.npy	
ReutersFKW04.npy	
ReutersFKW05.npy	
ReutersFKW06.npy	
ReutersFKW07.npy	
ReutersFKW08.npy	
ReutersFKW09.npy	
ReutersFKW10.npy	
ReutersFKW11.npy	
ReutersFKW12.npy	
ReutersFKW13.npy	
ReutersFKW14.npy	
ReutersFKW15.npy	
ReutersFKW16.npy	
ReutersFKW17.npy	
ReutersFKW18.npy	
ReutersFKW19.npy	

Supplementary Program Scripts

ACM-TWEB	
WordRankPlot.py	https://doi.org/10.21979/N9/CEGKQB
BigramRankPlot.py	https://doi.org/10.21979/N9/H3PH6V
FindLogBigramRankMovements.py	https://doi.org/10.21979/N9/VF6R3Z
FindLogWordRankMovements.py	https://doi.org/10.21979/N9/DWSDMH
wordBasedGnerator.py	https://doi.org/10.21979/N9/VDX7VD
bigramBasedGenerator.py	https://doi.org/10.21979/N9/R6JIRI

compareLWRM1LWRM2word.py	https://doi.org/10.21979/N9/LW0MZY
compareLWRM1TFIDF.py	https://doi.org/10.21979/N9/7ZE51Z
compareLWRM1RAKE.py	https://doi.org/10.21979/N9/CJTOM7
compareRAKELWRM2word.py	https://doi.org/10.21979/N9/3O7YHG
compareTFIDFLWRM2word.py	https://doi.org/10.21979/N9/RRHC17
compareTFIDFRAKE.py	https://doi.org/10.21979/N9/P7SHEG
compareLWRM1LWRM2bigram.py	https://doi.org/10.21979/N9/6YAJTB
compareLWRM1RAKEbigram.py	https://doi.org/10.21979/N9/4NEP8M
compareRAKELWRM2bigram.py	https://doi.org/10.21979/N9/PDMFKV
compareTFIDFLWRM2bigram.py	https://doi.org/10.21979/N9/UBVH8Z
compareTFIDFRAKEbigram.py	https://doi.org/10.21979/N9/R3QTZ5
FindWrongLogWordRankMovements.py	https://doi.org/10.21979/N9/JVJ7IS
TWEBstattestword.py	https://doi.org/10.21979/N9/GS3XPB
TWEBstattestbigram.py	https://doi.org/10.21979/N9/SSCUFT
TWEBstattestwrongmodel.py	https://doi.org/10.21979/N9/36ICLG
TWEBstattestwrongranks.py	https://doi.org/10.21979/N9/ORCIWL
Reuters	
ReutersWordCount.py	https://doi.org/10.21979/N9/CK5QLF
ReutersBigramCount.py	https://doi.org/10.21979/N9/JNDJ79
FindReutersLogBigramRankMovements.py	https://doi.org/10.21979/N9/NJSD93
FindReutersLogWordRankMovements.py	https://doi.org/10.21979/N9/OFMDKY
compareReutersLWRM2RAKEword.py	https://doi.org/10.21979/N9/FLSGV9
compareReutersLWRM1LWRM2word.py	https://doi.org/10.21979/N9/4GRKTC
compareReutersLWRM2TFIDFword.py	https://doi.org/10.21979/N9/ZUPA02
compareReutersLWRM1RAKEword.py	https://doi.org/10.21979/N9/XN6L6I
compareReutersLWRM1TFIDFword.py	https://doi.org/10.21979/N9/HQ5S7D
compareReutersTFIDFRAKEword.py	https://doi.org/10.21979/N9/L3XKZ2
compareReutersLWRM1LWRM2bigram.py	https://doi.org/10.21979/N9/3LCIMM
compareReutersLWRM2TFIDFbigram.py	https://doi.org/10.21979/N9/CEBYY8
compareReutersLWRM1RAKEbigram.py	https://doi.org/10.21979/N9/H9XIVZ
compareReutersTFIDFRAKEbigram.py	https://doi.org/10.21979/N9/Z5CEC5
compareReutersLWRM2RAKEbigram.py	https://doi.org/10.21979/N9/ULETZ9
RNstattestword.py	https://doi.org/10.21979/N9/A0XM3W
ReutersStatTest.py	

