https://doi.org/10.2352/EI.2025.37.3.MOBMU-325

CommunityInsight AI: A Community-Centric Urban Governance Application Driven by AI

Mohammad Rokim; Department Of Urban and Regional Planning, Khulna University of Engineering & Technology, Bangladesh
Mohammad Nadim; Center for Cybersecurity Innovation, Texas A&M University - Central Texas, Texas, USA
David Akopian; Department of Electrical & Computer Engineering, The University of Texas at San Antonio, Texas, USA
Adolfo Matamoros; School of Civil & Environmental Engineering, The University of Texas at San Antonio, Texas, USA

Abstract

In the era of data-driven decision making, cities and communities are increasingly seeking ways to effectively gather insights from public feedback and comments to shape their research and development initiatives. Town hall community meetings serve as a valuable platform for citizens to express their opinions, concerns, and ideas about various aspects of city life. In this study, we aim to explore the effectiveness of different keyword extraction tools and similarity matching algorithms in matching town hall community comments with city strategic plans and current research opportunities. We employ KPMiner, TopicRank, MultipartiteRank, and KeyBERT for keyword extraction, and evaluate the performance of cosine similarity, word embedding similarity, and BERT-based similarity for matching the extracted keywords. By combining these techniques, we aim to bridge the gap between community feedback and research initiatives, enabling data-driven decision-making in urban development. Our findings will provide valuable insights for more inclusive and informed strategies, ensuring that citizen opinions and concerns are effectively incorporated into city planning and development efforts.

Index Term *- Artificial Intelligence, Community Research Partnership, Document Matching, Information Retrieval, Keyword Extraction Tool, Web Application.*

Introduction

Public feedback plays a vital role in data-driven decision-making, providing valuable insights into community needs and priorities. By incorporating public input, policymakers can tailor policies and initiatives to address specific community concerns, fostering transparency and trust in governance. This approach not only enhances responsiveness to emerging issues but also promotes equity by ensuring that decision-making reflects the diverse perspectives of the community [1]. Overall, active engagement with the public enriches the decision-making process, leading to more effective and inclusive outcomes for the community.

Keyword extraction tools are instrumental in distilling insights from public feedback and community comments, aiding decision-makers in understanding prevalent themes and concerns. These tools automatically identify and extract relevant keywords or phrases from large text datasets, allowing decision-makers to grasp sentiments, topics, and issues expressed by the community. By applying keyword extraction techniques, decision-makers can discern patterns and trends within feedback, focusing on key areas of interest to address effectively [2]. Ultimately, leveraging these tools enables more informed and targeted decision-making processes aligned with community needs and priorities, fostering transparency and accountability in governance.

Employing diverse matching algorithms enhances the correlation between public feedback keywords and city planning documents, fostering nuanced insights [3]. By integrating methods like cosine similarity and BERT-based similarity, decision-makers gain a comprehensive understanding of relevance. This approach enables prioritization and resource allocation based on semantic and contextual similarities. Comparing algorithm outcomes guides the selection of the most effective approach for future matching tasks, facilitating data-driven urban development decisions. Nadim emphasises on the AI-based applications and the security aspects of community-based research partnership platforms [4].

The rest of the paper is structured as follows: Section II presents the background about selected KE tools, different methods for converting textual data into machine readable forms, and different textual similarity matching methods. Section III includes a brief overview of San Antonio Research Partnership Portal, System design of this experiment, a detailed discussion about data used in this experiment, and a discussion about the result. Finally, the concluding remarks are outlined in Section IV.

II. Background

This section introduces the selected KE tools. The selection process is based on the comparative analysis on unsupervised KE tools [5]. The selected KE tools also represent different undying methods including statistical-based, graph-based, and machine learning-based. The section also presents the techniques to convert the textual data into machine readable formats. Finally, it briefly discusses the selected text similarity methods.

