QuoteInspector: Gaining Insight about Social Media Discussions

Peizhi Wu
University of Pennsylvania
pagewu@cis.upenn.edu

Yi Zhang
AWS AI Labs
imyi@amazon.com

Wang-Chiew Tan
Meta AI
wangchiew@meta.com

Zachary G. Ives
University of Pennsylvania
zives@cis.upenn.edu

ABSTRACT

Our greatest source of insight into the real world today is via social media. Here, a major statement or quote by a public figure (world leader, politician, celebrity, scientist) can have wide-ranging impact, igniting extensive discussions and triggering reactions. It would be helpful to have *tools for monitoring, querying, and inspecting the "flow" of social discourse*. We introduce QuoteInspector, a system uniquely designed for efficient tracking and analysis of social media discussions around quotes. QuoteInspector leverages modern text embeddings and employs a clustering-based methodology for extracting topics from posts; it further integrates various NLP techniques for in-depth cluster analysis. Additionally, the system enhances the user experience by combining keyword- and relationship-based (structured) search for efficient and precise quote retrieval.

PVLDB Reference Format:
Peizhi Wu, Yi Zhang, Wang-Chiew Tan, and Zachary G. Ives. QuoteInspector: Gaining Insight about Social Media Discussions. PVLDB, 17(12): 4501 - 4504, 2024.
doi:10.14778/3685800.3685910

1 INTRODUCTION

Social media platforms provide significant, though at times overwhelming, insights into trends, viewpoints, and reactions among a large segment of the population [2]. Many efforts have been made to analyze the sentiment and semantics of posts, understand social network connections among users, and identify communities. However, we argue that one important element has not been studied to this point, which is the *provenance* of text [10]. The provenance of statements holds important yet unexplored significance, as, e.g, highly impactful *quotes from public figures* can trigger expansive discussions, clearly identifiable divisions, and strong emotion among a diverse user base that includes content creators and political figures. Sites with a tradition of long-form user posts and commentary, such as Reddit (and, recently, X) can thus provide important insights to reporters, fact-checkers, political advisors, political scientists, and others who seek to understand influence and opinion.

We propose to demonstrate QuoteInspector, which infers and reasons over text provenance within social media, to autonomously analyze and delve into the discussions surrounding quotations, on platforms such as Reddit. To our knowledge, our work is the first specifically designed for this important task. Its design and implementation posed two key challenges. First, there are no readily available off-the-shelf datasets containing quotations and their associated social media discussions, making it difficult to build models to reason about discourse within this important domain. Second, users may have different needs in tracking the "ebb and flow" of social media reactions, requiring a general design that supports a range of user intentions. Consider the following two use cases.

Example 1.1. A **political candidate** searches for quotes from public figures and their influence on social media, as inspiration in their own speeches. For example, "Ask not what your country can do for you, ask what you can do for your country?" is an extremely famous quote by John F. Kennedy during his presidential campaign in 1960. It has since been adapted by political candidates to evoke a sense of patriotism and public service.

Example 1.2. A **news analyst** wants to search for and analyze user reactions to major politician campaign events on social media for public opinion research. Often, an event triggers many quotations from different public figures. The goal of the analyst is to identify the related quotes linked to the events of interest, along with exploring the discussions that unfold across social media.

In both scenarios, the sheer volume of social media posts makes it impractical for system users to manually read through each one to comprehend the discussions surrounding a quote. Moreover, the goal is not merely to create a summary such as a word cloud, but to better understand divergent viewpoints, associated users, and trends in discussion. Recognizing this challenge, there is a critical need for sophisticated data analysis tools that empower users to efficiently and effectively digest the requisite information within a constrained time budget. Moreover, in the second scenario, users may not even know the quotes that are triggered by a given event. Hence, achieving the goal of the aforementioned example 1.2 requires a search engine that supports quote search (by event).

In response to the challenges highlighted earlier, we propose a novel system for visualizing and querying over posts and text provenance on social media, QuoteInspector. To develop the necessary techniques to analyze social media posts, relationships, and provenance, we curate a dataset by extracting quotes from public figures, drawing from a recent project [9], and supplementing it with data obtained from the sitemaps of five prominent news outlets. Then we employ API-based crawling to gather social media discussions surrounding these quotes from a leading platform, Reddit. This data collection process forms the foundation for the development and rigorous testing of QuoteInspector.

