FastLexRank: Efficient Lexical Ranking for Structuring Social Media Posts

Anonymous ACL submission

Abstract

In this paper, we present FastLexRank, a computationally efficient adaptation of the LexRank algorithm, which is an unsupervised approach to ranking texts based on graph-based centrality scoring of sentences, which we have tailored to be efficient text ranking. Addressing the computational and memory complexities of the original LexRank, FastLexRank employs a new algorithm to approximate the stationary distribution of sentence graphs, thereby enhancing efficiency while maintaining the quality of summarization. The correlation of FastLexRank's centrality scores with the original LexRank scores approaches a perfect match, and the Kendall rank correlation between ranked sequences produced by the original and the new approximation approach also reaches this high level of agreement. The paper details these algorithmic modifications and their transformative effect on the size of the data sets that can be processed, e.g., large social media corpora. Empirical results confirm FastLexRank's ability to effectively generate centrality scores for sentences in large social media corpora, underscoring its suitability for real-time analysis in various applications. We further suggest that FastLexRank can act as a ranker to identify the most central tweet, which can then be integrated with more advanced NLP technologies, such as Large Language Models, for enhanced analysis. This research contributes to Natural Language Processing by offering a scalable solution for text centrality calculation, critical for managing the ever-increasing volume of digital content.

In recent years, social media has emerged as a crucial data source for public opinion research, as highlighted by Murphy et al. (2014). Beyond traditional methodologies such as sentiment analysis and topic modeling, text summarization methods have gained prominence for distilling the essence of discussions within the social media landscape. Text summarization is an NLP task designed to condense a set of documents into a succinct representation of their gist (Abualigah et al., 2020). There are two main approaches to text summarization: abstractive and extractive (Abualigah et al., 2020). The abstractive method involves rephrasing the original text in shorter human-like narratives, abstracting away from the details, whereas the extractive method involves selecting specific sentences from the original documents that encapsulate the key ideas (Erkan and Radev, 2004).

These approaches enable researchers to capture and comprehend the vast array of conversations and viewpoints expressed online, providing insights into public sentiments and trends. Abstractive and extractive text summarization techniques have been employed in various applications, such as real-time event detection on Twitter using extractive methods (Alsaedi et al., 2021), summarizing opinionated texts (Liang et al., 2012), and identifying "event messages" within large volumes of tweets (Becker et al., 2021). Other studies have applied the abstractive approach to news reports and social media data, such as Reddit posts (Kim et al., 2018; Zhan et al., 2022), and Twitter/X (Blekanov et al., 2022; Li and Zhang, 2020).

Recent developments in LLMs have substantially enhanced the capabilities of abstractive summarization, yielding outputs of notable quality. Models such as PEGASUS (Zhang et al., 2020) and advanced LLMs like ChatGPT and Llama2 (Touvron et al., 2023) have demonstrated exceptional proficiency in condensing extensive texts into coherent and concise summaries. However, the fixed context windows of transformer-based models constrain their ability to process and distill exceedingly large text corpora. Although innovations like Gemini 1.5 Pro[1], with a context window of 1 million tokens, are now operational, the

[1]https://blog.google/technology/ai/google-gemini-next-generation-model-february-2024

computational load remains substantial for very lengthy documents due to the self-attention mechanism's complexity, denoted by $\mathcal{O}(n^2 \cdot d)$. This complexity underscores the challenges in scaling summarization tasks for extensive texts without incurring significant computational costs, explaining why previous summarization studies mostly focus on single documents/posts/threads rather than the entire relevant corpus.

Therefore, the question becomes, when facing millions of social media posts, how can we quickly identify the most important and representative posts to distill their information? Similar to the idea from Retrieval Augmented Generation (RAG) (Lewis et al., 2020), can we first target the most relevant posts and then ask LLMs to generate a summary based on this content? However, current retrieval models rely heavily on correctly specified queries to perform nearest-neighbor searches and cannot self-rank the posts based on their centrality or representativeness. This is where traditional extractive text summarization methods, like LexRank, can augment text summarization with ordering.