A. Keyword Extraction Tools

A keyword extraction tool is a software application or algorithm that automatically identifies and extracts important words or phrases from a given text [6]. These tools are designed to assist in information retrieval, content analysis, and other natural language processing tasks. We have selected four different KE tools for this experiment that represent different undying methods including statistical-based, graph-based, and machine learning-based.

1. KPMiner: El-Beltagy and Rafea introduced KPMiner, an unsupervised keyword extraction method utilizing a modified Tf-Idf approach with n-grams [7]. The method involves three key stages: candidate keyword selection, weight calculation, and keyword refinement, integrating statistical features to ensure robust keyword selection and balancing scores between compound and single keywords.

2. TopicRank: TopicRank utilizes a graph-based approach to extract significant keywords from documents by clustering related keywords and employing a variant of TextRank to identify the most crucial keywords within each cluster [8]. The method constructs a graph representing identified topics, where nodes represent topics and edges represent the similarity between topics. This method is beneficial for identifying important topics that may not be captured by individual keywords.

3. MultipartiteRank: MultipartiteRank constructs a multipartite graph representing both individual documents and phrases within them, utilizing a modified PageRank algorithm to consider the bipartite structure and importance of each phrase [9]. It produces representative keywords for entire document collections, incorporating positional information into edge weights, resulting in a bias toward keywords that appear earlier in the text.

4. KeyBERT: KeyBERT, a cutting-edge keyword extraction tool by Maarten Grootendorst, employs pre-trained word embedding models to identify crucial words or phrases in documents [10]. Using scikit-learn's CountVectorizer class, it generates potential keywords and computes pairwise cosine similarity scores between each keyword and the document's embedding vector, ranking them accordingly. Additionally, KeyBERT offers diversification options such as Maximal Marginal Relevance (MMR) or Max Sum Distance (MaxSum) measures.

B. Conversion to Machine Format

Word vectorization converts words or text into numerical vectors in high-dimensional space, aiming to capture semantic and syntactic relationships for machine understanding of human language. Various methods, from basic count-based approaches like Bag-of-Words to advanced models like BERT, are employed for word vectorization in natural language processing tasks.

1. Bag-of-Words: Bag-of-Words techniques like CountVectorizer and TfidfVectorizer convert text into numerical representations for natural language processing. CountVectorizer creates vectors by counting word occurrences, while TfidfVectorizer assigns weights based on term frequency and inverse document frequency, highlighting unique document characteristics. These techniques are vital for tasks like information retrieval and text classification, enabling accurate analysis by emphasizing word significance.

2. Word Embedding: Word2Vec is a popular word embedding method that represents words as dense vectors in a continuous space, capturing semantic relationships [11]. It learns these representations by training on large text datasets using CBOW or Skip-gram models. The resulting embeddings enable operations like vector arithmetic and finding nearest neighbors, proving effective in language modeling and document classification tasks.

3. Transformars: BERT, a transformative word vectorization model, utilizes a bidirectional transformer architecture to capture contextual nuances in language [12]. Pre-trained on vast text datasets, BERT learns rich representations of words for downstream tasks like sentiment analysis and text classification. Its bidirectional learning and contextual understanding have propelled it to the forefront of natural language processing, achieving state-of-the-art performance across various benchmarks.

C. Similarity Measure Algorithms

Text similarity measure algorithms are crucial in natural language processing and text analysis, facilitating tasks like plagiarism detection and recommendation systems. These methods include lexical-based approaches, which compare texts based on word occurrence and frequency, statistical methods like the Jaccard index, and semantic-based methods, which focus on capturing word meaning and context using techniques such as word embeddings or topic modeling. Each method offers valuable insights for analyzing and understanding textual data, catering to diverse purposes like document clustering and information retrieval.