QuoteInspector features three novel components that collectively contribute to its functionality, covering a wide range of user intentions in the analysis and exploration of quotations on social media. First, our analytics tools cluster the posts for each quote using a tailored clustering algorithm (§ 3.1), followed by cluster summarization (§ 3.2) using advanced Natural Language Processing (NLP) models such as pretrained language models and text embeddings. This dual-faceted approach enables users to swiftly grasp

Proceedings of the VLDB Endowment, Vol. 17, No. 12 ISSN 2150-8097.
doi:10.14778/3685800.3685910

the prevailing opinions and reactions on social media, ensuring a comprehensive understanding without overlooking crucial information. Second, our quote search component (§ 3.3) is powered by a full-text search engine and an efficient embedding similarity search library, enabling an efficient search of quotes using a curated list of relevant search terms. This seamlessly satisfies the goal of Example 1.2. Third, the network information explorer (§ 3.4) of QuoteInspector empowers users to delve into related quotes, posts, or subreddits, fostering a dynamic exploration of content.

We will demonstrate QuoteInspector using the crawled quote and social media discussion data, which lets users interact with the demo over real data. Users might start by searching with a quote, a keyword, a topic, or an event temporal pattern (§ 3.3.1). QuoteInspector efficiently processes the query in the quote search component and presents the search results to visitors in a visual timeline illustrating activity (§ 3.3.4). To gain more insights about the social media discussions, users can drill down on a quote or a cluster of interest, to see detailed conversation threads in real-time. To understand connections, users might jump to similar quotes or traverse network visualizations.

2 DATA COLLECTION

Quote Collection. We have collected popular quotations with their annotations (speakers and dates) since 2020 from a large collection of English news articles. Specifically, for quotes in 2020, we directly use the crawling result from the QuoteBank [9] project. For quotes after 2020 (2021-2023), we first crawl the English articles that contain "say" or "said" in headlines from the sitemaps of five major news outlets (CNN, CNBC, NBC, NYTimes, USAToday), and then extract the quotations and their attributions (speakers) using the `QuoteAnnotator` of `StanfordCoreNLP`.

Social Media Data Crawling. To identify relevant conversations for each candidate quote, we initially search for conversations that either directly mention the quote and the speaker or reference the news source from which the quote was extracted. Subsequently, we include all posts within these identified conversations as the posts associated with the given quote. This approach ensures that the analysis captures the full breadth of conversation surrounding each quote. All Reddit data crawling was conducted through the Reddit data API. We only keep the quotations that trigger more than 3 discussion threads. The total number of filtered posts is $\sim 15,000$, and the average number of discussions per quote is 6.

3 SYSTEM OVERVIEW

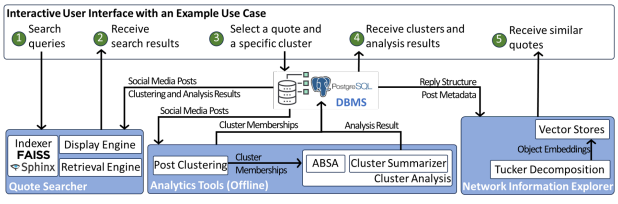

Figure 1: Overview of QuoteInspector

Figure 1 gives the design overview of QuoteInspector. It starts with a post clustering phase that clusters posts of various quotes. Once the clustering completes, QuoteInspector conducts cluster analysis, which consists of aspect-based sentiment analysis and cluster summarization. The two phases are completed offline.

After the clustering and analysis phases, QuoteInspector is ready to serve various types of search queries that correspond to different use scenarios by constructing indexes to support efficient quote search. Additionally, QuoteInspector also recommends similar quotes, users, and subreddits based on their interactions.

3.1 Post Clustering

The goal of post clustering is to extract salient topics from a collection of Reddit posts. QuoteInspector supports any embedding approaches for posts, such as TF-IDF, word2vec [7], and LLM-based embedding models [3, 4]. Since the number of topics of posts is not known *a priori*, we adapt the DP-Means algorithm from [5], which can learn the number of topics from the post collection. Another challenge of clustering Reddit posts is "off-topic posts". These posts lack relevance to meaningful topics and can adversely affect the results of clustering algorithms if included in the analysis without modification. To mitigate this issue, inspired by [8], we apply two effective techniques to remove the effects of off-topic posts. First, when assigning posts to any existing cluster, in addition to requiring their embedding distance to be close, we require the token probability distribution of the post to be close (within a threshold) to the average token probability distribution of the cluster. The second approach is removing very small clusters and merging similar clusters based on their pairwise distances.