References

1. Bornmann L, Mutz R. Growth rates of modern science: A bibliometric analysis based on the number of publications and cited references. Journal of the Association for Information Science and Technology. 2015; 66(11):2215-2222. https://doi.org/10.1002/asi.23329
2. Pautasso M. Publication growth in biological sub-fields: patterns, predictability and sustainability. Sustainability. 2012; 4(12):3234-3247. https://doi.org/10.3390/su4123234
3. Sotudeh H, Salesi M, Didegah F, Bazgir B. Does scientific productivity influence athletic performance? An analysis of countries' performances in sciences, sport sciences and Olympic Games. International Journal of Information Science and Management. 2012; 10(2):27-41.
4. Lyubetsky V, Piel WH, Quandt D. Current advances in molecular phylogenetics. BioMed Research International. 2014; 2014:596746. https://dx.doi.org/10.1155/2014/596746

5. Dhawan SM, Gupta BM, Gupta R. Social Science research landscape in South Asia: a comparative assessment of research output published during 1996-2013. Library Philosophy and Practice. 2015; 1251.

6. Powell JJ, Fernandez F, Crist JT, Dusdal J, Zhang L, Baker DP. Introduction: The Worldwide Triumph of the Research University and Globalizing Science. In: *The Century of Science (International Perspectives on Education and Society, Volume 33)* edited by Powell JJ, Baker DP, Fernandez F. Emerald Publishing Inc; 2017. pp. 1-36. https://doi.org/10.1108/S1479-367920170000033003

7. Pan RK, Petersen AM, Pammolli F, Fortunato S. The memory of science: Inflation, myopia, and the knowledge network. Journal of Informetrics. 2018; 12(3):656-678. https://doi.org/10.1016/j.joi.2018.06.005

8. Borgatti SP, Foster PC. The network paradigm in organizational research: A review and typology. Journal of Management. 2003; 29(6):991-1013. https://doi.org/10.1016/S0149-2063(03)00087-4

9. Iordanskii AL, Rogovina SZ, Berlin AA. Current state and developmental prospects for nanopatterned implants containing drugs. Review Journal of Chemistry. 2013; 3(2):117-132. https://doi.org/10.1134/S2079978013020027

10. Liew SL, Santarnecchi E, Buch ER, Cohen LG. Non-invasive brain stimulation in neurorehabilitation: local and distant effects for motor recovery. Frontiers in Human Neuroscience. 2014; 8:378. https://doi.org/10.3389/fnhum.2014.00378

11. Wang J, Choi HS, Wáng YXJ. Exponential growth of publications on carbon nanodots by Chinese authors. Journal of Thoracic Disease. 2015; 7(7):E201-E205. https://doi.org/10.3978/j.issn.2072-1439.2015.06.13

12. Li J, Wang Q, Oremland RS, Kulp TR, Rensing C, Wang G. Microbial antimony biogeochemistry: enzymes, regulation, and related metabolic pathways. Applied and Environmental Microbiology. 2016; 82(18):5482-5495. https://doi.org/10.1128/AEM.01375-16

13. Haunschild R, Bornmann L, Marx W. Climate change research in view of bibliometrics. PLoS One. 2016; 11(7):e0160393. https://dx.doi.org/10.1371/journal.pone.0160393

14. Nardi P, Di Matteo G, Palahi M, Mugnozza GS. Structure and evolution of mediterranean forest research: a science mapping approach. PloS One. 2016; 11(5): e0155016. https://dx.doi.org/10.1371/journal.pone.0155016

15. Schofield DJ, Zeppel MJ, Tan O, Lymer S, Cunich MM, Shrestha RN. A brief, global history of microsimulation models in health: past applications, lessons learned and future directions. International Journal of Microsimulation. 2018; 11(1):97-142.

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

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

18. Gupta TE. Keyword extraction: a review. International Journal of Engineering Applied Sciences and Technology. 2017; 2(4):215-220.

19. Lloret E, Palomar M. Text summarisation in progress: a literature review. Artificial Intelligence Review. 2012; 37(1):1-41. https://doi.org/10.1007/s10462-011-9216-z

20. Haque M, Pervin S, Begum Z. Literature review of automatic multiple documents text summarization. International Journal of Innovation and Applied Studies. 2013; 3(1):121-129.

