https://doi.org/10.2352/EI.2023.35.3.MOBMU-371

San Antonio Research Partnership Portal: Evaluating Keyword Extraction Tools to Automate Matchmaking for Community Research Partnership

Mohammad Nadim, David Akopian; Department of Electrical Computer Engineering, The University of Texas at San Antonio, San Antonio, Texas, USA
Adolfo Matamoros; Department of Civil Environmental Engineering, The University of Texas at San Antonio, San Antonio, Texas, USA

Abstract
Finding research professionals and collaborators to address community problems continues to be a significant barrier for many local government agencies. Research collaboration between researchers from universities, industries and local government agencies can be tremendously useful to all organizations. San Antonio Research Partnership Portal is a collaborative initiative to bring researchers and local government agencies in one place to solve community concerns. In this paper, we investigate the performance of popular keyword extraction tools by measuring the effectiveness of identifying the keywords from research opportunities. The extracted keywords are used in an automated process for San Antonio Research Partnership Portal to match academic researchers with corresponding research opportunities.

Index Term *- Community Research Partnership, Keyword Extraction Tool, Information Extraction, Web Application, Evaluation*

Introduction
Community-based research is a collaborative research partnership approach in which academic researchers, local government officials, members of the community, representatives of organizations, and others are all equally involved in all facets of the research process, contributing their knowledge, and sharing in the formulation of policy [1, 2]. Both the creators and the potential users of community research must collaborate to overcome the difficulties [3, 4]. Local governments can grant university researchers access to city data, enabling their work to have a wider impact beyond the confines of traditional scholarly publications, a better chance of obtaining research funding, and a more methodical and intentional approach to community contact. And to get the best outcome from community research partnership it is very important to match the collaborators.

The most valuable resource in modern world is data. As humans interact through many data types such as photographs, videos, music, and textual streams on various web- sites throughout the internet, this data is extremely valuable. The knowledge concealed in these massive amounts of data may be used to conduct several activities. We concentrated mostly on textual data. To extract essential insights or to provide data insights for textual data in a clear manner, there are two basic ways. The first option is manual analysis, which entails manually processing the data and beginning to log the information that appears to be valuable. Another method is automated text analysis, which uses computational tools to do text analysis. Due to large data quantities, the latter option is more enticing, as manually processing large amounts of text will be impossible. Automatic Keyword/key-Extraction (KE) is a technique for displaying essential information that is extensively used to annotate articles to represent their possible categories [5]. It is extremely important in the field of information retrieval. It's because keywords from any piece of data indicate whether the data source is appropriate and relevant to other significant domains and/or subject.

Firoozeh et al. conducted a comprehensive review of KE methods along with their benefits and drawbacks [6]. The authors also presented benchmark data-set for keyword extraction. Similar work of exploring state-of-the-art KE has been done by others that includes classifying KE methods, data prepossessing, advantages, challenges, and underlying data-sets [7] - [11]. Recent studies broadly discuss the technical details of KE methods, offer interesting insights to highlight open issues, and present a comparative experimental result [12]. We have experimented with several python-based implementations of KE methods to find the fastest and most efficient KE tool to automate the matchmaking process of community research partnership platform. The paper is composed as follows: Section II introduces prior research on evaluation of KE tools. In section III, San Antonio Research Partnership Portal is introduced and followed by a list of KE tools and benchmark data-sets selected for our experiment in Section IV. Section V includes the experimental results and discussion. Finally, we conclude this paper with future research direction in Section VI.