1. Jaccard Similarity: Jaccard similarity quantifies the overlap between two sets by dividing the size of their intersection by the size of their union, offering a range from 0 to 1 where 0 indicates no similarity and 1 signifies complete similarity [13]. While useful for disregarding word order and frequency, it overlooks element importance and semantic nuances, making it ideal for scenarios prioritizing set presence over contextual meaning.

2. Cosine Similarity: Cosine similarity measures the similarity between two vectors in a high-dimensional space, commonly used in text analysis to compare word representations of documents or sentences, considering both direction and magnitude [14]. While effective in capturing semantic similarities, it may overlook linguistic nuances and is influenced by the quality of word representations, yet it remains valuable for various text analysis tasks.

3. Levenshtein Distance Similarity: Levenshtein distance quantifies the difference between two strings by measuring the minimum number of single-character edits needed to transform one into the other, valuable for tasks like spell checking and DNA sequence alignment [15]. While useful for structural comparisons, it does not account for semantic meaning and can be computationally intensive for lengthy strings, yet it remains essential for various text analysis applications.

4. Word Mover's Distance Similarity: Word Mover's Distance (WMD) measures text similarity by considering the semantic relationships between words, leveraging word embeddings like Word2Vec or GloVe [16]. By calculating the minimum cumulative distance that words need to travel between two documents, WMD provides a nuanced measure of similarity, particularly valuable for capturing semantic nuances across documents with varying lengths or word distributions.

III. Experiment

In this section, the testbed of this experiment, San Antonio Research Partnership Portal is briefly discussed before explaining the system designed for this experiment. The data collection process and details about the data being used in this experiment is also discussed in this section. Finally, this section presents the results and discussion of this experiment.

A. San Antonio Research Partnership Portal

The San Antonio Research Partnership Portal facilitates collaboration between academic researchers and industry in San Antonio, Texas, serving as a centralized hub for knowledge exchange and research collaboration [17]. By bridging academia and industry, the portal aims to drive innovation, economic growth, and societal impact through interdisciplinary research teams and technology commercialization [18]. Providing a user-friendly inter-

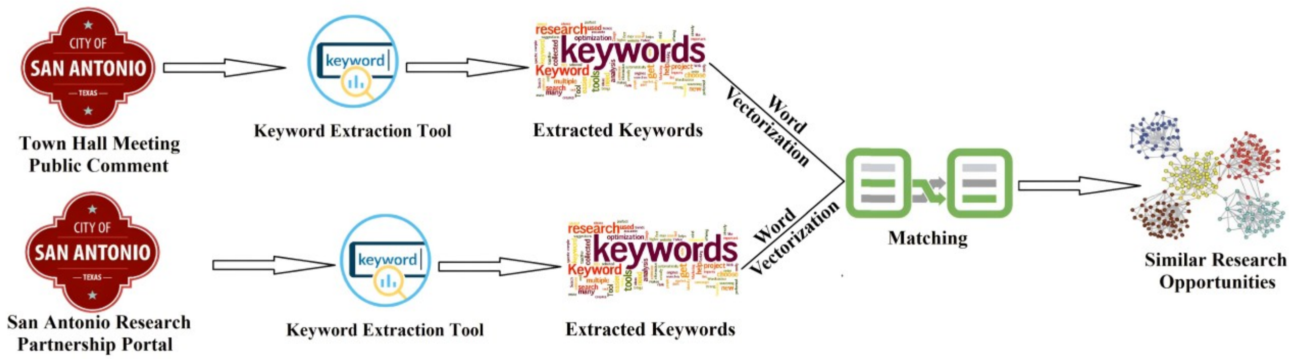

Figure 1. *Designed system to match public comment from town hall meeting with research opportunities from San Antonio Research Partnership Portal.*

face and powerful search functionalities, the portal enables researchers to showcase expertise and ongoing projects while allowing industry partners to outline research needs and explore collaborations, ultimately enhancing San Antonio's research ecosystem and competitiveness.

