Journal of Naval Sciences and Engineering
2021, Vol. 17, No. 2, pp. 217-239
Computer Engineering/Bilgisayar Mühendisliği

RESEARCH ARTICLE

**An ethical committee approval and/or legal/special permission has not been required within the scope of this study.*

ON THE EFFECT OF WORD POSITIONS IN GRAPH-BASED KEYWORD EXTRACTION*

Osman KABASAKAL[1]
Alev MUTLU [2]

[1]*Kocaeli University, Department of Computer Engineering, Kocaeli, Turkey,*
185112060@kocaeli.edu.tr; ORCID: 0000-0003-1187-5147

[2]*Kocaeli University, Department of Computer Engineering, Kocaeli, Turkey,*
alev.mutlu@kocaeli.edu.tr; ORCID: 0000-0003-0547-0653

Received: 28.11.2020 **Accepted: 25.06.2021**

ABSTRACT

In this study, we focus on the effect of word positions in unsupervised, graph-based keyword extraction. To this aim, we discuss the performance of four node-weighting procedures, namely Word Position (WP), Word Position Bidirectional (WPB), Sentence Position (SP), and Sentence Position Bidirectional (SPB). WP assigns higher weights to words that appear at the beginning of a text. WPB assigns higher weights to words that appear either at the beginning or end of a text. SP assigns higher weights to words that appear in the very first sentences of a text. SPB assigns higher weights to words that appear in sentences that are either close to the beginning or end of a text. Experiments conducted on six benchmark datasets show that WP and SP do not statistically differ. However, for datasets whose keywords appear early in the text WP performs better than SP with no statistical difference, while for datasets where keywords are evenly distributed in text SP statistically performs better than WP.

Keywords: *Keyword Extraction, Sentence Position, Word Position.*

ÇİZGE TABANLI ANAHTAR KELİME ÇIKARIMINDA KELİME POZİSYONLARININ ETKİSİ

ÖZ

Bu çalışmada gözetimsiz, çizge tabanlı anahtar kelime çıkarma yöntemlerinde kelime pozisyonlarının etkisine odaklanılmaktadır. Bu amaçla, düğümler için; Kelime Pozisyonu (WP), Kelime Pozisyonu Çift Yönlü (WPB), Cümle Pozisyonu (SP) ve Cümle Pozisyonu Çift Yönlü (SPB) isimli ilk ağırlıklandırma yöntemleri üzerinde durulmakta ve bunların performans üzerindeki etkileri tartışılmaktadır. WP, bir metnin başında yer alan kelimelere daha fazla ağırlık vermektedir. WPB, bir metnin başında ya da sonunda bulunan kelimelere daha fazla ağırlık vermektedir. SP, metnin ilk cümlelerinde geçen kelimelere daha fazla ağırlık vermektedir. SPB ise metnin başında ve sonunda yer alan cümlelerdeki kelimelere daha fazla ağırlık vermektedir. Altı veri kümesi üzerinde yapılan deneylerde, WP ve SP ağırlıklandırmalarına istatistiksel bir fark gözelemlenmemiştir. Ancak anahtar kelimelerin metnin başında geçen veri kümelerinde WP daha yüksek başarım göstermekle birlikte SP'den istatistiksel olarak ayrılmamaktadır. Anahtar kelimelerin metin içinde dağıtılmış olan veri kümelerinde SP, WP'den daha başarılı olmakta ve istatistiksel fark göstermektedir.

Anahtar Kelimeler: *Anahtar Kelime Çıkarımı, Cümle Konumu, Kelime Konumu.*

1. INTRODUCTION

Keyword extraction is the process of mining descriptive words from texts. It is a challenging text mining task as keywords provide means for document indexing, search, classification, and clustering. Furthermore, keywords may provide readers with the concept and theme of a text. With the increasing amount of stored online documents, the problem has gathered further importance, and the need for automated keyword extraction techniques has emerged.

Keyword extraction techniques differ by various aspects, such as the type of algorithm they employ, the type of the document they focus on, and the data structure they use to represent documents. Primarily, keyword extraction techniques can be classified as supervised and unsupervised. Supervised keyword extraction is considered as a binary classification task where words of a document are assigned either to the keyword or the non-keyword class. In literature, there are supervised keyword extraction studies that employ support vector machines (Ni, Liu, & Zeng, 2012; Armouty & Tedmori, 2019), neural networks (Azcarraga, Liu, & Setiono, 2012; Tafti et al., 2019), and conditional random fields (Patel & Caragea, 2019; Anju, Ramesh, & Rafeeque, 2018). Unsupervised methods for keyword extraction follow unsupervised learning methods. These include simple statistics methods that focus on word statistics such as *tf-idf* score (Sun, Wang, & Xia, 2017; Yao, Pengzhou, & Chi, 2019) and term relatedness (Campos et al., 2020); NLP-based approaches employ NLP tools such as lexical chains (Ercan & Cicekli, 2007); and graph-based approaches that focus graph algorithms such as node ranking (Florescu & Caragea, 2017). Keyword extraction studies also differ through text representation models. Schemes such as simple graphs (Tixier, Malliaros, & Vazirgiannis, 2016; Florescu & Caragea, 2017; Biswas, Bordoloi, & Shreya, 2018), hypergraphs (Bellaachia & Al-Dhelaan, 2014), and bag-of-words (Hulth, 2003) are extensively used. Furthermore, keyword extraction studies differ by the type of document they focus on. There are keyword extraction techniques developed specifically for microblog posts (Biswas, 2019), scientific documents (Thushara, Krishnapriya, & Nair, 2018), and news articles (Yao et al., 2019).