21. Mishra R, Bian J, Fiszman M, Weir CR, Jonnalagadda S, Mostafa J, Del Fiol G. Text summarization in the biomedical domain: a systematic review of recent research. Journal of Biomedical Informatics. 2014; 52:457-467. https://doi.org/10.1016/j.jbi.2014.06.009

22. Gaikwad DK, Mahender CN. A review paper on text summarization. International Journal of Advanced Research in Computer and Communication Engineering. 2016; 5(3):154-160.

23. Allahyari M, Pouriyeh S, Assefi M, Safaei S, Trippe ED, Gutierrez JB, Kochut K. Text summarization techniques: A brief survey. International Journal of Advanced Computer Science and Applications. 2017; 8(10):397-405. https://dx.doi.org/10.14569/IJACSA.2017.081052

24. Wallach HM. Topic modeling: beyond bag-of-words. In: *Proceedings of the 23rd International Conference on Machine Learning (Pittsburgh, PA; Jun 25-29, 2006)* edited by Cohen W, Moore A. pp. 977-984. ACM, 2006. https://doi.org/10.1145/1143844.1143967

25. Wang C, Blei DM. Collaborative topic modeling for recommending scientific articles. In: *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (San Diego, CA; Aug 21-24, 2011)*. pp. 448-456. ACM, 2011. https://doi.org/10.1145/2020408.2020480

26. Meeks E, Weingart SB. The digital humanities contribution to topic modeling. Journal of Digital Humanities. 2012; 2(1):1-6.

27. Yau CK, Porter A, Newman N, Suominen A. Clustering scientific documents with topic modeling. Scientometrics. 2014; 100(3):767-786. https://doi.org/10.1007/s11192-014-1321-8

28. Liu L, Tang L, Dong W, Yao S, Zhou W. An overview of topic modeling and its current applications in bioinformatics. SpringerPlus. 2016; 5(1):1608. https://doi.org/10.1186/s40064-016-3252-8

29. Berry MW, Kogan J. Text Mining: Applications and Theory. Chichester, UK: John Wiley & Sons; 2010.

30. Feather J, Sturges P. International Encyclopedia of Information and Library Science, Second Edition. London, UK: Routledge; 2003.

31. Manning CD, Schütze H. Foundations of Statistical Natural Language Processing. Cambridge, MA: MIT Press; 1999.

32. Salton G, Yang CS, Yu CT. A theory of term importance in automatic text analysis. Journal of the American Society for Information Science. 1975; 26(1):33-44. https://doi.org/10.1002/asi.4630260106

33. Cohen JD. Highlights: language and domain-independent automatic indexing terms for abstracting. Journal of the American Society for Information Science. 1995; 46(3):162-174. https://doi.org/10.1002/(SICI)1097-4571(199504)46:3<162::AID-ASI2>3.0.CO;2-6

34. Turney PD. Coherent keyphrase extraction via web mining. In: *Proceedings of the 18th International Joint Conference on Artificial Intelligence (Acapulco, Mexico; Aug 9-15, 2003).* pp. 434-439. Morgan Kaufmann Publishers Inc., 2003.

35. Andrade MA, Valencia A. Automatic extraction of keywords from scientific text: application to the knowledge domain of protein families. Bioinformatics. 1998; 14(7):600-607. https://doi.org/10.1093/bioinformatics/14.7.600

36. Jones KS. A statistical interpretation of term specificity and its application in retrieval. Journal of Documentation. 2004; 60(5):493-502. https://doi.org/10.1108/00220410410560573

37. Ortuño M, Carpena P, Bernaola-Galván P, Muñoz E, Somoza AM. Keyword detection in natural languages and DNA. Europhysics Letters. 2002; 57(5):759-764. https://doi.org/10.1209/epl/i2002-00528-3

38. Carpena P, Bernaola-Galván P, Hackenberg M, Coronado AV, Oliver JL. Level statistics of words: Finding keywords in literary texts and symbolic sequences. Physical Review E. 2009; 79(3): 035102(R). https://doi.org/10.1103/PhysRevE.79.035102