B. System Design

The system is designed to integrate the KE tools and matching methods to find the similarity between public comments from the town hall meetings and city documents like strategic plans or research opportunities. Figure 1 shows the design of the system developed for this experimental evaluation.

C. Experimental Data

The data of this experiment is collected from different sources and in different formats. The public feedback is collected from City of San Antonio (CoSA) Youtube channel. In 2023, they have organized eight in-person and two virtual town hall meetings. Public feedback as comment is collected from these video recordings. The strategic plans are collected by searching into the departmental website of five different city departments of CoSA. And San Antonio Research Partnership Portal contains the research opportunities collected for this experiment. To evaluate the different KE tools and matching methods, a ground truth is constructed by human evaluation. The data in strategic plan contains 5% positive class and 95% negative class. For the research opportunities, on-going opportunities from seven different city departments are collected. The data in research opportunity contains 4% positive class and 96% negative class. After the data collection, the data is cleaned using some commonly used preprocessing steps like removing punctuation, removing stop word, work tokenization, and lemmatizing words.

D. Software Tools

FuzzyWuzzy: FuzzyWuzzy, a Python library, employs the Levenshtein distance algorithm for fuzzy string matching, offering functions like *partial_ratio* to compare partial matches and *token_set_ratio* to assess similarity based on unique tokens, disregarding their order. Another function, *token_sort_ratio*, evaluates similarity considering the sorted tokens' order, ideal for comparing strings with similar words but varying arrangements or additional words.

SpaCy: SpaCy is a popular Python library for natural language processing that offers efficient and accurate tools for text similarity calculation. With its pre-trained word vectors, such as the popular *en_core_web_md* model, SpaCy can measure the similarity between two texts by comparing the similarity of their constituent words. By leveraging semantic information encoded in word embeddings, SpaCy captures both exact word matches and semantic similarities. To use SpaCy for text similarity, load the pre-trained word vectors, process the texts with SpaCy's pipeline, and then compare their similarity using the *similarity()* method.

Gensim: The Gensim library in Python offers tools for computing Word Mover's Distance (WMD) using the *WmdSimilarity* class, requiring a Word2Vec model trained on a corpus or pre-trained embeddings. Initializing a *WmdSimilarity* instance with the query document and Word2Vec model enables calculation of the semantic similarity, considering the distributional similarity of word embeddings and the transformation movement between sets of embeddings. This method is valuable for assessing semantic similarity, especially with varied document lengths and vocabulary, leveraging the *KeyedVectors* class for efficient handling of pre-trained embeddings from diverse formats like word2vec, GloVe, or FastText.

Scikit-Learn: Scikit-learn provides the *cosine_similarity* function in its pairwise module for computing cosine similarity between sample sets or documents, complemented by versatile vectorizers like CountVectorizer and TfidfVectorizer. CountVectorizer converts documents into matrices of term frequencies, while TfidfVectorizer additionally weighs word importance across the corpus using TF-IDF. These tools offer user-friendly means of converting text data into numerical representations for machine learning applications, enhancing their utility in natural language processing tasks.

Transformer: The transformer package in Python is a powerful tool for natural language processing (NLP) tasks. It is based on the transformer architecture, which has revolutionized the field of NLP. The transformer package provides a set of pre-trained models, such as BERT, GPT, and T5, that can be used for various tasks like text classification, named entity recognition, machine translation, and more. These models are capable of capturing complex linguistic patterns and contextual relationships in text. The transformer package also includes modules for tokenization, attention mechanisms, and model training, making it a comprehensive solution for building advanced NLP models.