In this study, we focus on unsupervised graph-based keyword extraction. Such studies represent a text as a graph where nodes represent the text's unique words and edges indicate relations between nodes. Such approaches formulate the keyword extraction problem as a node-ranking problem. An essential issue in this approach is the initialization of node weights. A good initial weight may produce high-quality keywords and speed up the process. Motivated by the discussion on the relationship between the position of a sentence and its informativeness presented in (Lynn, Lee, Choi, & Kim, 2017), in this study, we investigate the performance of three initial weight assignment procedures for nodes, namely Word Position Bidirectional (WPB), Sentence Position (SP), and Sentence Position Bidirectional (SPB) and compare them against initial node assignment procedure of PositionRank (Florescu & Caragea, 2017), namely WP. WP assigns higher initial weights to words that appear at the beginning of a text. In WPB, words appearing at the beginning and at the end of a text are assigned with higher initial weights than those appearing in the middle of a text. In SP, words appearing at the first sentences of a text are assigned with higher initial weights, and in SPB, words appearing in the sentences either close to the beginning or the end of a text is assigned to higher initial weights than words appearing in the middle sentences. Hence, WP and WPB consider word positions in weight assignment while SP and SPB consider sentence positions.

The performance of the initial weight assignment techniques is evaluated using six benchmark datasets, and the results are statistically analyzed. The experimental results regarding all datasets show that SP ranks best in terms of F1-score; however, it does not statistically differ from WP. Regarding the datasets whose author assigned keywords are mostly populated in the very beginning of texts, WP performs better than other weighting procedures but statistically differs only from WPB. Regarding the datasets whose author assigned keywords are evenly spread in the text, SP ranks first and statistically differs from WP. WPB always ranked last.

The organization of the paper is as follows. In Section 2, we provide the general framework of the unsupervised, graph-based keyword extraction procedure, and introduce the PositionRank algorithm in some detail. In

Section 3, we introduce the proposed word weighting heuristics. In Section 4, we introduce the datasets used to evaluate the proposed word weighting heuristics, the experimental setting, and discuss the findings. The last section concludes the paper.

2. BACKGROUND

This section introduces the general framework for unsupervised, graph-based keyword extraction, and later explains the PositionRank algorithm.

2.1. Graph-based Keyword Extraction

Graph-based keyword extraction is an unsupervised procedure (Biswas et al., 2018; Beliga, 2014). The general framework for graph-based keyword extraction consists of text preprocessing, word-graph construction, candidate keyword generation, and keyword extraction steps. Below we describe these steps.

- **Text Preprocessing:** In this step, a text is tokenized, and tokens are annotated with the part of speech tags. These tags are later used to filter words of certain types. This step also includes the removal of stop words and unimportant words.
- **Word-graph Construction:** In this step, a graph called *word-graph* is constructed to represent a text document. In this representation, unique words of a text constitute the nodes, and edges imply certain relations among the words. In word-graphs, nodes are also assigned with initial weights that indicate the importance of the words they represent. TextRank (Mihalcea & Tarau, 2004) considers all words equally important and assigns initial weight 1 to all nodes. PositionRank (Florescu & Caragea, 2017) determines the initial weight of a word according to its positions in the document. Keyword Extraction using Collective Node Weight (KECNW) (Biswas et al., 2018) considers several features of a node such as the distance of the node from the central node, node's selectivity centrality, and the position of the word. Keyword from Weighted Graph (KWG) (Biswas, 2019) aggregates word frequency and degree of the node in the initial weight assignment. In TextRank and

PositionRank, edges connect nodes that represent co-occurring words, i.e. words that appear within a predefined size of windows. In KECNW and KWG edges connect nodes that are immediate neighbors.

- **Candidate Keyword Generation:** Candidate keywords are generated by applying a node-ranking algorithm on the word-graphs. These ranking algorithms are generally derived from the Hyperlink-Induced Topic Search (HITS) algorithm (Kleinberg, 1999) and the PageRank algorithm (Brin & Page, 1998).
- **Keyword Extraction:** *k* nodes with highest ranks are extracted from a word-graph as keywords. However, in word-graphs nodes represent individual words and this procedure generates single-word keywords. To generate key-phrases, i.e. keywords with two or more words, keyword extraction algorithms employ various heuristics. The PositionRank algorithm concatenates one-word keywords that appear in contiguous positions of the original text to generate candidate keyphrases. Scores of the candidate keyphrase is calculated by summing individual words' scores. Then it selects top-*k* ranking candidate keywords / keyphrases as a solution. The TextRank algorithm, on the other hand, firstly selects top-*k* ranking single word-keywords from the word-graph and then merges those that appear in contiguous positions of the original text.