39. Herrera JP, Pury PA. Statistical keyword detection in literary corpora. The European Physical Journal B. 2008; 63:135-146. https://doi.org/10.1140/epjb/e2008-00206-x

40. Matsuo Y, Ishizuka M. Keyword extraction from a single document using word co-occurrence statistical information. International Journal on Artificial Intelligence Tools. 2004; 13(01):157-169. https://doi.org/10.1142/S0218213004001466

41. Mehri A, Darooneh AH. Keyword extraction by non-extensivity measure. Physical Review E. 2011; 83(5):056106. https://doi.org/10.1103/PhysRevE.83.056106

42. Salton G, Wong A, Yang CS. A vector space model for automatic indexing. Communications of the ACM. 1975; 18(11):613-620. https://doi.org/10.1145/361219.361220

43. Mihalcea R, Tarau P. TextRank: bringing order into texts. In: Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing (Barcelona, Spain; Jul 25-26, 2004). pp. 404-411. Association for Computational Linguistics, 2004. Accessed at: https://www.aclweb.org/anthology/W04-3252.pdf.

44. Palshikar GK. Keyword extraction from a single document using centrality measures. In: *Pattern Recognition and Machine Intelligence (Lecture Notes in Computer Science volume 4851) edited by Ghosh A, De RK, Pal SK.* pp.503-510. Springer-Verlag, 2007. https://doi.org/10.1007/978-3-540-77046-6_62

45. Wang J, Liu J, Wang C. Keyword extraction based on PageRank. In: Advances in Knowledge Discovery and Data Mining (Lecture Notes in Computer Science, volume 4426) edited by Zhou Z-H, Li H, Yang Q. pp. 857-864. Springer, 2007. https://doi.org/10.1007/978-3-540-71701-0_95

46. Litvak M, Last M. Graph-based keyword extraction for single-document summarization. In: *Proceedings of the Workshop on Multi-source Multilingual Information Extraction and Summarization*. pp.17-24. ACM, 2008.

47. Boudin F. A comparison of centrality measures for graph-based keyphrase extraction. In: *Proceedings of the 6th International Joint Conference on Natural Language Processing (Nagoya, Japan; Oct 14-19, 2013)*. pp. 834-838. Association for Computational Linguistics, 2013. https://www.aclweb.org/anthology/I13-1102.pdf.

48. Lahiri S, Choudhury SR, Caragea C. Keyword and keyphrase extraction using centrality measures on collocation networks. arXiv preprint arXiv:1401.6571, 2014.

49. Litvak M, Last M, Aizenman H, Gobits I, Kandel A. DegExt — a language-independent graph-based keyphrase extractor. In: Advances in Intelligent and Soft Computing, volume 86 edited by Mugellini E, Szczepaniak PS, Pettenati MC, Sokhn M. pp 121-130. Springer, 2011. https://doi.org/10.1007/978-3-642-18029-3_13

50. Abilhoa WD, de Castro LN. A keyword extraction method from twitter messages represented as graphs. Applied Mathematics and Computation. 2014; 240:308-325. https://doi.org/10.1016/j.amc.2014.04.090

51. Rose S, Engel D, Cramer N, Cowley W. Automatic keyword extraction from individual documents. In: *Text Mining: Applications and Theory* edited by Berry MW, Kogan J. pp. 3-20. John Wiley & Sons Ltd, 2010. https://doi.org/10.1002/9780470689646.ch1

52. Wan X, Xiao J. Single document keyphrase extraction using neighborhood knowledge. In: *Proceedings of the 23rd AAAI Conference on Artificial Intelligence (Chicago, IL; Jul 13-17, 2008)*. pp. 855-860. ACM, 2008.