E. Evaluation Metrics

Precision: Precision metric is used to evaluate the accuracy of a model's prediction and it is defined as the ratio of the true positive predictions to the total number of predictions made by the model.

$$Precision = \frac{Number\ of\ correct\ instances}{Number\ of\ predicted\ instances} \quad (1)$$

Recall: Recall metric is used to evaluate the effectiveness of a model in identifying all relevant instances and defined as the ratio of true positive prediction to the total number of actual instances in the data.

$$Recall = \frac{Number\ of\ correct\ instances}{Number\ of\ actual\ instances} \quad (2)$$

F-score: F-score is used to evaluate the overall performance of a model and it is the harmonic mean of precision and recall that balances the trade-off between them.

$$F-score = \left(1+\beta^2\right)\frac{precision * recall}{\beta^2 * precision + recall} \quad (3)$$

When $\beta = 1$, it is called F1-score. F1-score is a good metric to use in situations where both precision and recall are important, and where we need to balance the trade-off between the two metrics to optimize overall performance.

Accuracy: Accuracy is a commonly used metric to evaluate the performance of a classification model. It represents the ratio of correctly predicted instances to the total number of instances in the dataset. The accuracy score provides an overall measure of how well the model predicts the correct class labels.

$$Accuracy = \frac{Correctly\ predicted\ instances}{Number\ of\ total\ instances} \quad (4)$$

F. Results and Discussions

Two systems are designed to evaluate the performance of different KE tools and text similarity matching methods. The first system matches the public comments with city strategic plan and the second system matches the public comments with research opportunities.

1. Public Feedback to Strategic Planning

The first system is an innovative application that transforms the way public feedback is integrated into strategic planning processes. Leveraging advanced natural language processing, it systematically analyzes community input, extracting insights to align with existing strategic plans. By bridging the gap between public sentiment and governance, this application enables decision-makers to make informed, responsive, and community-centric decisions, driving greater transparency, inclusivity, and effectiveness in governance.

Figure 2 presents a comparative analysis of the performance of various KE and similarity measure tools in terms of their F-scores. In this figure, different KE tools such as KPMiner, TopicRank, MultipartiteRank, KeyBERT(mmr), and a baseline method without a KE tool (denoted as X) are evaluated across several similarity measures. The results indicate that KPMiner and TopicRank generally achieve the highest F-scores across most similarity measures, suggesting they are the most effective KE tools for this dataset. MultipartiteRank and KeyBERT(mmr) also perform well but show some variability depending on the specific similarity measure used. The baseline method (X), which does not utilize any KE tool, consistently exhibits lower F-scores, underscoring the importance and effectiveness of applying KE tools to improve text similarity measurements. This comprehensive analysis demonstrates that employing advanced KE tools significantly enhances the F-score of text similarity assessments, particularly in applications requiring precise text analysis and comparison.

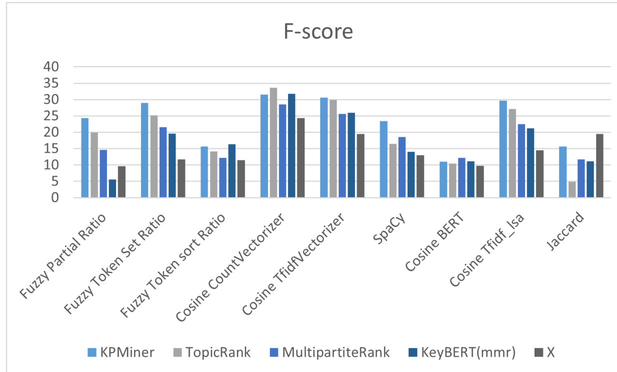

Figure 2. *F-score of system 1 experiment.*

The accuracy results depicted in the figure 3 highlight the performance of various KE and text similarity measure tools, evaluated across multiple configurations. The baseline system without any KE tool is denoted by X. Cosine CountVectorizer measure consistently shows high accuracy when paired with KE tools, indicating that even simpler vector-based methods can be effective when combined with strong KE tools. Cosine TfidfVectorizer shows competitive performance, especially when integrated with KPMiner, TopicRank, and MultipartiteRank, indicating its robustness in capturing text similarity. Fuzzy measures also perform well across various KE tools. Their accuracy is notably higher when used with KPMiner and TopicRank, although not as high as Cosine TfidfVectorizer. SpaCy, which leverage advanced NLP techniques, show moderate to high accuracy. SpaCy, when used with KPMiner and MultipartiteRank, performs notably well, reflecting the benefit of using pre-trained word vectors for semantic similarity. Cosine BERT shows the lowest accuracy among all the similarity measures across different KE tools. This suggests