LexRank, introduced by Erkan and Radev in 2004 (Erkan and Radev, 2004), applies the principles of the PageRank algorithm to a graph representation of sentences, calculating the importance of each sentence within the corpus. It uses TF-IDF representations to construct a sentence-based graph. As an automatic summarization technique, LexRank has shown a remarkable ability to identify the most salient texts (with high centrality scores) within a set of documents. Despite its limitations in coherence and consistency inherent to extractive approaches, LexRank's ability to pinpoint the most representative text segments is invaluable in data mining and information retrieval. Unlike transformer-based language models, LexRank can theoretically analyze texts of unlimited length, using centrality scores to determine their typicality. This feature is particularly advantageous for distilling core information and bringing order to social media posts from large-scale textual datasets.

In this study, we reinterpret and expand the traditional scope of the LexRank algorithm beyond its original function as merely a text summarization tool, proposing its application as a comprehensive ranking algorithm. We do not treat LexRank as an alternative to LLM summarization; instead, we propose it as an augmented method that can help LLMs address context window limitations and make the summarization process more efficient.

By conceptualizing the selected social media sample as an extended document and viewing each tweet as an individual sentence, LexRank can effectively prioritize tweets based on their centrality or representativeness, highlighting how each tweet relates to the corpus as a whole. Consequently, this approach enables the identification of a subset of tweets that most accurately represents the corpus. In public opinion research, showcasing actual tweets rather than AI-generated paraphrases becomes important and informative. In this context, an extractive method—or, more aptly, a lexical ranking algorithm—offers significant contributions to leveraging social media for public opinion analysis.

Despite requiring fewer computational resources than LLMs, the immense volume of data in social media corpora continues to present a substantial challenge to the original LexRank approach, given its $\mathcal{O}(n^2)$ time and space complexity. For LexRank to remain effective and relevant, it must be adapted to handle and process large-scale data efficiently. This adaptation is essential for extracting meaningful and representative summaries from extensive and continuously expanding digital content, which is our test domain.

Accordingly, our research introduces FastLexRank, a novel approach to improving the efficiency of ranking texts using LexRank. By leveraging LexRank for its ranking capabilities, we can organize the massive volume of social media posts based on their centrality scores. These scores enable us to identify and select the most representative posts, which LLMs can then summarize into a coherent and comprehensive narrative. This improvement not only addresses the computational challenges posed by large datasets but also enhances the applicability of LLMs in generating concise summaries to voluminous data.

Algorithm 1: Streamlined LexRank Algorithm for Centrality Scores

Input: Corpus C
Output: Centrality Scores Vector $Scores$
$Sentences \leftarrow$ ExtractSentences(C);
$Embedding \leftarrow$ Embeddings$(Sentences)$;
$SimMat \leftarrow$ ComputeSim$(Embedding)$;
$TransMat \leftarrow$ Transition$(SimMat)$;
$Scores \leftarrow$ PowerMethod$(TransMat)$;
return $Scores$;

1 Limitation of Original LexRank

Algorithm 2: Power Method for Computing LexRank scores

Input: Sentence embedding matrix E, tolerance ϵ, maximum iterations max_iter
Output: Centrality scores c

// Compute the similarity matrix S
for $i = 1$ **to** n **do**
  for $j = 1$ **to** n **do**
    $S[i, j] \leftarrow \frac{E[i] \cdot E[j]}{|E[i]| \cdot |E[j]|}$;
  end
end
// Derive the transition matrix M from S
for $i = 1$ **to** n **do**
  $M[i] \leftarrow \frac{S[i]}{\sum S[i]}$;
end
$c \leftarrow$ random vector of length n;
$c \leftarrow \frac{c}{|c|_1}$;
for $i = 1$ **to** max_iter **do**
  $c_{old} \leftarrow c$;
  $c \leftarrow M^T \cdot c$;
  $c \leftarrow \frac{c}{|c|_1}$;
  if $|c - c_{old}|_1 < \epsilon$ **then**
    // convergence reached
    return c;
  end
end
return c;

The streamlined original implementation of the LexRank algorithm is outlined in Algorithm 1. It is important to note that in this study, we have not taken into account additional hyperparameters, such as the threshold for similarity scores and the damping factor; instead, we assume a fully connected corpus graph. As demonstrated, although LexRank proves effective for text summarization, its space, and computational complexities present considerable challenges when applied to vast datasets, including those with millions of posts.