53. Grineva M, Grinev M, Lizorkin D. Extracting key terms from noisy and multi-theme documents. In: *Proceedings of the 18th International Conference on World Wide Web (Madrid, Spain; Apr 20-24, 2009)*. pp. 661-670. ACM, 2009. https://doi.org/10.1145/1526709.1526798

54. Sonawane SS, Kulkarni PA. Graph based representation and analysis of text document: a survey of techniques. International Journal of Computer Applications. 2014; 96(19):1-8. https://doi.org/10.5120/16899-6972

55. Duari S, Bhatnagar V. sCAKE: semantic connectivity aware keyword extraction. Information Sciences. 2019; 477:100-117. https://doi.org/10.1016/j.ins.2018.10.034

56. Vega-Oliveros DA, Gomes PS, Milios EE, Berton L. A multi-centrality index for graph-based keyword extraction. Information Processing & Management. 2019; 56(6):102063. https://doi.org/10.1016/j.ipm.2019.102063

57. Lim JB, Lee JH, Gil J-M. A keyword extraction scheme from CQI based on graph centrality. In *Advanced Multimedia and Ubiquitous Engineering* edited by Park JJ, Yang LT, Jeong Y-S, Hao F. pp. 158-163. Springer, 2019. https://doi.org/10.1007/978-981-32-9244-4_22

58. Chen Y, Wang J, Li P, Guo P. Single document keyword extraction via quantifying higher-order structural features of word co-occurrence graph. Computer Speech & Language. 2019; 57:98-107. https://doi.org/10.1016/j.csl.2019.01.007

59. Anjali S, Nair M, Thushara MG. A graph based approach for keyword extraction from documents. In: *Proceedings of the 2019 Second International Conference on Advanced Computational and Communication Paradigms (Gangtok, India; 25-28 Feb 2019)*. pp. 1-4. IEEE, 2019. https://doi.org/10.1109/ICACCP.2019.8882946

60. Syafiandini AF, Mustika HF, Manik LP, Rianto Y, Akbar Z. Implementing graph based rank on online news media keyword extraction. In: *Proceedings of the 2019 International Conference on Computer, Control, Informatics and its Applications (Serpong, Indonesia; 23-24 Oct 2019)*. pp. 108-113. IEEE, 2019. https://doi.org/10.1109/IC3INA48034.2019.8949575

61. Thushara MG, Anjali S, Nair M. A graph-based model for keyword extraction and tagging of research documents. In: *Proceedings of the 2019 2nd International Conference on Intelligent Computing, Instrumentation and Control Technologies (Kannur, India; 5-6 Jul 2019)*. vol. 1, pp. 942-946. IEEE, 2019. https://doi.org/10.1109/ICICICT46008.2019.8993142

62. Chatterjee PC, Bordoloi M, Biswas SK. Keyword extraction using graph based supervised term weighting. In: *Proceedings of the 2019 2nd International Conference on Innovations in Electronics, Signal Processing and Communication (Shilong, India; 1-2 Mar 2019)*. pp. 142-147. IEEE, 2019. https://doi.org/10.1109/IESPC.2019.8902431

63. Škrlj B, Repar A, Pollak S. RaKUn: rank-based keyword extraction via unsupervised learning and meta vertex aggregation. In: *Statistical Language and Speech Processing*. pp. 311-323. Springer, 2019. https://doi.org/10.1007/978-3-030-31372-2_26

64. Xiong A, Guo Q. Chinese news keyword extraction algorithm based on TextRank and topic model. In: *Artificial Intelligence for Communications and Networks*. pp. 334-341. Springer, 2019. https://doi.org/10.1007/978-3-030-22968-9_29

65. Wang H, Ye J, Yu Z, Wang J, Mao C. Unsupervised keyword extraction methods based on a word graph network. *International Journal of Ambient Computing and Intelligence*. 2020; 11(2):68-79. https://doi.org/10.4018/IJACI.2020040104

66. Steier A, Belew R. Exporting phrases: A statistical analysis of topical language. In: Proceedings of the Second Annual Symposium on Document Analysis and Information Retrieval (Las Vegas, Nevada, USA; 26-28 Apr 1993). pp. 179-190. University of Nevada, 1993.

67. Krulwich B, Burkey C. Learning user information interests through the extraction of semantically significant phrases. In: *Proceedings of the AAAI 1996 Spring Symposium on Machine Learning in Information Access (Stanford, California, USA; 25-27 Mar 1996)*. pp. 110-112. AAAI Press, 1996.