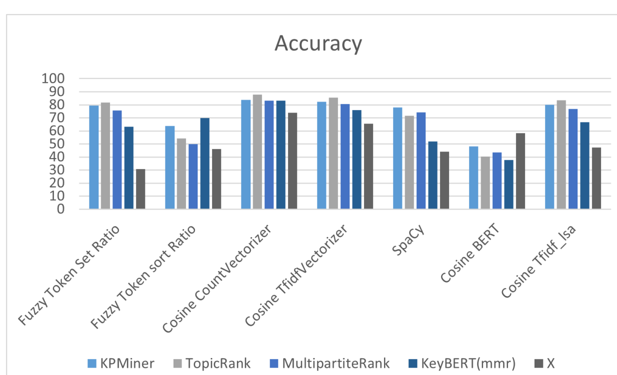

Figure 3. *Accuracy of system 1 experiment.*

that, despite BERT's powerful contextual embeddings, it might not be the most effective in this particular experiment. The complexity and the computational requirements of BERT could contribute to this lower performance. Additionally, it indicates that BERT embeddings may not align well with the public feedback and strategic plan data in this context, potentially due to domain-specific nuances that BERT did not capture effectively. Cosine Tfidf_lsa exhibits high accuracy, particularly with TopicRank and KPMiner, which indicates its effectiveness in this context. This high performance suggests that traditional methods like TF-IDF, when properly tuned, can outperform more complex models like BERT in specific scenarios. The baseline performance without any KE tool is generally lower across all similarity measures. This underscores the importance of KE tools in enhancing text similarity matching, thereby improving the overall accuracy.

2. Public Feedback to Research Opportunities

The second system is an innovative application that harnesses advanced AI algorithms to align public feedback with research opportunities, fostering community-driven research initiatives. By analyzing public input from various sources and matching it with available research opportunities, this application facilitates collaborations between academic researchers and community projects. The application empowers decision-makers to make informed choices that prioritize community needs and preferences, thus promoting inclusive and community-centric research and development.

Figure 4 displays the F-scores for various KE and similarity measure tools when matching public feedback with research opportunities. For Cosine TfidfVectorizer, KeyBERT(mmr), TopicRank, and MultipartiteRank show high F-scores, highlighting the effectiveness of TF-IDF in conjunction with these KE tools. KeyBERT(mmr) outperforms other selected KE tools when combined with Cosine CountVectorizer. Cosine Tfidf_lsa shows most constant output across all KE tools with TopicRank showing best performance. For Fuzzy Token Set Ratio, TopicRank achieves the highest F-score, indicating this method effectively captures the similarity between public feedback and research opportunities. Meanwhile, Fuzzy Token Sort Ratio measure shows relatively low F-scores across all methods, suggesting that sorting tokens is not as effective in this context. TopicRank achieves the highest F-score for SpaCy, indicating that SpaCy's pre-trained word vectors effectively capture semantic similarity when used with this KE tool. KPMiner and KeyBERT(mmr) perform moderately when using cosine similarity with BERT embedding. In gerenal, TopicRank and KeyBERT(mmr) perform best across most similarity measures, indicating their robustness in capturing text similarity between public feedback and research opportunities. KPMiner and MultipartiteRank show moderate performance for this system. Interestingly, the baseline method without a KE tool does not show worst performance in terms of F-score in this experiment.