II. Related Works
The reliability of evaluation techniques and approaches and an examination of their flaws remain two of the largest problems for keyword extraction. There have been several works on evaluating the performance of KE tools. But unfortunately, there has been a lack of research on integrating faster and efficient KE tool on a community research partnership web platform. Using several evaluation techniques and metrics, Papagiannopoulou and Tsoumakas provided an empirical study comparing commercial APIs with cutting-edge as well as popular unsupervised methodologies [12]. They thoroughly analyzed the exact and partial matching approaches in their evaluation study, recommending that one consider their average and emphasizing

the need for methods that consider the semantic similarity of anticipated and golden keywords. Piskorski et al. conducted a study to select the most appropriate state-of-the-art keyword extraction technique for indexing news articles in a large-scale real world news analysis engine [13]. The authors evaluated the algorithms with random samples of 50 news articles published in 2020. Giarelis et al. performed a comparative assessment of five different KE methods and experimented the KE methods with different scientific and new articles [14]. A similar approach of our experiment. Kumar et al. also investigated the performance of three popular KE methods by measuring their effectiveness in identifying keywords [15]. In this paper, we have experimented with more KE tools than previous studies to have a broader insight. As well as we have evaluated the KE tools integrated with San Antonio Research Partnership Portal which is a partnership web portal for community research [16].

III. San Antonio Research Partnership Portal

A prototype project from the R&D League effort of the Office of Innovation, City of San Antonio, is the San Antonio Research Partnership Portal. Initiation of the project took place in the first quarter of 2021. The Office of Innovation in the City of San Antonio launched the R&D League, a research and development initiative, in partnership with the University of Texas at San Antonio (UTSA), the Southwest Research Institute (SwRI), and the United Services Automobile Association (USAA) [17]. For the sake of the people of San Antonio, this league's goal is to create cross-sector, multidisciplinary alliances to conduct innovative concept research, promote evidence-based policy making, and pursue the cutting edge of innovation. The R&D League has a goal to create and develop a central platform to help the community research partnership. This initiative's goal is to develop an interactive open research partnership portal that will enable university academic researchers to participate more actively in open collaboration projects with government agencies, local departments, other organizations, universities, and the community. It is important to match the research opportunities with academic researchers' research interests to facilitate the research partnership process. And for that, we will extract the important keywords from research opportunities and match it with researchers' research interests.

IV. Keyword Extraction

The process of choosing terms that accurately describe a document is known as keyword extraction. Its objective is to provide a brief synopsis of a lengthy text. The process of automatically extracting keywords using computational technologies is known as automatic keyword extraction (AKE). It can give the user a brief summary of the information in the pertinent document. We selected the KE methods that have python-based implementation for our experiment as the San Antonio Research Partnership Portal is a python-based web application. In this section, we will briefly describe the selected KE tools, benchmark datasets, and evaluation metrics. Table 1 shows the selected KE tools, their approach, and parameter settings.

A. Keyword Extraction Tools

RAKE Rapid Automatic Keyword Extraction (RAKE) is a method for quickly extracting keywords from a single document that is unsupervised and independent of any particular domain [18]. A set of phrase and word delimiters as well as a list of stop words are included in the input parameters for the RAKE Algorithm. It uses stop words and phrase delimiters to separate the text into candidate keywords. Most of these candidate keywords are words that help a developer find the exact keyword needed to extract data from a text.

YAKE Yet Another Keyword Extractor (YAKE) is a quick and easy unsupervised automatic keyword extraction technique that uses text statistical data gathered from individual documents to identify the most crucial keywords in a text [19]. The system is not dependent on dictionaries, external corpora, text size, language, or domain, nor does it require training on a specific collection of documents. There is a web API available for YAKE.

Gensim Gensim is an open-source framework that uses contemporary statistical machine learning for unsupervised topic modeling, document indexing, retrieval by similarity, and other natural language processing features [20]. Gensim stands out from most other machine learning software solutions because it is built to handle massive text collections using incremental online algorithms and data streaming.

SpaCy SpaCy is a Python software package for advanced natural language processing that is free and open source [21]. Software for use in production is spaCy's primary focus. SpaCy has a lot of statistical models created for many different languages. Additionally, it enables deep learning workflows that let users combine statistical models trained using well-known machine learning libraries with libraries of their own.

TextRank TextRank is a python tool to extract keywords and summarizes text [22]. By examining whether words follow one another, the algorithm can identify how closely they are related. The PageRank method is then used to order the text's most crucial phrases. The spaCy pipeline typically works with TextRank.

