Proactive Query Expansion for Streaming Data Using External Sources

Farah Alshanik*, Amy Apon†, Yuheng Du†, Alexander Herzog†, and Ilya Safro ‡
*Department of Computer Science, Jordan University of Science and Technology, Irbid, Jordan
†School of Computing, Clemson University, Clemson, SC
‡Department of Computer and Information Sciences, University of Delaware, Newark, DE
Emails: *fmalshanik@just.edu.jo,†aapon, yuhengd, aherzog@clemson.edu ‡isafro@udel.edu

***Abstract*—Queries used to draw data from high-volume, high-velocity social media data streams, such as Twitter, typically require a set of keywords to filter the data. When topics and conversations change rapidly, initial keywords may become outdated and irrelevant, which may result in incomplete data. We propose a novel technique that improves data collection from social media streams in two ways. First, we develop a query expansion method that identifies and adds emergent keywords to the initial query, which makes the data collection a dynamic process that adapts to changes in social conversations. Second, we develop a "predictive query expansion" method that combines keywords from the streams with external data sources, which enables the construction of new queries that effectively capture emergent events that a user may not have anticipated when initiating the data collection stream. We demonstrate the effectiveness of our approach with an analysis of more than 20.5 million Twitter messages related to the 2015 Baltimore protests. We use newspaper archives as an external data source from which we collect keywords to expand the queries built from the primary stream.**

Reproducibility: https://github.com/FarahAlshanik/QE

***Index Terms*—Query Expansion; Streaming Data; Proactive Query; Information Retrieval; Emergent Events; fastText; Word Embedding;**

I. Introduction

Social media streaming data (e.g., Twitter messages or Facebook posts) have become a primary source to analyze public opinion [1], track user sentiment [2], or study emergent safety events [3] including public health crises [4], [5], natural disasters [6], [7], and political or social movements [8], [9]. Queries used to draw data from these high-volume, high-velocity, real-time sources typically require a set of words to filter the data. For example, tracking the public sentiment surrounding the COVID-19 pandemic on Twitter may rely on words such as "covid", "corona", and "lockdown" to filter out relevant messages. Such queries initiated by a static word list can be problematic because they rely on the domain expertise of users, and therefore reflect their biases or jargon, which can result in the exclusion of words required to retrieve relevant information. Static words also fail to keep up with changes in language and emergent words, resulting in incomplete data.

Information retrieval systems typically use query expansion techniques to enhance the initial user query, e.g., by adding inflected forms, cognates, and related words manually retrieved from the text [10], [11]. We propose a novel query expansion technique that addresses the challenges of analyzing data from high-volume, high-velocity social media streams. We argue that effectively filtering a data stream in an environment in which language and terms can rapidly change should not necessarily rely only on information from the stream itself. Instead, we develop a query expansion technique that integrates words from the current stream with *external data sources* (in our experiments, newspaper archives) in order to predict the occurrence of relevant words that have not appeared in the stream yet.

The idea behind our algorithm is best explained with an example. Suppose a user collects real-time Twitter messages related to an on-going protest. Filtering messages with the keyword "protest" will receive some, but not all relevant messages, because the word "protest" itself is not specific enough to find all relevant tweets.

Our solution uses archived data, such as newspaper articles, to identify keywords that were associated with previous protests, such as "looting" or "curfew". Expanding our initial query with such keywords allows us to build *proactive queries* that have the potential to detect messages that are relevant to capture the dynamics of the protest, using keywords that have not appeared in the stream yet.

We demonstrate the validity and effectiveness of our approach with an analysis of more than 20.5 million Twitter messages surrounding the 2015 Baltimore protests. We find that our proactive query expansion method outperforms alternative approaches, exhibiting particularly good results in identifying future emergent topics.

II. Related Work

Query expansion has been an active area of research, and several studies have sought an automated method to deal with the word mismatch in information retrieval [12]. In [13], the authors perform automatic query expansion using three representative techniques. The first technique is the global analysis based on the method introduced in [14]. The global analysis technique creates a thesaurus-like database with a ranked list of phrases for a given query. The method is known as the global analysis approach because the association

978-1-6654-8045-1/22/$31.00 ©2022 IEEE

database it uses considers the entire collection of documents, and the process is frequently computationally intensive.

The task in [13] is different from our task in which we use the streaming data as a primary stream to the query. This means that the thesaurus-like database used in [14] is not directly applicable. Besides, the database in this solution would require to be updated with each new tweet, which makes the method inapplicable for large-scale data [15].