Figure 5 presents the accuracy results of various KE and similarity measure tools in matching public feedback with research opportunities. For Fuzzy Token Sort Ratio measure, all KE tools show average accuracy except KPMiner with over 70% accuracy. While all KE tools show moderately well accuracy for Fuzzy Token Set Ratio measure with accuracy scores between 65-75%. The baseline model without any KE tool exhibits lower accuracy compared to the other KE tools, indicating the importance of using a KE tool for better accuracy. Both Cosine CountVectorizer and Cosine TfidfVectorizer measures demonstrate high accuracy across all KE tools, with scores generally exceeding 70%. TopicRank and KeyBERT(mmr) show slightly higher accuracy compared to other KE tools. The baseline also shows competitive accuracy, but slightly lower than when using KE tools. All KE tools show moderately competitive accuracy for Cosine Tfidf_lsa measure. KeyBERT(mmr) demonstrate lower accuracy compared to other tools for SpaCy. Cosine BERT measure shows average accuracy for all KE tools. The baseline accuracy is also relatively high, but still lower than KPMiner and MultipartiteRank. Overall, the results indicate that using KE tools significantly improves accuracy in matching public feedback with research opportunities. The baseline accuracy, while sometimes competitive, generally lags behind when KE tools are employed.

VI. Conclusion

In conclusion, our comparative analysis on aligning public feedback from town hall meetings with city strategic plans and research opportunities has provided valuable insights into the potential for leveraging community input for data-driven decision making. Through the use of various keyword extraction tools and similarity matching algorithms, we have demonstrated the effectiveness of these techniques in identifying relevant research opportunities based on public feedback. Our study highlights the importance of gathering insights from community comments and leveraging them to inform city planning and research initiatives.

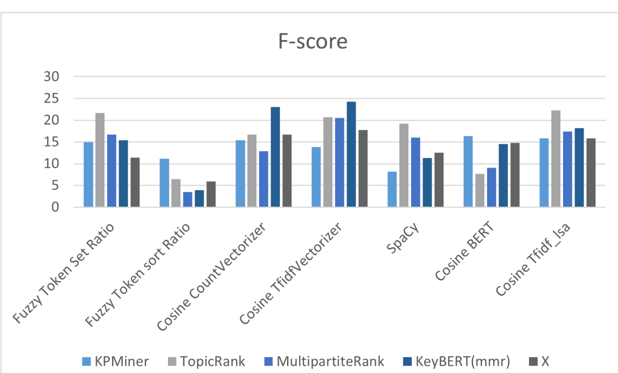

__Figure 4.__ F-score of system 2 experiment.

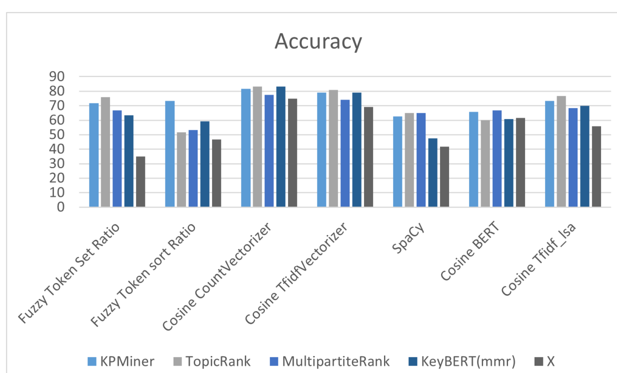

__Figure 5.__ Accuracy of system 2 experiment.

By utilizing advanced text analysis methods, we can uncover valuable patterns and connections between public feedback and strategic plans, enabling decision-makers to make informed choices that better align with the needs and aspirations of the community. This research contributes to the growing body of knowledge in the field of community engagement and data-driven governance, providing a framework for leveraging public feedback to drive positive change and promote citizen-centric decision making.

References

[1] Charnley, S., & Engelbert, B. (2005). Evaluating public participation in environmental decision-making: EPA's superfund community involvement program. Journal of environmental management, 77(3), (pp. 165-182).