PKE Python Keyphrase Extraction (PKE) is an open-source python-based keyword and key phrase extraction library [23]. Currently the following non-learning ranking and graph-based KE models are implemented in PKE.

- TF-IDF: Term Frequency-Inverse Document Frequency (TF-IDF) is a weighting measure that identifies whether a word is significant in a certain document of a corpus. A document's keywords ought to have a high TF-IDF rating.
- KP-Miner: KP-Miner is statistical approach to extract key phrases [24]. Candidates for a key phrase are groups of words devoid of stopwords or punctuation. Candidates that don't show up three times or that first appear outside of a specific position are eliminated. Following that, candidates are weighted using a modified TF-IDF method that takes document length into consideration.
- TopicRank: Another unsupervised graph-based key word extractor is TopicRank [25]. Unlike TextRank, the graph's nodes are topics, and each topic is a collection of comparable single- and multi-word expressions.

Table 1: Selected KE tools with parameter configuration.

Algorithm	Parameters	Approach
RAKE	-	Statistical
YAKE	lan="en", n=3, top=10, windowsSize=3	Statistical
Gensim	words=10, lemmatize=True	Statistical+Deep Learning
SpaCy	not is_stop and not is_punct and not like_num	Statistical+Deep Learning
TextRank	-	Graph-based
TF-IDF	language='en', grammar = "NP: {<ADJ>*<NOUN | PROPN>+}	Non-learning Ranking
KPMiner	language='en', grammar="NP: {<ADJ>*<NOUN | PROPN>+}	Non-learning Ranking
TopicRank	language='en, pos={'NOUN','PROPN','ADJ','ADV'}	Graph-based
PositionRank	language='en', maximum_word_number=5, window=3, pos='NOUN','PROPN','ADJ','ADV'	Graph-based
SingleRank	language='en', window=3, pos={'NOUN','PROPN','ADJ','ADV'}	Graph-based
KeyBERT (mmr)	keyphrase_ngram_range=(1,3), top_n=10, stop_words='english', use_mmr=True	Deep Learning
KeyBERT (maxsum)	keyphrase_ngram_range=(1,3), top_n=10, stop_words='english', use_maxsum=True	Deep Learning

- PositionRank: In order to calculate a position-biased PageRank score for each word, PositionRank, a graph-based model, additionally takes into account the positions and frequency of words inside a document [26].
- SingleRank: TextRank is expanded into SingleRank, but there are two key changes [27]. Instead of having an unweighted graph in TextRank, the edges are first given a weight. Second, unlike TextRank, which only keeps the top one-third of vertices depending on their ratings, SingleRank keeps every uni-gram.

KeyBERT KeyBERT is a simple and user-friendly keyword extraction technique that uses BERT word embedding to provide the keywords and key phrases that are most comparable to a given document. BERT, also known as Bi-directional Encoder Representation of Transformers, is an encoder-only model designed to learn deep bidirectional representations of text segments from an unlabeled text.

- mmr: Maximal Marginal Relevance (mmr) considers the similarity of keywords/key phrases with the document, along with the similarity of already selected keywords and key phrases [28].
- maxsum: For maxsum, first the document's 2*top_n keywords are extracted [29]. Then these keywords' pairwise similarities are computed. The algorithm then extracts the terms that are most relevant and least similar to one another.

B. Benchmark Data-sets

To evaluate the python-based unsupervised KE tools, three different data-sets are selected. Table 2 shows the details of selected data-sets.

SemEval 2010 (Task 5) SemEval2010 consists of 244 full scientific papers from four different areas of computer science research, each ranging from 6 to 8 pages and taken from the ACM Digital Library (one of the most popular data-sets previously used for keyword extraction evaluation) (distributed systems; information search and retrieval; distributed artificial intelligence – multiagent systems; social and behavioral sciences – economics) [30]. Each article contains two sets of keywords: one set supplied by the author (which is included in the original pdf file), and the other set assigned by expert editors; both sets of keywords may or may not be included directly in the text.