2.2. The PositionRank Algorithm

PositionRank is an unsupervised, graph-based algorithm proposed for keyword extraction from scientific publications. In the data-preprocessing step of PositionRank, all words other than nouns and adjectives are removed. Word-graph of PositionRank is weighted and undirected, where nodes represent unique words of the preprocessed text and edges connect nodes representing words that are at most d-distant from each other. Edge weights, w_{ij}, indicate the number of the co-occurrences of two words. Initial node weights are assigned relative to the positions of the words they represent. The first word of a text has initial weight 1/1; word appearing at position p has weight $1/p$. If a word appears in multiple positions their weights are summed. PositionRank follows the PageRank's node ranking procedure. The node weighting procedure is formulated in Equation 1, where $S(v_i)$ is the weight of node v_i after iteration i. $O(v_j)$ is the summation of the edge weights of the nodes that are adjacent to v_j. w_{ji} is the weight of the edge between v_i and v_j, p_i is the initial weight of node v_i. λ is a dumping factor determining the transition probability from a node to the next node.

$$S(v_i) = (1-\lambda).\widetilde{p_i} + \lambda \sum_{v_j \in Adj(v_i)} \frac{w_{ji}}{O(v_j)} S(v_j) \tag{1}$$

PositionRank is assumed to converge when nodes' weights differ by at most 0.001 between two consecutive iterations or the iteration number reaches 100. Once the node weights converge, PositionRanks sorts nodes based on their ranks. If two or more words are immediate neighbors, they are concatenated to form a keyphrase with a weight equal to the summation of weights of the words in the keyphrase. Top *k*-keywords/key-phrases are selected as the solution.

3. THE INITIAL WEIGHTS PROCEDURES

In (Lynn et al., 2017), sentences forming a text are classified into three groups: topic sentences, supporting sentences, and concluding sentences. The study states that the topic sentences appear at the beginning of a text, concluding sentences appear at the end, and supportive sentences are placed in between. Furthermore, the study cites that topic and concluding sentences are more informative compared to supportive sentences hence are more likely to contain keywords. Motivated by these observations we investigate three procedures for initial weight assignment for words, namely SentenceRank (SP), SentenceRankBidirectional (SPB), and WordPositionBidirectional (WPB). Below we describe these procedures:

- **Sentence Position (SP):** In this approach, we assign weights to sentences based on their positions, i.e. the first sentence gets weight 1 and the second sentence gets weight 2. The weight of a word is calculated according to the weight of the sentence it appears in. A word that appears in a sentence with weight *i*, has initial weight of 1/*i*. If a word appears in multiple sentences, individual weights of the word are summed. Equation (1) formulates this initial weight assignment procedure, where w_i is a word in sentence S_t, t indicating position of the sentence.

$$SP(w_i) = \frac{1}{t}, w_i \in S_t \quad \textbf{(1)}$$

- **Sentence Position Bidirectional (SPB):** In this approach, the first and the last sentences of a text are assigned weight 1; the second sentence and the second to the last sentence are assigned weight 2 and so on. Similar to SP, a word that appears in a sentence with weight i, has initial weight $1/i$. If a word appears in multiple sentences, these individual weights of the word are summed. Both in SP and SPB, words appearing in the same sentence have equal weights. This assignment procedure is formulated in Equation (2), where n is the number of sentences in the text, t is the position of the sentence S_t.

$$SPB(w_i) = \begin{cases} \frac{1}{t}, & w_i \in S_t, \quad t < n/2 \\ \frac{1}{n+1-i}, & w_i \in S_t, \quad t \geq n/2 \end{cases} \quad \textbf{(2)}$$

- **Word Position Bidirectional (WPB):** In this approach, we follow weighting procedure of PositionRank, however, we also favor words that appear close to the end of a text. In WPB, the first and the last words are assigned with initial weight 1/1, the second word and the second to the last word are assigned with initial weight of 1/2 and so on. This procedure is formulated in Equation (3), where i indicates the position of a word, n is the number of words in the text.

$$WPB(w_i) = \begin{cases} \frac{1}{i}, & i < n/2 \\ \frac{1}{n+1-i}, & i \geq n/2 \end{cases} \quad \textbf{(3)}$$

When compared to ranking procedure of PositionRank, SP and SPB assign more gradual initial weights to words. However, WPB is steep in initial weight assignment while also favoring last words of a text.

4. EXPERIMENTS

In this section, we firstly introduce the datasets and metrics used to evaluate the performance of proposed initial weight assignment procedures. Later, we discuss the experimental findings.

4.1. Datasets and Experimental Setting

- **Inspec:** It is a collection of abstracts of 2000 scientific journals written in English and related to computer science and information technology. The dataset is introduced in (Hulth, 2003) and is one of the most cited datasets in keyword extraction studies.
- **Nguyen:** This dataset is introduced in (Nguyen & Kan, 2007) and consists of 211 academic conference papers written in English. Each article has two keyword sets: one provided by the authors of the article and the other provided by annotators. In this study, we use the union of these keywords sets. Although the dataset provides full text of the articles, we consider the abstracts.
- **SemEval2010:** This dataset consists of 284 scientific papers written in English, compiled from the ACM library, and focusing on various domains such as economics, information retrieval, and multi-agent systems. The dataset is introduced in (Kim, Medelyan, Kan, & Baldwin, 2010).
- **SemEval2017:** The SemEval2017 dataset is introduced in (Augenstein, Das, Riedel, Vikraman, & McCallum, 2017) and consists of 500 academic papers written in English related to computer science, material sciences, and physics.
- **WWW:** This dataset is introduced in (Gollapalli & Caragea, 2014) and consists of abstracts of 1330 papers presented in the World Wide Web conference between 2004 and 2014.