[2] Nadim, M., Akopian, D., & Matamoros, A. (2023). San Antonio Research Partnership Portal: Evaluating keyword extraction tools to automate matchmaking for community research partnership. Electronic Imaging, 35, 1-6.

[3] Gomaa, W. H., & Fahmy, A. A. (2013). A survey of text similarity approaches. international journal of Computer Applications, 68(13), (pp.13-18).

[4] Nadim, M. (2023). Machine Learning for Empowering Community Applications and Security (Doctoral dissertation, The University of Texas at San Antonio).

[5] Nadim, M., Akopian, D., & Matamoros, A. (2023). A Comparative Assessment of Unsupervised Keyword Extraction Tools. IEEE Access.

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

[7] El-Beltagy, S. R., & Rafea, A. (2009). KP-Miner: A keyphrase extraction system for English and Arabic documents. Information systems, 34(1), (pp. 132-144).

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

[9] Boudin, F. (2018). Unsupervised keyphrase extraction with multipartite graphs. arXiv preprint arXiv:1803.08721.

[10] Grootendorst, M., 2020. KeyBERT: Minimal keyword extraction with BERT. Zenodo

[11] Goldberg, Y., & Levy, O. (2014). word2vec Explained: deriving Mikolov et al.'s negative-sampling word-embedding method. arXiv preprint arXiv:1402.3722.

[12] Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805.

[13] Niwattanakul, S., Singthongchai, J., Naenudorn, E., & Wanapu, S. (2013, March). Using of Jaccard coefficient for keywords similarity. In Proceedings of the international multiconference of engineers and computer scientists (Vol. 1, No. 6, pp. 380-384).

[14] Rahutomo, F., Kitasuka, T., & Aritsugi, M. (2012, October). Semantic cosine similarity. In The 7th international student conference on advanced science and technology ICAST (Vol. 4, No. 1, p. 1).

[15] Yujian, L., & Bo, L. (2007). A normalized Levenshtein distance metric. IEEE transactions on pattern analysis and machine intelligence, 29(6), (pp.1091-1095).

[16] Huang, G., Guo, C., Kusner, M. J., Sun, Y., Sha, F., & Weinberger, K. Q. (2016). Supervised word mover's distance. Advances in neural information processing systems, 29.

[17] San Antonio Research Partnership Portal. Accessed on May, 2024. https://researchpartnerships.sanantonio.gov/

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

Author Biography

Mohammad Rokim *is an undergraduate student of Urban and Regional Planning Department at Khulna University of Engineering and Technology, Bangladesh. He received prestigious Dean's Award for three successive years for his outstanding achievement in academic career. His research interest includes design development, smart city planning, data analysis, and sustainable development.*

Mohammad Nadim *is currently an Assistant Research Scientist at Texas A&M University - Central Texas. He received his Ph.D. and M.Sc. in Electrical and Computer Engineering from The University of Texas at San Antonio (UTSA) and B.Sc. degree from Bangladesh University of Engineering and Technology (BUET). His research interest includes the development of community applications, software development, machine learning, and cyber security.*

David Akopian *(Senior Member, IEEE) received the Ph.D. degree from the Tampere University of Technology, Finland. He is currently a Professor with The University of Texas at San Antonio (UTSA). Prior to joining UTSA, he was a Senior Research Engineer and a Specialist with Nokia Corporation. His current research interests include digital signal processing algorithms for communication and navigation receivers, positioning, dedicated hardware architectures, and platforms for software defined radio and communication technologies for healthcare applications. He is a fellow of U.S. National Academy of Inventors.*

Adolfo Matamoros *is a faculty (Structure) and Peter T. Flawn Distinguished Professor at The University of Texas at San Antonio (UTSA). He received his Ph.D. degree in Civil Engineering from University of Illinois at Urbana, USA in 1999. His research interest includes Design and behavior of reinforced concrete members, Fatigue repair in structural steel bridges, Earthquake engineering, and Community research partnership.*