Inspec Inspec is made up of 2,000 abstracts of computer science journal articles that were gathered between 1998 and 2002 [31]. Two sets of keywords are assigned to each document: the controlled keywords, which are manually selected keywords that appear in the Inspec thesaurus but may not appear in the document, and the uncontrolled keywords, which are freely chosen by the editors and are not limited to the thesaurus or the document. In the repository, the ground truth is defined as the union of the two sets.

SemEval 2017 (Task 10) SemEval2017 is made up of 500 sentences drawn randomly from 500 ScienceDirect journal articles in the fields of computer science, material sciences, and physics [32]. A number of keywords were chosen for each manuscript by a professional annotator and one undergraduate student. When there is a difference of opinion between the two annotators, the expert's annotation takes precedence. Extraction of keywords and relationships from scholarly articles was the first goal.

C. Evaluation Metrics

The comparison of the human annotated data set and the system generated results is frequently used to evaluate KE tools. Precision, recall, and f-score are among the most often used assessment metrics in the Information Retrieval area for such comparisons. The ratio of successfully recognized entities to the total number of predicted entities by the systems is known as precision. Recall, on the other hand, shows the proportion of properly recognized entities to the total number of entities in the human annotated data set. In other words, precision denotes the accuracy of a produced system, whereas recall denotes the system's coverage.

Table 2: Statistics of Benchmark Data-sets.

Data-set	Text Category	Number of Documents	Words per Document				Keywords per Document			
			Max	*Min*	*Mean*	*Std*	*Max*	*Min*	*Mean*	*Std*
SemEval 2010	Article	244	375	40	166	60	38	5	16	4
Inspec	Abstract	2000	502	15	124	59	40	2	14	6
SemEval 2017	Paragraph	493	355	60	169	48	45	4	17	7

We use the fuzzy matching framework for partial matching to assess the selected KE tools. This framework's justification is that even while KE approaches frequently produce the right key phrase, exact matching tests frequently produce subpar results. The following metrics are described using this framework:

$$Precision = \frac{number\ of\ partially\ matched\ keywords}{total\ amount\ of\ extracted\ keywords} \quad (1)$$

$$Recall = \frac{number\ of\ partially\ matched\ keywords}{total\ amount\ of\ assigned\ keywords} \quad (2)$$

High accuracy and recall are generally desired. As a result, the f-score, a statistic that incorporates both metrics, is extensively employed. The harmonic mean of recall and accuracy, as described below, can be used to determine the partial balanced F-score.

$$F - score = \frac{2*Precision*Recall}{Precision+Recall} \quad (3)$$

V. Experiments

Python programming language is used to implement and evaluate the selected KE tools. The experiment code, data-sets, evaluations results are freely available at GitHub repository [33]. In this section we will discuss our experimental setup and result.

A. Experimental Setup

An Intel core i7-6400 CPU @ 3.40GHz with 16GB RAM on 64-bit Windows operating system machine is used for this experiment. We employ all python-based implementation of KE tools from https://pypi.org/. All algorithms are configured in their parametric setups to generate n-grams with sizes ranging from 1 to 3. The top ten key phrases for each method are retrieved, and the key phrases that were manually allocated are then compared.

B. Result and Discussion

For evaluating the KE tools, we calculate the execution time, execution time per document, matched keywords per document, precision, recall, and f-score. Table 3 shows the evaluation result.

As shown in Table 3, KeyBERT (maxsum) achieves highest F-score, and RAKE achieves lowest average execution time per document for all three benchmark data-sets. Fig. 1 shows F-score for all selected KE tools on three benchmark data-sets.

As execution time is very important for a web application, we created a ranking by calculating the ratio of the average F-score and the average execution time per document. The highest value receives the highest ranking. Fig. 2 shows the performance score of selected KE tools in log scale. Top four algorithms are RAKE, Gensim, YAKE, and KeyBERT (mmr). All of them have execution time less than 1 second. Though KeyBERT (maxsum) shows highest f-score, it has average execution time per document more than 6 seconds. And three algorithms RAKE, YAKE, and KeyBERT (mmr) have f-score within 3% of highest f-score. Later we implement these three algorithms in San Antonio Research Partnership Portal and re-evaluated their performance.