- **KDD:** This dataset is introduced in (Gollapalli & Caragea, 2014) and contains abstracts of 755 papers presented in ACM Conference on Knowledge Discovery and Data Mining between 2004 and 2014.

To evaluate the performance of the initial weight assignment procedures, precision, recall, and F-score are used. In the context of keyword extraction, precision refers to the fraction of the number of correctly extracted keywords over the total number of keywords extracted; recall refers to the fraction of the number of correctly extracted keywords over the number of keywords assigned to the document, and F-score is the harmonic mean of precision and recall. Recall, precision, and F-score are defined, respectively, in equations (4), (5), and (6).

$$Recall = \frac{number\ of\ correctly\ matched\ keywords}{number\ of\ assigned\ keywords} \quad \textbf{(4)}$$

$$Precision = \frac{number\ of\ correctly\ matched\ keywords}{number\ of\ extracted\ keywords} \quad \textbf{(5)}$$

$$F - score = 2x\frac{precision\ x\ recall}{precision + recall} \quad \textbf{(6)}$$

Friedman's test and Nemenyi post-hoc test are used to statistically analyze the results. Friedman's test is a non-parametric test to detect differences of variance by ranks across multiple attempts. The null hypothesis for the Friedman test is that there are no differences among the attempts, i.e. groups come from populations with the same median (Pereira, Afonso, & Medeiros, 2015). Although the Friedman test can discover if any of the attempts statistically differ, it cannot detect the differing attempts. Nemenyi post-hoc test is employed to detect the differing attempts. It performs pairwise multiple comparisons of the ranked data. To this aim, the pairwise multiple comparison of mean ranks (PMCMR) package (Pohlert, 2016) of R is used. The tool is also used to create the plots. The visual representation of the Nemenyi test consists of methods that are placed on an axis according to their mean rank and a critical difference (CD) ruler. If the difference in average rank between two attempts, say *i* and *j*, exceeds critical difference,

$R_i - R_j > \Delta_\alpha$, then the performance of algorithm i is better than the performance of algorithm j. Methods that do not statistically differ are connected via straight lines. Methods with higher ranks, in the context of this study, are assumed to perform better compared to methods with lower ranks.

In the experiments, we set the window size, *d*, to 3, and the dumping factor in PageRank algorithm, λ, to 0.85, which is the common practice (Mihalcea & Tarau, 2004; Florescu & Caragea, 2017; Biswas, Bordoloi, & Shreya, 2018). We evaluated the proposed initial node weighting procedures for 2, 4, 6, 8, 10, and 15 keywords. In the subsequent tables, cells highlighted in yellow indicate the highest scores.

4.2. Results

In Table 1, we report the recall results. As the results show, WP achieved the best results for the WWW and KDD datasets. For the Nguyen dataset, WP achieved the best score for four cases and SP for three cases. For the SemEval2010 dataset, best scores are obtained for SP and SPB. For the SemEval2017, SP achieved the best score for five cases and SPB for three cases. For the Inspec dataset, SP achieved the best score for five cases, WP for one case, and SPB for one case. WPB did not score any best result.

In Table 2, we report the precision results. WP achieved the highest results for the WWW and KDD datasets for all cases, and all but for the Inspec dataset. For SemEval2010, SemEval2017, and Inspec datasets, SP and SPB achieved the best results for the most of the cases. More specifically, for the SemEval2010 dataset, SP achieved the best result for all cases and SPB for three cases. For the SemEval2017 dataset, SP achieved the best result for five cases, and SPB for three. For the Inspec dataset, SP achieved the best results for five cases, WP and SPB for one cases. Similar to recall results, WPB did not score any best result.

Table 1. Recall results.

Dataset	Method	R@2	R@4	R@6	R@8	R@10	R@15
WWW	WP	0.057	0.105	0.138	0.16	0.178	0.206
	SP	0.044	0.09	0.12	0.142	0.16	0.194
	WPB	0.041	0.082	0.112	0.137	0.153	0.186
	SPB	0.045	0.089	0.115	0.138	0.159	0.193
KDD	WP	0.06	0.121	0.161	0.188	0.207	0.234
	SP	0.052	0.108	0.15	0.171	0.184	0.23
	WPB	0.05	0.091	0.131	0.16	0.179	0.211
	SPB	0.05	0.105	0.144	0.163	0.182	0.225
Nguyen	WP	0.041	0.082	0.112	0.145	0.164	0.198
	SP	0.042	0.08	0.112	0.137	0.16	0.201
	WPB	0.028	0.074	0.106	0.13	0.149	0.186
	SPB	0.04	0.079	0.111	0.134	0.158	0.198
SemEval2010	WP	0.018	0.032	0.047	0.06	0.07	0.09
	SP	0.019	0.036	0.049	0.063	0.073	0.095
	WPB	0.017	0.028	0.044	0.058	0.069	0.091
	SPB	0.019	0.035	0.05	0.063	0.073	0.094
SemEval2017	WP	0.049	0.092	0.131	0.168	0.2	0.272
	SP	0.052	0.098	0.141	0.176	0.208	0.279
	WPB	0.049	0.094	0.131	0.169	0.2	0.27
	SPB	0.052	0.096	0.138	0.172	0.21	0.279
Inspec	WP	0.065	0.107	0.144	0.174	0.2	0.248
	SP	0.063	0.109	0.148	0.18	0.208	0.257
	WPB	0.062	0.106	0.143	0.173	0.2	0.247
	SPB	0.062	0.108	0.148	0.178	0.207	0.256