A significant limitation of LexRank pertains to its memory complexity, quantified as $\mathcal{O}(n^2)$, attributable to the requirement of forming a dense stochastic matrix. Shifting the focus to computational complexity, another notable challenge emerges. The primary computational burden stems from calculating the stationary distribution of the Markov chain, which requires determining the eigenvector associated with the eigenvalue of one. In this context, the power method, as detailed in the original LexRank paper, becomes crucial. The specifics of this method are further expounded in Algorithm ??.

In conclusion, the computational complexity of the power method, integral to LexRank, is $\mathcal{O}(n^2)$. This complexity primarily arises from the matrix-vector multiplication in each iteration, involving the sentence similarity matrix of size $n \times n$. As n, the number of sentences, increases, the computational demands escalate, posing a major bottleneck in the efficiency of the LexRank algorithm when dealing with large-scale documents.

2 FastLexRank Approach

We consider embeddings where, without loss of generality, it is assumed that each embedding vector is normalized to unit length, i.e., $|E[i]| = 1$. Consequently, $S = EE^T$, where E represents the embedded vector for each document (i.e., the posts in the social media analysis scenario), and S represents the covariance matrix of embedding vectors. Let $\sigma \equiv S\mathbf{1}$ represent the row sums of S. Further, let Σ denote the diagonal matrix with diagonal entries corresponding to σ. Thus, $M = \Sigma^{-1}S$, and centrality scores are obtained by identifying the eigenvector of M^T corresponding to the eigenvalue 1. It is important to note that σ is this eigenvector, since $M^T\sigma = EE^T\Sigma^{-1}\sigma = EE^T\mathbf{1} = \sigma$.

Hence, in theory, centrality scores can be computed solely using the text embedding matrix E. Initially, calculating $E^T\mathbf{1}$ yields the column sum vector z of E. Subsequent matrix multiplication Ez then produces the centrality score, achievable within linear time complexity. This approach is not only mathematically straightforward but also computationally simpler compared to the traditional power method, offering identical results.

It is also, we feel, more interpretable. Firstly, note that $E^T\mathbf{1} = \frac{1}{n}\bar{E}$, where n is the number of sentences (or tweets in our case) in the corpus, and $\bar{E}$ is the mean embedding. Thus, $EE'\mathbf{1} \propto E\bar{E} \propto S_C(E, \bar{E})$, where S_C denotes cosine similarity. In other words, the typicality score of any given sentence embedding is simply (after rescaling) the cosine similarity of that embedding with the corpus' mean embedding — i.e., how similar that sentence is to the average sentence.

The pseudo-code demonstrating this method is presented in Algorithm 3.

Algorithm 3: FastLexRank Method

Input: Sentence Embeddings matrix E
Output: Centrality Score c_a
$z \leftarrow \text{ColumnSum}(E)$;
for $i = 1$ **to** $\mathit{length}(z)$ **do**
  $z[i] \leftarrow \frac{z[i]}{\sqrt{\sum(z[i]^2)}}$;
end
$c_a \leftarrow E \cdot z$;
return c_a;

3 Experiment

3.1 Dataset

As we discussed before, the FastLexRank algorithm proves particularly useful for ranking text centrality in large-scale corpora, such as posts from social media. To evaluate the performance of FastLexRank against the original LexRank algorithm, we conducted experiments using a Twitter corpus focused on U.S. political discussions, comprising 2004 tweets (Marchetti-Bowick and Chambers, 2012). This dataset serves as an ideal testbed to assess LexRank's efficacy in identifying key discussions or micro-blogs (tweets) within a substantial corpus.

3.2 Experimental Setting

This study contrasts our methodology, which calculates a centrality score, with the conventional power method. Specifically, we utilized the degree_centrality_scores function from the Python lexrank package to implement the traditional power method. A significant aspect of this comparison is our integration of a novel sentence embedding technique alongside the traditional **TF-IDF** representation, allowing for a comprehensive evaluation of our approach across various word representation methods. Two text representations were constructed using `TfidfVectorizer` from the scikit-learn package, and `SentenceTransformer` class with "all-MiniLM-L6-v2" model from the sentence-transformer package. This comparison elucidated the differences in speed and performance between the two algorithms.