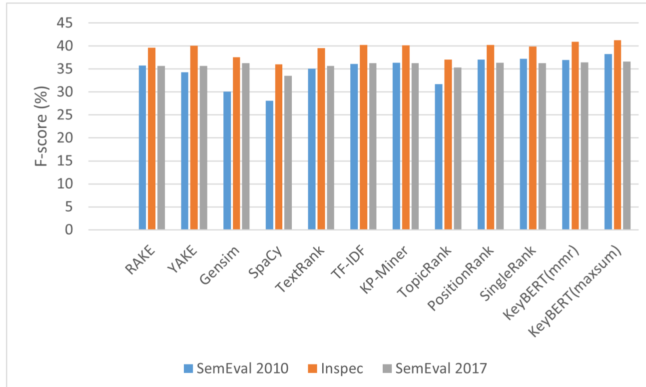

Figure 1. F-score of all selected KE tools on three benchmark data-set.

C. Integrating KE Tools with San Antonio Research Partnership Portal

The next step is to integrate the KE tools with San Antonio Research Partnership Portal to evaluate them. When the administrative user upload a new research opportunity on the portal, our program first collect all relative information and run KE tool to extract the important keywords. Those extracted keywords are then matched with the research interest of academic researchers. After the matching, the profile of matched researchers are pulled from the database. Table 4 shows the average execution time for

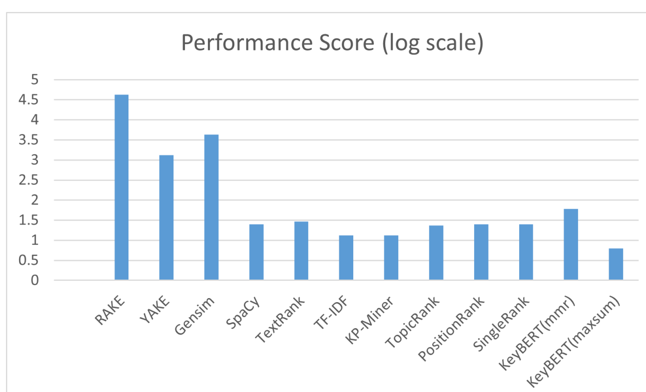

Figure 2. Performance score in log scale.

Table 3: Evaluation result of each KE tools for all three benchmark data-sets.