The second approach introduced by [13] is the local feedback method, which overcomes the drawback of the global analysis by using the documents in the query results to generate a list of top-ranked words. The efficacy of this method crucially depends on the quality of the query result itself. The reliability of the local feedback method, therefore, remains an issue even it is less expensive to perform [15]. The third technique introduced by [13] is local context analysis, which is a combination of the global analysis approach and the local feedback approach, using the ranked query results to identify the top concepts. Based on the distance of each concept to the original query in the global thesaurus and their TF-IDF scores, the local context analysis picks new words. This method achieves better performance than using either global analysis or local feedback separately. However, it also requires a static metric to rank the documents in the query result.

For querying social streaming data, the metrics to evaluate the quality of the query results are often dynamically changing and may comprise a mixture of various sub-metrics [15]. Hence, it is not feasible to directly use the local context analysis method introduced by [13]. The query expansion method proposed in [16] uses tweet data as the query platform. In [16], the approach is similar to our query expansion work in that the authors employ a time-based indicator to deal with the data stream dynamic nature, which is similar to our method of measuring query quality using a dynamic metric. In [16], the authors use repost count and followers of posts as an indicator of a tweet's quality which changes with time. Our approach uses tweet count and hashtags information as a quality indicator of query results.

Many studies use different techniques to detect emergent events in streaming data [3], [17]–[20]. The application of topic models, such as LDA [21], has been an active area of research in informational retrieval, and several studies have sought an automated method for using topic models in query expansion [22]–[28]. Similar to this line of work, we propose a method that uses topic distributions of the targeted documents in addition to an external data source to expand the query. To the best of our knowledge, there is no existing study that uses topic modeling with external sources to expand the initial query in streaming data.

At the heart of our proposed method is the idea that previous external media coverage of social events can inform the data collection for a current ongoing event. There is existing work in information retrieval that uses external records for query expansions using logged user queries from search engines [12], which is a user centric approach, while we use records on past events. Previous work has found that social media facilitates protests [29], which has the potential to create similarities among different events. Past and present protests are also often responses to the same underlying social issue, such as protests in response to police misconduct or in response to economic inequality (e.g., the recent Occupy Wall Street movement). In general, there is belief among experts that looking at past events can be helpful when assessing current trends [30].

III. Proactive Query Expansion Method

A. System Overview

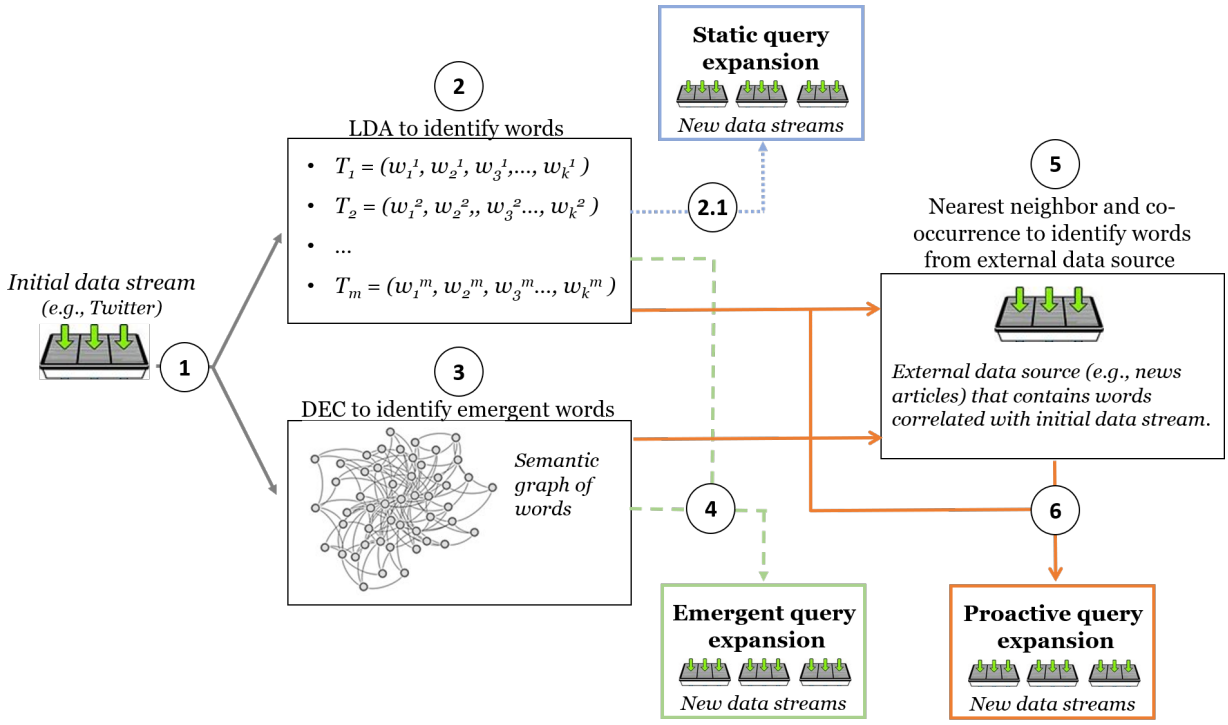
We introduce the proactive query expansion approach to detect emerging events in streaming data. Our approach utilizes external data sources to expand and enrich an initial user query with words that do not appear in the stream yet but are highly correlated with emergent words in the existing stream. By adding these words, our system can construct proactive queries that capture emergent events that were not anticipated when initiating the data collection.