68. Muñoz A. Compound key word generation from document databases using a hierarchical clustering ART model. Intelligent Data Analysis. 1996; 1(1):25-48. https://doi.org/10.3233/IDA-1997-1103

69. Barker K, Cornacchia N. Using nounphrase heads to extract document keyphrases. In: *Advances in Artificial Intelligence, Lecture Notes in Computer Science, volume 1822/2000*. pp 40-52. Springer, 2000. https://doi.org/10.1007/3-540-45486-1_4

70. Tomikoyo T, Hurst M. A language model approach to keyphrase extraction. In: *Proceedings of the ACL Workshop on Multiword Expressions: Analysis, Acquisition and Treatment*. vol 18, pp. 33-40. ACL, 2003. https://doi.org/10.3115/1119282.1119287

71. Bracewell DB, Ren F, Kuriowa S. Multilingual single document keyword extraction for information retrieval. In: Proceedings of the 2005 International Conference on Natural Language Processing and Knowledge Engineering (Wuhan, China; 30 Oct–1 Nov 2005). pp. 517-522. IEEE, 2005. https://doi.org/10.1109/NLPKE.2005.1598792

72. Liu F, Pennell D, Liu F, Liu Y. Unsupervised approaches for automatic keyword extraction using meeting transcripts. In: *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics (Boulder, Colorado, USA; 31 May–5 Jun 2009)*. pp. 620-628. ACL, 2009.

73. Liu Z, Li P, Zheng Y, Sun M. Clustering to find exemplar terms for keyphrase extraction. In: *Proceedings of Conference on Empirical Methods in Natural Language Processing (Singapore; 6–7 May 2009)*. pp. 257-266. ACL, 2009.

74. Gazendam L, Wartena C, Brussee R. Thesaurus based term ranking for keyword extraction. In: *Proceedings for Workshops on Database and Expert Systems Applications (Bilbao, Spain; 30 Aug–3 Sep 2010)*. pp. 49-53. IEEE, 2010. https://doi.org/10.1109/DEXA.2010.31

75. Litvak M, Last M, Aizenman H, Gobits I, Kandel A. DegExt — a language-independent graph-based keyphrase extractor. In: *Advances in Intelligent and Soft Computing, volume 86*. pp 121-130. Springer, 2011. https://doi.org/10.1007/978-3-642-18029-3_13

76. Bao H, Deng Z. An extended keyword extraction method. In: *Proceedings of the 2012 International Conference on Applied Physics and Industrial Engineering, edited by Yang D*. pp. 1120-1127, Elsevier, 2012. https://doi.org/10.1016/j.phpro.2012.02.167

77. Turney PD. Learning algorithms for keyphrase extraction. Information Retrieval. 2000; 2:303-336. https://doi.org/10.1023/A:1009976227802

78. Frank E, Paynter GW, Witten IH, Gutwin C, Nevill-Manning CG. Domain-specific keyphrase extraction. In: *Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence (Stockholm, Sweden; 31 Jul–6 Aug 1999), volume 2*. pp. 668-673. IJCAI, 1999.

79. Medelyan O, Witten H. Thesaurus based automatic keyphrase indexing. In: *Proceedings of the 6th ACM/IEEE-CS Joint Conference on Digital Libraries (Chapel Hill, North Carolina, USA; 11–15 Jun 2006)*. pp. 296-297. ACM, 2006. https://doi.org/10.1145/1141753.1141819

80. Song M, Song I-Y, Hu X. KPSpotter: a flexible information gain-based keyphrase extraction system. In: *Proceedings of the 5th ACM International Workshop on Web Information and Data Management (New Orleans, Louisiana, USA; 2–8 Nov 2003)*. pp. 50–53, ACM, 2003. https://doi.org/10.1145/956699.956710

81. Hulth A. Improved automatic keyword extraction given more linguistic knowledge. In: *Proceedings of the 2003 Conference on Empirical Methods in Natural Language Processing (Sapporo, Japan; 11–12 Jul 2003)*. pp. 216-223. ACL, 2003. https://doi.org/10.3115/1119355.1119383

82. Tang J, Li J.-Z, Wang K-H, Cai Y-R. Loss minimization based keyword distillation. In: *Advanced Web Technologies and Applications (Lecture Notes in Computer Science, Volume 3007)*. pp. 572-577. Springer, 2004. https://doi.org/10.1007/978-3-540-24655-8_62

83. Zhang C, Wang H, Liu Y, Wu D, Liao Y, Wang B. Automatic keyword extraction from documents using conditional random fields. Journal of Computational Information Systems. 2008; 4(3):1169-1180.