Algorithm	Dataset	Execution Time (sec)	Execution Time per Document (sec)	Matched Keywords per Document	Precision (%)	Recall (%)	F-score (%)	Ranking
RAKE	SemEval2010	0.22	0.0009	4.57	45.7	29.42	35.8	1
	Inspec	1.37	0.0007	4.77	47.91	33.79	39.63	
	SemEval2017	0.5	0.001	4.87	48.68	28.14	35.66	
YAKE	SemEval2010	7.1	0.0291	4.37	43.73	28.15	34.25	3
	Inspec	54.34	0.0272	4.83	48.35	34.26	40.1	
	SemEval2017	13.2	0.0268	4.87	48.72	28.16	35.69	
Gensim	SemEval2010	1.9	0.0078	3.64	41.83	23.43	30.04	2
	Inspec	16.2	0.0081	4.36	45.19	32.13	37.56	
	SemEval2017	4.19	0.0085	4.72	54.3	27.27	36.31	
SpaCy	SemEval2010	332.79	1.3639	3.58	35.82	23.06	28.06	7
	Inspec	2537.92	1.269	4.33	43.53	30.72	36.02	
	SemEval2017	619.97	1.2575	4.58	45.76	26.45	33.52	
TextRank	SemEval2010	313.65	1.2855	4.48	44.8	28.84	35.09	5
	Inspec	2576.03	1.288	4.75	48.03	33.66	39.58	
	SemEval2017	606.09	1.2294	4.86	48.64	28.12	35.64	
TF-IDF	SemEval2010	717.35	2.94	4.61	46.07	29.66	36.09	11
	Inspec	5694.68	2.8473	4.85	48.5	34.37	40.23	
	SemEval2017	1388.04	2.8155	4.95	49.53	28.63	36.29	
KP-Miner	SemEval2010	715.62	2.9329	4.64	46.39	29.87	36.34	10
	Inspec	5595.92	2.798	4.81	48.75	34.09	40.12	
	SemEval2017	1377.03	2.7932	4.95	49.49	28.61	36.26	
TopicRank	SemEval2010	387.74	1.5891	4.05	40.5	16.07	31.72	9
	Inspec	2916.21	1.4581	4.42	45.24	31.35	37.04	
	SemEval2017	724.12	1.4688	4.82	48.22	27.87	35.32	
PositionRank	SemEval2010	398.3	1.6324	4.73	47.34	30.47	37.08	8
	Inspec	2912.82	1.4564	4.82	48.92	34.17	40.24	
	SemEval2017	723.26	1.4671	4.97	49.68	28.71	36.39	
SingleRank	SemEval2010	363.32	1.489	4.75	47.54	30.61	37.24	6
	Inspec	2973.72	1.4869	4.8	48.41	33.98	39.93	
	SemEval2017	750.69	1.5227	4.95	49.51	28.62	36.27	
KeyBERT (mmr)	SemEval2010	155.48	0.6372	4.72	47.17	30.37	36.95	4
	Inspec	1090.22	0.5451	4.93	49.3	34.94	40.9	
	SemEval2017	340.7	0.6911	4.98	49.8	28.78	36.48	
KeyBERT (maxsum)	SemEval2010	1490.18	6.1073	4.88	48.81	31.42	38.23	12
	Inspec	12289.1	6.1446	4.97	49.74	35.25	41.26	
	SemEval2017	3067.85	6.2228	4.99	49.94	28.87	36.59	

the top three KE tools integrated with San Antonio Research Partnership Portal. All of them performed as expected with the web application.

VI. Conclusion

We have comparatively evaluated a set of unsupervised KE tools across different benchmark data-sets. Our experimental result shows that the RAKE algorithm is the fastest with f-score within 3% of the highest one. One limitation of our experiment might be the limited number of benchmark data-sets. Future research will take into account other data-sets in an effort to further validate the findings of this publication. We also incorporated best three KE tools with San Antonio Research Partnership Portal and

Table 4: Average Execution time of fastest KE tools on San Antonio Research Partnership Portal.

Algorithm	Average Execution Time (sec)
RAKE	0.002
YAKE	0.03
KeyBERT (mmr)	0.45

re-evaluated their performance on the web application. Our future work direction also includes the comparative evaluation of additional unsupervised KE tools and fine tuning the KE models for domain specific applications.

References

[1] Jason, L.A., Keys, C.B., Suarez-Balcazar, Y.E., Taylor, R.R. and Davis, M.I., 2004. Participatory community research: Theories and methods in action. American Psychological Association.

[2] Fawcett, S.B., 1991. Some values guiding community research and action. Journal of applied behavior analysis, 24(4), pp.621-636.

[3] Ross, L.F., Loup, A., Nelson, R.M., Botkin, J.R., Kost, R., Smith Jr, G.R. and Gehlert, S., 2010. The challenges of collaboration for academic and community partners in a research partnership: Points to consider. Journal of Empirical Research on Human Research Ethics, 5(1), pp.19-31.

[4] Minkler, M., 2005. Community-based research partnerships: challenges and opportunities. Journal of urban health, 82(2), pp.ii3-ii12.

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

[6] Firoozeh, N., Nazarenko, A., Alizon, F. and Daille, B., 2020. Keyword extraction: Issues and methods. Natural Language Engineering, 26(3), pp.259-291.

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

[8] Siddiqi, S. and Sharan, A., 2015. Keyword and keyphrase extraction techniques: a literature review. International Journal of Computer Applications, 109(2).