Table 2. Precision results.

Dataset	Method	P@2	P@4	P@6	P@8	P@10	P@15
WWW	WP	0.113	0.108	0.094	0.084	0.075	0.059
	SP	0.087	0.089	0.082	0.074	0.067	0.056
	WPB	0.08	0.083	0.077	0.071	0.065	0.053
	SPB	0.088	0.089	0.078	0.072	0.066	0.056
KDD	WP	0.109	0.111	0.099	0.086	0.076	0.058
	SP	0.1	0.101	0.092	0.08	0.068	0.057
	WPB	0.094	0.082	0.079	0.074	0.066	0.052
	SPB	0.097	0.099	0.088	0.075	0.067	0.056
Nguyen	WP	0.196	0.189	0.174	0.168	0.155	0.126
	SP	0.172	0.175	0.165	0.158	0.15	0.129
	WPB	0.141	0.164	0.155	0.147	0.138	0.118
	SPB	0.165	0.167	0.161	0.15	0.144	0.126
SemEval2010	WP	0.136	0.118	0.117	0.112	0.106	0.091
	SP	0.142	0.135	0.123	0.119	0.11	0.097
	WPB	0.123	0.101	0.108	0.108	0.103	0.092
	SPB	0.14	0.131	0.123	0.118	0.11	0.097
SemEval2017	WP	0.383	0.368	0.348	0.335	0.322	0.296
	SP	0.412	0.392	0.376	0.356	0.338	0.306
	WPB	0.382	0.369	0.349	0.339	0.322	0.296
	SPB	0.403	0.381	0.37	0.348	0.339	0.307
Inspec	WP	0.355	0.302	0.278	0.257	0.242	0.214
	SP	0.352	0.313	0.29	0.27	0.255	0.223
	WPB	0.339	0.302	0.279	0.258	0.244	0.214
	SPB	0.346	0.309	0.288	0.267	0.253	0.222

In Table 3, we report the F1 scores. Similar to the recall and precision results, WP achieved the best results for the WWW, KDD, and Nguyen datasets. For the SemEval2010, SemEval2017, and Inspec datasets SP and SPB achieved the highest scores for all scores but 2.

Table 3. F1 score results.

Dataset	Method	F1@2	F1@4	F1@6	F1@8	F1@10	F1@15
WWW	WP	0.073	0.103	0.108	0.107	0.103	0.09
	SP	0.057	0.086	0.094	0.094	0.092	0.084
	WPB	0.052	0.08	0.088	0.091	0.088	0.081
	SPB	0.058	0.086	0.09	0.092	0.091	0.084
KDD	WP	0.075	0.112	0.119	0.115	0.108	0.091
	SP	0.067	0.101	0.111	0.106	0.097	0.09
	WPB	0.063	0.083	0.095	0.098	0.094	0.082
	SPB	0.064	0.098	0.106	0.1	0.096	0.088
Nguyen	WP	0.066	0.106	0.126	0.143	0.146	0.142
	SP	0.063	0.102	0.122	0.135	0.143	0.145
	WPB	0.045	0.095	0.115	0.126	0.132	0.134
	SPB	0.059	0.098	0.12	0.13	0.138	0.143
SemEval2010	WP	0.032	0.05	0.067	0.077	0.084	0.089
	SP	0.034	0.057	0.07	0.082	0.087	0.094
	WPB	0.03	0.043	0.062	0.075	0.081	0.091
	SPB	0.033	0.055	0.07	0.082	0.086	0.095
SemEval2017	WP	0.085	0.144	0.184	0.216	0.238	0.272
	SP	0.091	0.153	0.198	0.227	0.248	0.281
	WPB	0.085	0.145	0.184	0.217	0.237	0.271
	SPB	0.09	0.149	0.195	0.222	0.249	0.281
Inspec	WP	0.106	0.151	0.18	0.197	0.208	0.22
	SP	0.104	0.155	0.186	0.205	0.218	0.228
	WPB	0.101	0.15	0.18	0.197	0.209	0.219
	SPB	0.102	0.153	0.186	0.203	0.216	0.227

As seen from the results, WP performs better for WWW, KDD, and Nguyen datasets while SP performs better for SemEval2010, SemEval2017, and Inspec. To understand why SP and WP perform better for different datasets, we analyzed the spatial distribution of the author assigned keywords in the documents. As seen in Figure 1, keywords of SemEval2010 and SemEval2017 are evenly distributed within the documents. However, for

KDD and WWW keywords are mostly clustered at the beginning of the documents. As WP assigns higher weights to words that appear early in a document, it performs better than SP for KDD and WWW. On the other hand, SP assigns weights in a more gradual manner, and assigns higher weights to words that appear at the end of a document. The Nguyen and Inspec datasets do not pose such a clear difference with respect to the spatial distribution of the keywords, however decrease in the frequency of the keywords that occur in at the very end are sharper for the Nguyen dataset. This may be the reason SP performs better for Inspec compared to Nguyen.