Also, we want to note that our experiments were carried out on a high-performance computing cluster, configured with Redhat8, Intel Xeon Gold 6226R CPUs, 180 GB of RAM, and 1 NVIDIA A40 GPU, with 48 GB of VRAM.

3.3 Results

In the evaluation of LexRank as a ranking algorithm, a pivotal factor to consider is the consistency of the outcomes it yields across various implementations. This study focuses on whether different implementations of LexRank (i.e., our FastLexRank and original LexRank) yield closely aligned centrality scores and ranking sequences, a criterion for considering them as effective approximations. We present a scatter plot comparing vectors of centrality scores with the scores computed using the original LexRank method. Furthermore, we augment this analysis with Kendall's tau (τ) test, conducted using the `kendalltau` function from the SciPy package (see SciPy documentation). The computation of Kendall's τ is shown in equation 1, where P denotes the number of concordant pairs, Q the number of discordant pairs, T the number of ties in x only, and U the number of ties in y only. Pairs tied in both x and y are excluded from T and U[2]. This quantitative analysis compares the ranked sequences generated by both the original LexRank and the FastLexRank approach, providing a rigorous evaluation of their alignment. Values of τ near 1 suggest strong agreement, whereas values near -1 indicate strong disagreement.

$$\tau = \frac{P - Q}{\sqrt{(P + Q + T) \times (P + Q + U)}} \quad (1)$$

Figure 1 depicts the scatter plot between FastLexRank centrality scores and original centrality scores utilizing **SBERT** embeddings. The alignment of the two vectors of scores is striking, with the correlation coefficient reaching 1.0, indicating a perfect positive linear correlation. This close correlation is a strong indication of the reliability of the FastLexRank method. Further, we assessed the consistency of the ranking sequences derived from these two approaches. The Kendall's τ statistics is 1, indicating the rankings obtained using the FastLexRank and the original algorithm were found to be identical, i.e., there are no discordant pairs. The results suggest no difference (at

[2] https://docs.scipy.org/doc/scipy-0.15.1/reference/generated/scipy.stats.kendalltau.html

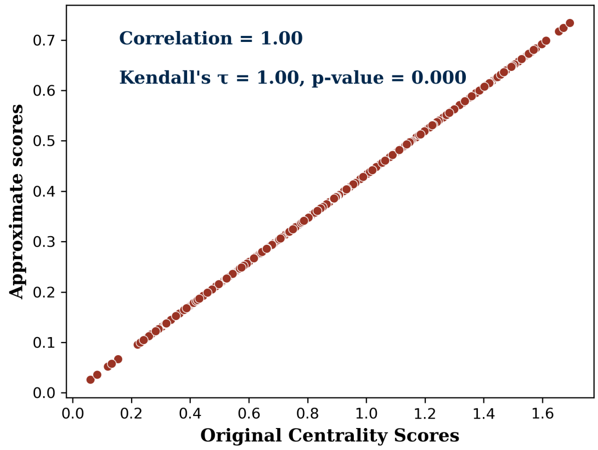

Figure 1: Comparison of FastLexRank Centrality Scores and Original Centrality Scores Using SBERT Embeddings

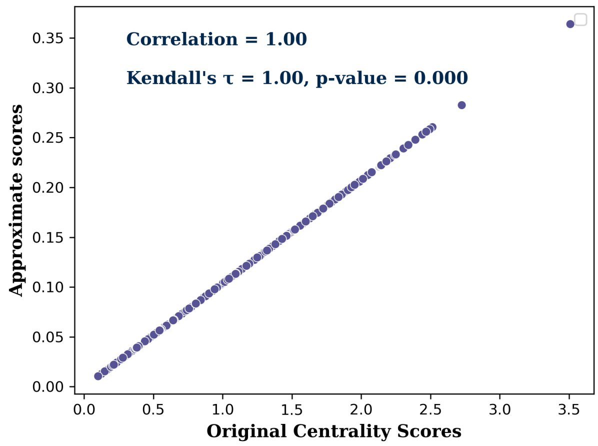

Figure 2: Comparison of FastLexRank Scores and Original Centrality Scores Using TF-IDF Embeddings

all) in terms of ranking efficacy between the two methods.