[9] Beliga, S., Meštrović, A. and Martinčić-Ipšić, S., 2015. An overview of graph-based keyword extraction methods and approaches. Journal of information and organizational sciences, 39(1), pp.1-20.

[10] Merrouni, Z.A., Frikh, B. and Ouhbi, B., 2016, October. Automatic keyphrase extraction: An overview of the state of the art. In 2016 4th IEEE international colloquium on information science and technology (CiSt) (pp. 306-313). IEEE.

[11] Nasar, Z., Jaffry, S.W. and Malik, M.K., 2019. Textual keyword extraction and summarization: State-of-the-art. Information Processing & Management, 56(6), p.102088.

[12] Papagiannopoulou, E. and Tsoumakas, G., 2020. A review of keyphrase extraction. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, 10(2), p.e1339.

[13] Piskorski, J., Stefanovitch, N., Jacquet, G. and Podavini, A., 2021, April. Exploring linguistically-lightweight keyword extraction techniques for indexing news articles in a multilingual set-up. In Proceedings of the EACL Hackashop on News Media Content Analysis and Automated Report Generation (pp. 35-44).

[14] Giarelis, N., Kanakaris, N. and Karacapilidis, N., 2021, June. A Comparative Assessment of State-Of-The-Art Methods for Multilingual Unsupervised Keyphrase Extraction. In IFIP International Conference on Artificial Intelligence Applications and Innovations (pp. 635-645). Springer, Cham.

[15] Kumar, A., Sharma, A., Sharma, S. and Kashyap, S., 2017, July. Performance analysis of keyword extraction algorithms assessing extractive text summarization. In 2017 International Conference on Computer, Communications and Electronics (Comptelix) (pp. 408-414). IEEE.

[16] Nadim, M., Akopian, D. and Matamoros, A., 2022. Community research partnership: A case study of San Antonio Research Partnership Portal. Electronic Imaging, 34, pp.1-6.

[17] R&D League, Retrieved from: https://www.sanantonio.gov/Innovation/R-D-League

[18] Rose, S., Engel, D., Cramer, N. and Cowley, W., 2010. Automatic keyword extraction from individual documents. Text mining: applications and theory, 1(1-20), pp.10-1002.

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

[20] Rehurek, R. and Sojka, P., 2010. Software framework for topic modelling with large corpora. In In Proceedings of the LREC 2010 workshop on new challenges for NLP frameworks.

[21] Honnibal, M. and Montani, I., 2017. spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing. To appear, 7(1), pp.411-420.

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

[23] Boudin, F., 2016, December. PKE: an open-source python-based keyphrase extraction toolkit. In Proceedings of COLING 2016, the 26th international conference on computational linguistics: system demonstrations (pp. 69-73).

[24] El-Beltagy, S.R., 2006, November. Kp-miner: A simple system for effective keyphrase extraction. In 2006 Innovations in Information Technology (pp. 1-5). IEEE.

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

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

[27] Wan, X. and Xiao, J., 2008, July. Single document keyphrase extraction using neighborhood knowledge. In AAAI (Vol. 8, pp. 855-860).

[28] Bennani-Smires, K., Musat, C., Hossmann, A., Baeriswyl, M. and Jaggi, M., 2018. Simple unsupervised keyphrase extraction using sentence embeddings. arXiv preprint arXiv:1801.04470.

[29] Grootendorst, M., 2020. Keybert: Minimal keyword extraction with bert. Internet]. Available: https://maartengr. github. io/KeyBERT/index. html.

[30] Kim, S.N., Medelyan, O., Kan, M.Y., Baldwin, T. and Pingar, L.P., SemEval-2010 Task 5: Automatic Keyphrase Extraction from Scientific.

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

[32] Augenstein, I., Das, M., Riedel, S., Vikraman, L. and McCallum, A., 2017. Semeval 2017 task 10: Scienceie-extracting keyphrases and relations from scientific publications. arXiv preprint arXiv:1704.02853.

[33] Keyword Extraction, Github repository. https://github.com/MohammadNadim/KeyWordExtraction