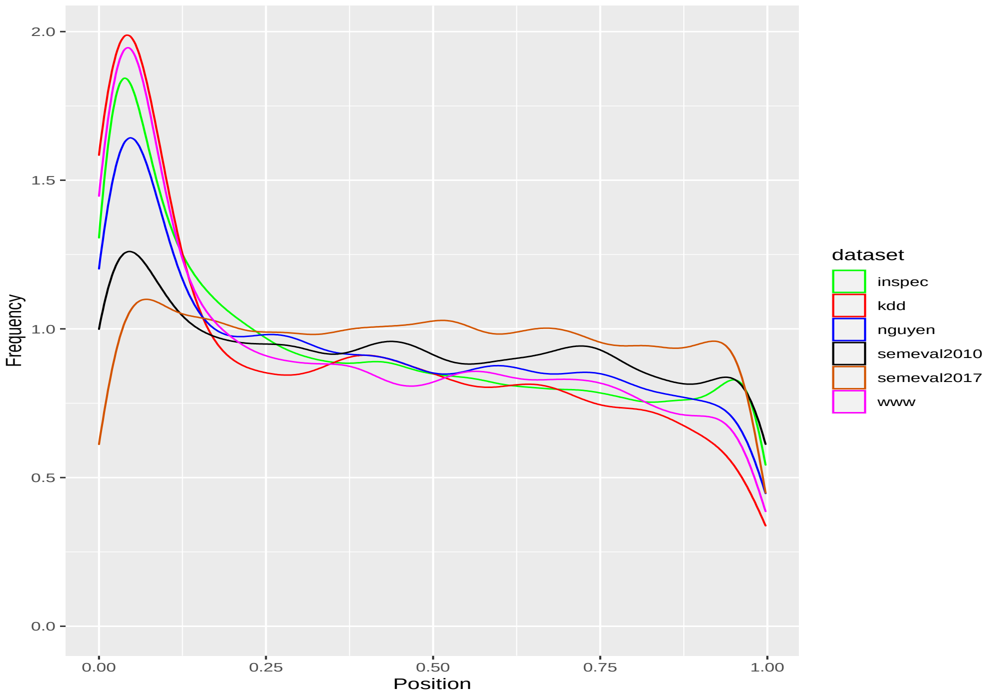
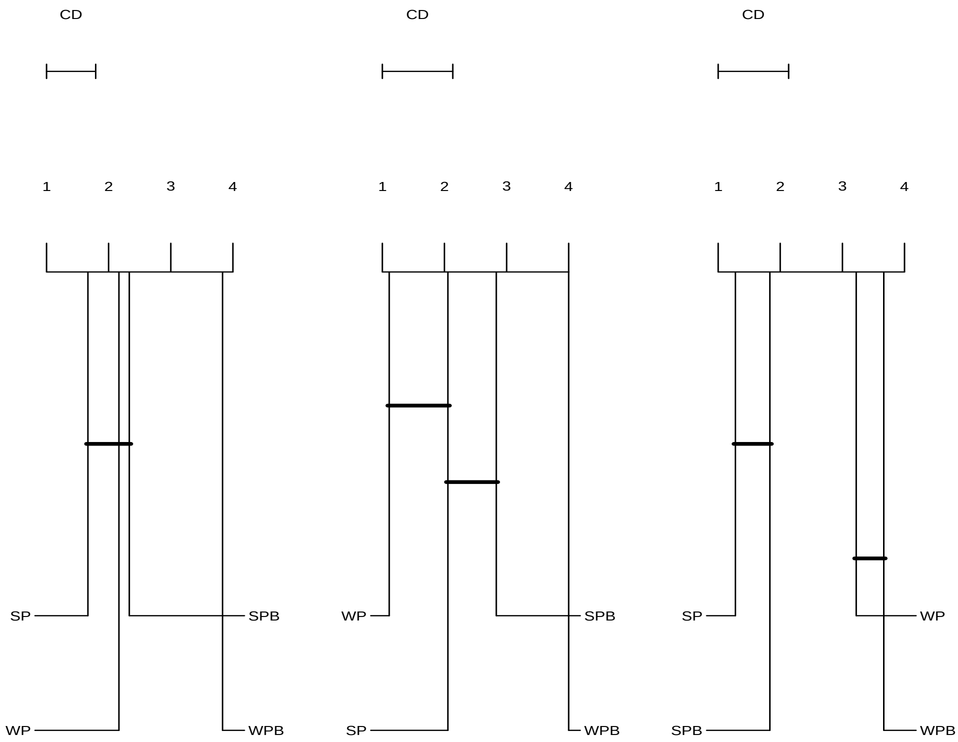


Figure 1. Spatial distribution of the author assigned keywords within documents.