Furthermore, we conducted a similar comparison analysis using **TF-IDF** embeddings. In this study, we applied stop-word filtering and did not filter the term by minimum frequency while constructing the **TF-IDF** representation. Figure 2 presents results identical to those obtained using **SBERT** embeddings, i.e., the correlation of the scores is 1.0, and the ranking sequence is identical. The perfect correlation and identical ranking sequence indicate that the FastLexRank algorithm performs equally well in the **TF-IDF** representation. When ranking tweets, the resulting sequences are identical, whether using the power method or our approach.

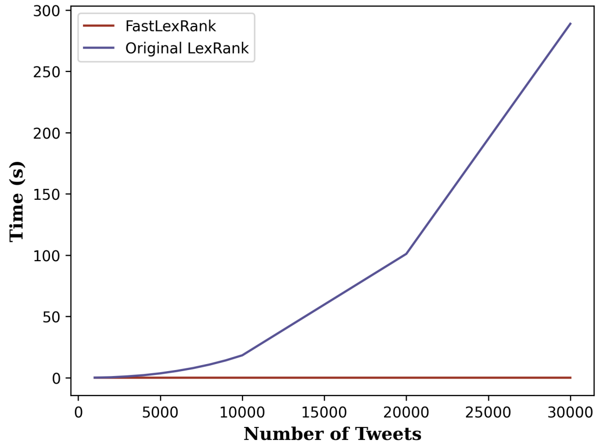

Figure 3: Time spent on Fast LexRank and Original LexRank

3.4 Assessment of Robustness

To assess the robustness of our method, we expanded our evaluation to include additional datasets, particularly selecting various Twitter corpora that contain keywords pertinent to the U.S. 2020 Census, spanning from January 1st, 2020, to February 29th, 2020. This corpus comprises 189,496 unique tweets. We randomly selected 12 subsets from the original corpus, with sizes ranging from 1,000 to 30,000 tweets. In addition to evaluating the correlation of scores and Kendall's τ, we also measured the computational time of each method, excluding the time required for embedding phrases, as it is a prerequisite step for both our method and the original LexRank algorithm. The outcomes of these evaluations are detailed in Table 1, and the computational times are illustrated in Figure 3.

As depicted in Table 1, our algorithm exhibits consistent performance across the diverse social media datasets. While our FastLexRank algorithm does not always yield identical ranking results as the original algorithms, the variations are slight, highlighting the efficacy of our approach.

Figure 3 shows that while computing time for the original LexRank algorithm increases sharply the more tweets it is processing, there is no increase in computing time for FastLexRank. This radically improved performance for FastLexRank is anticipated as the algorithm's time complexity of $\mathcal{O}(n)$ in contrast to the $\mathcal{O}(n^2)$ time complexity of the power method employed by the original LexRank for computing the stationary distribution.

Size	1,000	2,000	3,000	4,000	5,000	6,000	7,000	8,000	9,000	10,000	20,000	30,000
Kendall's τ	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
Correlation	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0

Table 1: Assessment of Robustness on different size of corpora

4 Case Study

In this section, we demonstrate how to integrate FastLexRank with other Natural Language Processing (NLP) pipelines, such as text summarization. Using the citizenship dataset, we collected a total of 189,496 unique tweets. Utilizing the OpenAI tokenizer toolkit, tiktoken, we identified 9,805,237 tokens that need processing. Summarizing this entire corpus with GPT-4o would encounter context window limitations. While breaking the corpus into digestible chunks for GPT-4o is possible, the coherence of the overall summarization would be challenging to maintain. Moreover, the cost of using the OpenAI API (for input alone) would be approximately $50.

We also observe that keyword searches within social media corpora often yield noisy posts, which may not align with the target discussion. Incorporating such noise into the LLM-generated summaries would dilute the quality of the information distilled. To highlight this, Table 2 presents the two most representative and two least representative posts, as determined by the FastLexRank similarity score. This table underscores the necessity of filtering social media corpora to focus on posts containing topical information.

Leveraging FastLexRank streamlines the summarization pipeline for the 189,496 tweets. Initially, we apply the FastLexRank algorithm to the original corpus, selecting the top 100 most representative posts. We then summarize these posts using LLMs. This approach yielded a comprehensive summary from GPT-4o, covering seven distinct topics related to Census 2020. The complete GPT-4o response is available in Appendix A.