3.2 Cluster Analysis

QuoteInspector performs two analysis tasks on top of the clusters extracted from the previous phase to enhance user experience.

3.2.1 Aspect-Based Sentiment Analysis (ABSA). The goal of ABSA is to identify the aspects (e.g., entities, topics) with the sentiment (i.e., negative, neutral, or positive). Through ABSA for each cluster, users can efficiently monitor trending people or topics and their social media interaction. It helps moderators and content creators understand what resonates with their audience. Specifically, QuoteInspector uses a BERT-based model `deberta-v3-base` that was further fine-tuned on various ABSA datasets to perform accurate ABSA for posts from each cluster.

3.2.2 Cluster Summarization. By summarizing the themes or key points of each cluster, users can quickly grasp the essence of discussions or sentiments around a quote without reading through each post. This is valuable in platforms, such as Reddit, where numerous posts are generated. It can save user's time and reduce their cognitive load. Additionally, cluster summaries allow users to easily compare different viewpoints emerging around a quote.

QuoteInspector deploys a powerful pre-trained language model, BART [6] to summarize each cluster (The model can be replaced with other LLMs such as GPT3 and XLNET). Due to the model's input length limit, we selectively choose a few "representative" posts among all posts in a cluster before running the summarization. Specifically, QuoteInspector finds out the top five posts that are nearest to the cluster center in the embedding space and concatenates them as the input to BART. Of course, the summarization

model could produce inaccurate results as it is not tailored to social media posts. It can be improved by prompt engineering or finetuning, as a future work.

3.3 Efficient Quote Search

Now, QuoteInspector is ready to serve queries that allow users to efficiently and accurately locate the quotes of their interests, and thus to explore the associated social media discussions.

3.3.1 Supported Search Queries.

• **Search by Textual Context.** This core search functionality allows users to locate quotes based on their textual context, including quote keywords, news sources, topics of the quotes, and social media discussion/reaction keywords, This multifaceted textual search capability of QuoteInspector significantly enhances the precision and relevance of the search results.

• **Search by Event Temporal Patterns.** Quotes often share temporal patterns linked to a sequence of triggering events, which we call event temporal patterns (Consider, e.g., political campaign speeches or incidents in an ongoing global conflict). In QuoteInspector, event temporal patterns are defined by three components — 1) Number of spikes: a spike occurs when there is a sudden increase followed by a decrease in the number of daily posts; 2) Periodicity (yes, no), which refers to identifying the repetitive patterns of spikes. and 3) Trend (up, down, flat, up-down): which exhibits an overall increase or decrease in event popularity during the time frame.

In addition, QuoteInspector allows users to apply more nuanced constraints to refine their search results, which is supported by a range of constraint filters: 1) Start and End Date: limit the search to quotes within a specific period; 2) Time Duration: find quotes that have sustained social media discussions over a time period; 3) Number of Posts on Peak Day and 4) Number of Clusters on Peak Day: search for quotes that generated a minimum number of posts and clusters on their peak day of discussion, respectively.

3.3.2 Storing and Indexing. QuoteInspector initially stores all the textual content (e.g., quote/post texts, extracted event/topics, and more), the timestamp information associated with quotes and posts, and cluster information (e.g., cluster membership) in a normalized form with a PostgreSQL relational database management system.

In the indexing phase, QuoteInspector first runs a series of preprocessing steps to fetch the desired data from the DBMS. Specifically, they can be retrieved from DBMS using SPJ queries (possibly with SQL aggregate functions such as `GROUP BY`). The textual results of the preprocessing steps are then indexed in an inverted index alongside the timestamps and numerical values (e.g.,post date) as features, thus providing support for matching to textual keywords or temporal patterns with additional constraints. The indexing functionality is powered by the Sphinx open-source full-text search engine. To support similar word retrieval for topics/events, we store and index the embeddings of all topics/events extracted using the popular open-source Faiss library, which provides efficient similarity search for BERT embeddings.

Note that the entire indexing phase (including Sphinx and Faiss indexing) is scheduled to execute periodically to ensure the search results provided by QuoteInspector remain up-to-date.

3.3.3 Retrieval Pipeline. When a search query asks for information spanning both Sphinx and Faiss (e.g., `mask AND topic:Covid-19`), QuoteInspector follows a two-stage retrieval pipeline. QuoteInspector first retrieves a set of initial results (i.e., quote IDs) from Faiss that meet the topic criteria. The initial results are then filtered through Sphnix to obtain the ultimate quotes that meet the search criteria in addition to the topics/events term as well.