We also statistically analyzed the performance of the weighting procedures. To this aim we conducted the Friedman's test to see if weighting procedures statistically differ, and if so, we conducted the Nemenyi post-hoc test to detect the differing weighing procedures. In Figure 2, we report the Nemenyi test results. More specifically, in Figure 2.a, we compare all methods considering all datasets. In Figure 2.b, we compare all methods for KDD, WWW, and Nguyen dataset where WP performs better compared to the other methods. In Figure 2.c, we compare all methods for SemEval2010, SemEval2017, and Inspec dataset where SP performs better than the other methods.



p=2.2e-16 p=2.2e-16 p=4.441e-16

(a) **(b)** **(c)**

Figure 2. Nemenyi pot-hoc test results.

As seen in Figure 2(a), when all datasets are considered, SP, WP, and SPB do not statistically differ, however, SP ranks the top. WPB statistically differs from the other weighting procedures and ranks the least. In case of the WWW, KDD, and Nguyen datasets, WP ranks the top but does not statistically differ from SP. When considered for the SemEval2010, SemEval2017, and Inspec datasets, SP ranks the best, SPB ranks the second best, however, they do not statistically differ. SP and SPB statistically differ from WB, which ranks third.

As the experimental results show, WPB did not perform well for any of the datasets. We believe that this is due to the shortness of the text. Moreover, the last words in scientific publication abstracts report some experimental finding, which are not likely to be keywords.

In Table 4, we compare performance of WP and SP to three statistical keyword extraction systems, namely TF.IDF, KP-Miner, and Rake; and to a supervised keyword extraction system KEA. The results of the reference methods are reported in (Campos et al., 2020). The results report F1 score for 10 keywords, and the best scores are highlighted. As the results indicate, TF.IDF, KP-Miner, and SP score the best results for two datasets. Although supervised approach did not score any top results regarding the six datasets used in this study, (Campos et al., 2020) reports several datasets for which KEA ranks best.

Table 4. Comparison to other systems, F1@10.

	WP	SP	TF.IDF	KP-Miner	RAKE	KEA
WWW	0.103	0.092	0.130	0.037	0.011	0.072
KDD	0.108	0.097	0.115	0.036	0.006	0.063
Nguyen	0.146	0.143	0.225	0.314	0.002	0.221
SemEval2010	0.084	0.087	0.177	0.261	0.003	0.215
SemEval2017	0.238	0.248	0.181	0.071	0.065	0.201
Inspec	0.208	0.218	0.155	0.047	0.052	0.150

5. CONCLUSION

In this study, we evaluated the performance of three initial node weighting procedures for graph based keyword extraction and compared them against PositionRank's initial node weight assignment procedure. The first procedure, namely WPB, considers positions of the words and assigns higher weights to words that are either at the beginning or at the end of a text. The second procedure, namely SP, assigns higher weights to words that appear in the sentences close to the beginning of the text relative to the position of sentence. The last procedure, namely SPB, assigns higher weights to words that appear in the sentences either close to the beginning of the text or end of the text, relative to the position of the sentences. Hence, WPB assigns weights to words that steeply decrease while SP and SPB assign gradually decreasing weights. Experiments show that, WP performs better for documents whose keywords appear close to the beginning of the document, while SP and SPB perform better for documents whose keywords are evenly spread within the text.

ACKNOWLEDGEMENT

This work is partially supported by TUBITAK with grant number 117E566.

REFERENCES

Armouty, B., & Tedmori, S. (2019). “Automated Keyword Extraction using Support Vector Machine from Arabic News Documents”. In *2019 IEEE Jordan International Joint Conference on Electrical Engineering and Information Technology (JEEIT)*, IEEE, 342-346. doi:10.1109/JEEIT.2019.8717420.

Anju, R. C., Ramesh, S. H., & Rafeeque, P. C. (2018). “Keyphrase and Relation Extraction from Scientific Publications”. In Damodar Reddy Edla, Pawan Lingras, and Venkatanareshbabu K. (Eds), *Advances in Machine Learning and Data Science: Recent Achievements and Research Directives* (pp. 113-120). Vol. 705, Singapore, Springer.

Augenstein, I., Das, M., Riedel, S., Vikraman, L., & McCallum, A. (2017). “Semeval 2017 Task 10: ScienceIE-Extracting Keyphrases and Relations from Scientific Publications”. arXiv preprint arXiv:1704.02853. Retrieved from https://arxiv.org/pdf/1704.02853.pdf

Azcarraga, A., Liu, M. D., & Setiono, R. (2012, June). “Keyword extraction using backpropagation neural networks and rule extraction”. In *the 2012 International Joint Conference on Neural Networks (IJCNN),* IEEE, 1-7. doi:10.1109/IJCNN.2012.6252618.

Beliga, S. (2014). “Keyword Extraction: A Review of Methods and Approaches”. University of Rijeka, Department of Informatics, Rijeka, 1-9.

Bellaachia, A., & Al-Dhelaan, M. (2014). “HG-Rank: A Hypergraph-based Keyphrase Extraction for Short Documents in Dynamic Genre”. In *#MSM, #Microposts2014, 4th Workshop on Making Sense of Micropost,* 42-49.