5 Discussion

Our study reveals that the implementation of our proposed method markedly decreases the time complexity of computing centrality scores, from $\mathcal{O}(n^2)$ to $\mathcal{O}(n)$. Furthermore, by replacing the conventional approach for determining the stationary distribution with our approximation technique, we effectively reduce the overall time complexity of the LexRank algorithm from $\mathcal{O}(n^2)$ to $\mathcal{O}(n)$. In this revised framework, the majority of the computational time is allocated to constructing the sentence embedding matrix. Additionally, our method also reduces the memory requirements during the calculation process.

The FastLexRank algorithm introduces a novel mechanism for the rapid assessment of text centrality or representativeness within expansive text corpora. This utility is especially pronounced in two primary use cases. Firstly, it facilitates the expedited extraction of salient information from large-scale corpora, such as social media datasets, enabling researchers to swiftly pinpoint central tweets encapsulating the corpus's overarching narrative. This rapid identification of central messages significantly streamlines the initial phase of qualitative analysis, allowing for immediate insights into the corpus content.

Moreover, the computational efficiency of FastLexRank permits the preliminary selection of the top n central posts within a voluminous dataset. Subsequently, integrating these identified posts with Large Language Models (LLMs) yields enriched, coherent summaries. This methodology substantially enhances the capacity of social media researchers to efficiently navigate and interpret extensive datasets, thereby broadening their understanding of prevalent user discourses. Furthermore, this approach is amenable to integration with other NLP methodologies, such as topic modeling, to delineate dominant conversations within each thematic cluster, thereby augmenting the analytical granularity afforded to researchers in the domain of computational linguistics.

We also believe this approximation method can inform other ranking algorithms inspired by PageRank or similar graph-based algorithms. In summary, the rapidly evolving information landscape necessitates more efficient methods to handle the vast amount of information generated daily.

6 Limitation

A key assumption in FastLexRank's design is treating sentence graphs as fully connected. This approach simplifies the computational model, facilitating significant efficiency gains. However, it in-

Centrality	Tweet
0.87	#Census2020 is so important for our community because #WeCount. It will determine:School funding, Roads, public transportation, and infrastructure, Community Resources, Political representation . . . for the next 10years!
0.87	It's important to our future. The Census 2020 is coming. Get more information at 2020census.gov.
-0.06	@uscensusbureau Beside the verse dramas, with their lyrical interludes, on which many operas were based, there were poetic texts which were set as secular cantatas. One of the earliest was Alessandro Stradella's La Circe, in a setting for three voices that bordered on the operatic.
-0.11	Which can mislead you one key ways to know the difference is to always trust your gut feelings focus on the sensations you feel in your body at any given situation because what your soul sees your body always feels you're so census and reads energy and vibrations, cause it is

Table 2: Most and Least Representative posts Identified by FastLexRank

herently overlooks the potential benefits of thresholding—a process that could mitigate noise by disregarding connections between sentences with negligible similarity. Implementing such a threshold might filter out irrelevant connections, but it comes at the cost of reducing the model's flexibility and potentially overlooking subtle yet relevant relationships.

FastLexRank significantly advances the efficiency of the original LexRank algorithm, a foundational technique in lexical ranking that has stood for decades. By integrating the idea of self-ranking, we aim to revitalize this classical approach. Nonetheless, the reliance on LexRank's fundamental principles means that FastLexRank may inherit some of its predecessor's limitations, particularly in adapting to the nuances of modern data sets and evolving linguistic patterns.

FastLexRank represents a significant advance in modeling efficiency. However, it is crucial to recognize the evolving landscape of retrieval models, particularly those leveraging transformer-based deep learning architectures. These cutting-edge algorithms have demonstrated potential superiority in identifying the most 'representative' elements within datasets, including tweets. Although the majority of contemporary models prioritize query-based retrieval, our acknowledgment of these advances underscores an essential direction for future research. Specifically, there is a pressing need for comparative studies aimed at evaluating FastLexRank's performance and efficiency relative to these innovative models.

7 Ethical Consideration

In this study, we utilized the Twitter dataset in compliance with the data-sharing agreement, ensuring that no personally identifiable information (PII) is presented. For the four examples listed in Table 2, we removed all PII and ensured that it is impossible to trace the authors based on the provided information.