84. Ercan G, Cicekli I. Using lexical chains for keyword extraction. Information Processing and Management. 2007; 43(6):1705-1714. https://doi.org/10.1016/j.ipm.2007.01.015

85. Feng J, Xie F, Hu X, Li P, Cao J, Wu X. Keyword extraction based on sequential pattern mining. In: *Proceedings of the Third International Conference on Internet Multimedia Computing and Service (Chengdu, China; 5–7 Aug 2011)*. pp. 34-38. ACM, 2011. https://doi.org/10.1145/2043674.2043685

86. Armouty B, Tedmori S. Automated keyword extraction using support vector machine from Arabic news documents. In: *2019 IEEE Jordan International Joint Conference on Electrical Engineering and Information Technology (Amman, Jordan; 9-11 Apr 2019)*, pp. 342-346. IEEE, 2019. https://doi.org/10.1109/JEEIT.2019.8717420

87. Sharifi A, Mahdavi MA. Supervised approach for keyword extraction from Persian documents using lexical chains. Signal and Data Processing. 2019; 15(4):95-110.

88. Ogul IU, Ozcan C, Hakdagli O. Keyword extraction based on word synonyms using WORD2VEC. In: *27th Signal Processing and Communications Applications Conference (Sivas, Turkey; 24-26 Apr 2019)*. pp. 1-4. IEEE, 2019. https://doi.org/10.1109/SIU.2019.8806496

89. Duari S, Bhatnagar V. sCAKE: semantic connectivity aware keyword extraction. Information Sciences. 2019; 477:100-117. https://doi.org/10.1016/j.ins.2018.10.034

90. Rohith P, Kumar SS, Anju RC. Keyword extraction from Malayalam news articles using conditional random fields. In: *Second International Conference on Advanced Computational and Communication Paradigms (Gangtok, India; 25-28 Feb 2019)*. pp. 1-4. IEEE, 2019. https://doi.org/10.1109/ICACCP.2019.8882999

91. Zipf GK. *The Psychobiology of Language*. Routledge, 1935.
92. Piantadosi ST. Zipf's word frequency law in natural language: a critical review and future directions. Psychonomic Bulletin & Review. 2015; 21(5):1112-1130. https://dx.doi.org/10.3758%2Fs13423-014-0585-6
93. Prün C. Validity of Menzerath-Altmann's law: graphic representation of language, information processing systems and synergetic linguistics. Journal of Quantitative Linguistics. 1994; 1(2):148-155. https://doi.org/10.1080/09296179408590009
94. Cramer I. The parameters of the Altmann-Menzerath law. Journal of Quantitative Linguistics. 2005; 12(1):41-52. https://doi.org/10.1080/09296170500055301
95. Kulacka A, Macutek J. A discrete formula for the Menzerath-Altmann law. Journal of Quantitative Linguistics. 2007; 14(1):23-32. https://doi.org/10.1080/09296170600850585
96. Kulacka A. The coefficients in the formula for the Menzerath-Altmann law. Journal of Quantitative Linguistics. 2010; 17(4):257-268. https://doi.org/10.1080/09296174.2010.512160
97. Chen Q, Guo J, Liu Y. A statistical study on Chinese word and character usage in literatures from the Tang Dynasty to the present. Journal of Quantitative Linguistics. 2012; 19(3):232-248. https://doi.org/10.1080/09296174.2012.685305
98. Eroglu S. Menzerath-Altmann law for distinct word distribution analysis in a large text. Physica A: Statistical Mechanics and its Applications. 2013; 392(12):2775-2780. https://doi.org/10.1016/j.physa.2013.02.012
99. Eroglu S. Menzerath-Altmann law: statistical mechanical interpretation as applied to linguistic organization. Journal of Statistical Physics. 2014; 157:392-405. https://doi.org/10.1007/s10955-014-1078-8
100. Chierichetti F, Kumar R, Pang B. On the power laws of language: word frequency distributions. In: Proceedings of SIGIR (7-11 Aug 2017, Tokyo, Japan).