Biswas, S. K. (2019). “Keyword Extraction from Tweets Using Weighted Graph”. In Pradeep Kumar Mallick, Valentina Emilia Balas, Akash Kumar Bhoi, and Ahmed F. Zobaa (Eds.), *Cognitive Informatics and Soft Computing: Proceeding of CISC 2017* (pp. 475-483). Vol 768, Singapore, Springer.

Biswas, S. K., Bordoloi, M., & Shreya, J. (2018). “A graph based keyword extraction model using collective node weight”. *Expert Systems with Applications,* Vol. 97, 51-59. doi:10.1016/j.eswa.2017.12.025.

Brin, S., & Page, L. (1998). “The anatomy of a large-scale hypertextual web search engine”. *Computer Networks and ISDN Systems.* Vol. 30, Issues 1-7, 107-117.

Campos, R., Mangaravite, V., Pasquali, A., Jorge, A., Nunes, C., & Jatowt, A. (2020). “YAKE! Keyword extraction from single documents using multiple local features”. *Information Sciences,* Vol. 509, 257-289. doi:10.1016/j.ins.2019.09.013.

Ercan, G., & Cicekli, I. (2007). “Using lexical chains for keyword extraction”. *Information Processing & Management,* 43(6), 1705-1714. doi:10.1016/j.ipm.2007.01.015.

Florescu, C., & 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,* Vol. 1, 1105-1115.

Gollapalli, S. D., & Caragea, C. (2014). “Extracting keyphrases from research papers using citation networks”. In *AAAI'14: Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence,* 28(1), 1629-1635.

Hulth, A. (2003). “Improved automatic keyword extraction given more linguistic knowledge”. In *Proceedings of the 2003 Conference on Empirical Methods in Natural Language Processing,* 216-223.

Kim, S. N., Medelyan, O., Kan, M.-Y., & Baldwin, T. (2010). “SemEval-2010 Task 5 : Automatic Keyphrase Extraction from Scientific Articles”. In *Proceedings of the 5th International Workshop on Semantic Evaluation,* 21-26.

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

Lynn, H. M., Lee, E., Choi, C., & Kim, P. (2017). “SwiftRank: An Unsupervised Statistical Approach of Keyword and Salient Sentence Extraction for Individual Documents”. *Procedia Computer Science,* Vol. 113 , 472-477.

Mihalcea, R., & Tarau, P. (2004). “Textrank: Bringing order into text”. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing,* 404-411.

Nguyen, T. D., & Kan, M.-Y. (2007). “Keyphrase Extraction in Scientific Publications”. In *10th International Conference on Asian Digital Libraries, ICADL 2007,* DBLP, 317-326.

Ni, W., Liu, T., & Zeng, Q. (2012). “Extracting keyphrase set with high diversity and coverage using structural svm”, In APWeb'12: Proceedings of the 14th Asia-Pacific international conference on Web Technologies and Applications, 122-133.

Patel, K., & Caragea, C. (2019). “Exploring word embeddings in crf-based keyphrase extraction from research papers”. In *K-CAP ’19: Proceedings of the 10th International Conference on Knowledge Capture*, 37-44.

Pereira, D. G., Afonso, A., & Medeiros, F. M. (2015). “Overview of Friedman's test and post-hoc analysis”. *Communications in Statistics-Simulation and Computation,* 44(10), 2636-2653.

Pohlert, T. (2016). “The Pairwise Multiple Comparison of Mean Ranks Package (PMCMR)”. *R package.* Retrieved from https://cran.r-project.org/web/packages/PMCMR/vignettes/PMCMR.pdf

Sun, P., Wang, L., & Xia, Q. (2017). "The Keyword Extraction of Chinese Medical Web Page Based on WF-TF-IDF Algorithm". In *2017 International Conference on Cyber-enabled Distributed Computing and Knowledge Discovery (CYBERC),* 193-198. Retrieved from https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8250358&tag=1

Tafti, A. P., Wang, Y., Shen, F., Sagheb, E., Kingsbury, P., & Liu, H. (2019). "Integrating word embedding neural networks with pubmed abstracts to extract keyword proximity of chronic diseases". In *2019 IEEE EMBS International Conference on Biomedical & Health Informatics (BHI),* 1-4.

Thushara, M. G., Krishnapriya, M. S., & Nair, S. S. (2018). "Domain Classification of Research Papers Using Hybrid Keyphrase Extraction Method". In Pankaj Kumar Sa, Sambit Bakshi, Ioannis K. Hatzilygeroudis, and Manmath Narayan Sahoo (Eds.), *Recent Findings in Intelligent Computing Techniques : Proceedings of the 5th ICACNI* (pp. 387-398). Vol. 708, Singapore, Springer.

Tixier, A., Malliaros, F., & Vazirgiannis, M. (2016). "A Graph Degeneracy-based Approach to Keyword Extraction". In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, 1860-1870.

Yao, L., Pengzhou, Z., & Chi, Z. (2019). "Research on News Keyword Extraction Technology Based on TF-IDF and TextRank". In 2019 IEEE/ACIS 18th International Conference on Computer and Information Science (ICIS), 452-455.