References

Laith Abualigah, Mohammad Qassem Bashabsheh, Hamzeh Alabool, and Mohammad Shehab. 2020. Text Summarization: A Brief Review. In Mohamed Abd Elaziz, Mohammed A. A. Al-qaness, Ahmed A. Ewees, and Abdelghani Dahou, editors, *Recent Advances in NLP: The Case of Arabic Language*, volume 874, pages 1–15. Springer International Publishing, Cham. Series Title: Studies in Computational Intelligence.

Nasser Alsaedi, Pete Burnap, and Omer Rana. 2021. Automatic Summarization of Real World Events Using Twitter. *Proceedings of the International AAAI Conference on Web and Social Media*, 10(1):511–514.

Hila Becker, Mor Naaman, and Luis Gravano. 2021. Selecting Quality Twitter Content for Events. *Proceedings of the International AAAI Conference on Web and Social Media*, 5(1):442–445.

Ivan S Blekanov, Nikita Tarasov, and Svetlana S Bodrunova. 2022. Transformer-based abstractive summarization for reddit and twitter: single posts vs. comment pools in three languages. *Future Internet*, 14(3):69.

G. Erkan and D. R. Radev. 2004. LexRank: Graph-based Lexical Centrality as Salience in Text Summarization. *Journal of Artificial Intelligence Research*, 22:457–479.

Byeongchang Kim, Hyunwoo Kim, and Gunhee Kim. 2018. Abstractive summarization of reddit posts with multi-level memory networks. *arXiv preprint arXiv:1811.00783*.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in Neural Information Processing Systems*, 33:9459–9474.

Quanzhi Li and Qiong Zhang. 2020. Abstractive event summarization on twitter. In *Companion Proceedings of the Web Conference 2020*, pages 22–23.

Xu Liang, Youli Qu, and Guixiang Ma. 2012. Research on Extension LexRank in Summarization for Opinionated Texts. In *2012 13th International Conference on Parallel and Distributed Computing, Applications and Technologies*, pages 517–522, Beijing, China. IEEE.

Micol Marchetti-Bowick and Nathanael Chambers. 2012. Learning for microblogs with distant supervision: political forecasting with twitter. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 603–612.

Joe Murphy, Michael W Link, Jennifer Hunter Childs, Casey Langer Tesfaye, Elizabeth Dean, Michael Stern, Josh Pasek, Jon Cohen, Mario Callegaro, Paul Harwood, et al. 2014. Social media in public opinion research: report of the aapor task force on emerging technologies in public opinion research. *American Association for Public Opinion Research*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Hongli Zhan, Tiberiu Sosea, Cornelia Caragea, and Junyi Jessy Li. 2022. Why do you feel this way? summarizing triggers of emotions in social media posts. *arXiv preprint arXiv:2210.12531*.

Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter Liu. 2020. Pegasus: Pre-training with extracted gap-sentences for abstractive summarization. In *International Conference on Machine Learning*, pages 11328–11339. PMLR.

A GPT-4o summary about most representative posts

The collection of tweets emphasizes the importance of participating in the 2020 U.S. Census. Key points highlighted across the tweets include:

1. Impact on Community Resources: Many tweets underscore that census data determines funding for critical services such as schools, roads, public transportation, healthcare, and community resources.
2. Political Representation: Accurate census data is crucial for determining political representation at various levels of government.
3. Misinformation and Participation: Some tweets address the challenges of misinformation and encourage communities to educate themselves to ensure everyone is counted.
4. Civic Duty: Participating in the census is described as a civic duty that affects the allocation of federal funding and public resources for the next decade.
5. Public Awareness Campaigns: Numerous tweets mention efforts by public officials, organizations, and communities to raise awareness and encourage census participation.
6. Inclusivity: The emphasis is on ensuring that everyone, especially hard-to-count populations, is included to achieve a fair and accurate count.
7. Educational Efforts: Various educational resources, videos, and public service announcements are shared to inform people about the census.
8. Partnerships: Tweets highlight partnerships between the U.S. Census Bureau and local organizations, schools, and governments to promote census participation.

Overall, the overarching message is to participate in the 2020 Census to ensure fair distribution of resources, accurate political representation, and better planning for community needs.