3.3.4 Search Result Display. QuoteInspector displays quotes search results using two different views as follows.

• **Text View** (Figure 2 (a)) is designed primarily for displaying the results of textual context queries. It shows the full list of the textual context of each quote in tabular format. They are sorted based on the text similarities between the search terms and textual corpus, in the order of quote keywords, events/topics, and social media reaction keywords. Notably, for events/topics, QuoteInspector employs a similar word retrieval using BERT embeddings. This allows for effectively matching events or topics that are closely related in meaning, even if they are expressed differently.

• **Temporal View** (Figure 2 (b)) complements the text view and mainly shows the temporal patterns of social media reactions to searched quotes. After retrieving the quotes that satisfy the filters, it further clusters these quotes based on their temporal patterns. QuoteInspector employs K-Means on the time series data (numbers of posts/clusters over time) for its efficiency using the `tslearn.clustering` library and presents clustered temporal patterns in the order of cluster size along with the events (e.g., rectangles) that trigger the discussions surrounding each quote. Specifically, each row shows the daily number of posts and clusters for a quote over time, with darker blue and larger circles indicating more posts and clusters, respectively.

3.4 Exploring Network Information

QuoteInspector discovers relationships among quotes, users, and subreddits from their networked interactions. Users or subreddits with similar interests often discuss quotes from certain events/topics. Similarly, quotes from related events/topics tend to engage similar user groups of users (or subreddits). QuoteInspector leverages these patterns to recommend similar quotes, users, and subreddits, helping users efficiently find new, relevant content.

3.4.1 Triple Decomposition. QuoteInspector first learns the embedding representations for posts, users, and subreddits from their networked interactions. QuoteInspector formulates this task as a link prediction problem and adapts Tucker Decomposition [1] over triples $\{(q, u, s)\}$, with each (q, u, s) indicating user q wrote a post about the quote u in subreddit s. Tucker Decomposition is a simple and efficient model for triple decomposition.

3.4.2 Efficient Online Recommendation. After obtaining the embedding representations of objects, QuoteInspector stores and indexes them and supports efficient embedding similarity search at query time, using the `Vector stores` of Langchain.

3.4.3 Reply Network Visualization. Reddit posts follow a tree structured reply relationship within discussions. QuoteInspector visualizes this with the `react-graph-vis` library, allowing users to observe and analyze how post clusters develop and expand following the reply tree pattern. Users can click on post nodes to expand the reply network, enhancing engagement by providing an interactive way to navigate complex discussion threads.

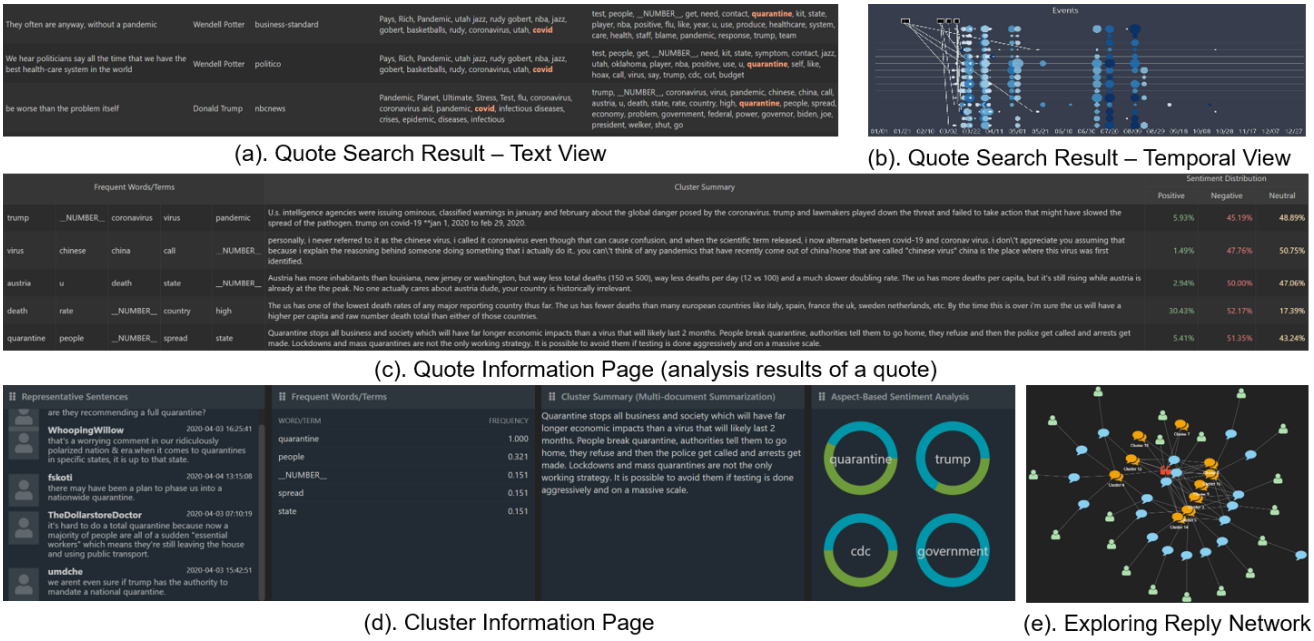

Figure 2: Demonstrating the visualization and user interface of QuoteInspector

4 DEMONSTRATION

We demonstrate QuoteInspector on real-world quotes from public figures, as well as related social media (Reddit) discussions; this data comes from QuoteBank plus a crawl we conducted in 2023 (Section 2). We will run QuoteInspector on a laptop or deploy it on an EC2 node, and users can freely interact with the system. We describe the demonstration scenario of QuoteInspector: A user engages with QuoteInspector to enhance productivity through the exploration of quotes of interest and insight gained from their social media discussions. We will illustrate our demonstration via a politician who wants to oppose lockdown/quarantine during the COVID-19 pandemic, as an example throughout this section.

- **Search the quotes:** The user first submits a quote search query to QuoteInspector by specifying textual context or event temporal patterns. For example, a politician might issue a query (`topic: covid AND reaction keywords: quarantine`) to find quotes on COVID-19 that spark social media discussions about quarantine.
- **Choose the display view:** The user specifies the display view for the quote search results, depending on the information needed.
- **View the search results:** Once the search query is submitted and processed, QuoteInspector displays search results that satisfy all search terms in text view or temporal view. Users can browse the list of searched quotes and their meta information, such as the speakers, and social media reaction keywords (Figure 2 (a), (b)).
- **Select and preview a quote:** The user finds a quote of interest and clicks the relative row to preview the summary of social media reactions. This comprises the clustering results along with frequent words, cluster summary, and sentiment distribution per cluster. For example, the politician may find the quote "we cannot let the cure be worse than the problem itself" from a public figure helpful since it generates a post cluster that focuses on quarantine (Figure 2 (c)).
- **Select and preview a cluster:** Clicking on a cluster row gives users detailed insights, including representative sentences, a summary, and ABSA results, providing a thorough understanding of the cluster's content. For instance, a politician can quickly learn from the cluster summary that the majority of social media users oppose lockdowns or quarantine measures due to perceived negative effects on businesses, societal functions, and the economy. This information can then be used to reinforce their stance in speeches (Figure 2 (d)).

ACKNOWLEDGMENTS

This work was funded in part by NSF grant III-1910108.

REFERENCES

[1] Balažević, I., Allen, C., and Hospedales, T. M. Tucker: Tensor factorization for knowledge graph completion. *EMNLP* (2019).
[2] Bär, D., Calderon, F., Lawlor, M., Lickleder er, S., Totzauer, M., and Feuerriegel, S. Analyzing social media activities at bellingcat. In *WebSci '23*.
[3] Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. *NAACL-HLT* (2019).
[4] Khattab, O., and Zaharia, M. Colbert: Efficient and effective passage search via contextualized late interaction over bert. In *SIGIR* (2020), pp. 39–48.
[5] Kulis, B., and Jordan, M. I. Revisiting k-means: New algorithms via bayesian nonparametrics. *ICML* (2012).
[6] Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., Stoyanov, V., and Zettlemoyer, L. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461* (2019).
[7] Mikolov, T., Chen, K., Corrado, G., and Dean, J. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781* (2013).
[8] Silburt, A., Subasic, A., Thompson, E., Dsilva, C., and Fares, T. Fanatic: Fast noise-aware topic clustering. In *Findings of EMNLP* (2021), pp. 650–663.
[9] Vaucher, T., Spitz, A., Catasta, M., and West, R. Quotebank: a corpus of quotations from a decade of news. In *WSDM* (2021), pp. 328–336.
[10] Zhang, Y., Ives, Z., and Roth, D. "who said it, and why?" provenance for natural language claims. In *ACL* (2020), pp. 4416–4426.