In Fig. 1 we illustrate the entire proposed query expansion system scheme. First, the initial data stream (in our application below it is Twitter) is being monitored for emergent events using the Dynamic Eigenvector Centrality (DEC) algorithm introduced in [3]. Emergent events in this approach are those that reflect a change in the vocabulary found in the stream and hence provide a good starting point for constructing a new query to adapt to these changes. When an emergent event is detected, a new set of queries is triggered. In step 2, we use LDA to identify new keywords from the current stream, which results in a *static query expansion*. In step 3, we use DEC to identify emergent words in the initial stream and combine these words with LDA words in step 4 to construct an *emergent query expansion*. In step 5, we identify proactive words from an external data sources (in our application newspaper archives), using two methods that in previous work were used to identify correlated words. In the final step 6, we combine these proactive words with the LDA words and DEC words, which results in the *proactive query expansion*.

B. Query Expansion Methods

At the core of our system is the proactive query expansion method, which combines words retrieved from the primary stream with novel words extracted from the external source (archival news in our application) with the goal to anticipate words that have not appeared yet. Our method belongs to the class of approaches that employ LDA to build new queries.

A large stream of literature uses LDA for query expansion [28], [31]. LDA estimates latent topics from a given corpus, where each topic represents a ranked list of the words included in the corpus. Overall, using LDA for query expansion means that each topic discovered by LDA serves as a new query restricted to the top-ranked words in each topic. However, when applied to dynamic data, this approach ignores emergent words. A key characteristic of high-volume, high-velocity streaming data, such as Twitter, is that topics can change rapidly. Relying on LDA alone for query expansion hence



Step 1. Detect emergent events in the initial stream using DEC [3] to trigger query expansion process.
Step 2. Use LDA to identify words in the initial stream.
Step 2.1 Use LDA words to construct **static** queries.
Step 3. Use DEC to identify emergent words in the initial stream.
Step 4. Combine LDA and DEC words to construct **emergent** queries.
Step 5. Use nearest neighbor and co-occurrence applied to external data source to identify words that are correlated with LDA/DEC words extracted from the initial stream.
Step 6. Combine LDA, DEC and words from external data source to construct **proactive** queries using (vector space or co-occurrence).

Fig. 1: Systems overview.

means that the extracted words might be unable to capture emergent topics.

We use the Dynamic Eigenvector Centrality (DEC) method [3] to detect emergent words because of its ability to detect meaningful, less noisy, and more interpretable information in data streams than frequency-based measures used in [32], [33]. A natural improvement of LDA in the context of streaming data is therefore to combine words extracted with LDA with words identified as emergent based on the DEC method.

Combining LDA with DEC to construct new queries (LDA-DEC) overcomes the static nature of LDA and is arguably better suited to construct queries for streaming data where topics are rapidly changing. However, this method still only relies on words extracted from the current stream, which means that the resulting queries will potentially miss words that have not appeared in the stream yet. For this reason we propose the proactive query expansion which extends the LDA-DEC approach by adding words that, historically, were correlated with the words identified from the stream. We detect these correlated words through two different methods applied to the same external data source. In our first method, we construct a low-dimensional representation of the external data. We then use this vector space of proactive words to identify words that are close to those detected by the combined LDA-DEC method, using nearest neighbor search as proximity measure. In our second approach, we select proactive words based on their co-occurrence frequency with the LDA and DEC words.

The three query expansion methods discussed above are summarized in Table I that relates each query expansion with the method used for identifying new words.

IV. Algorithm

In this section we give a detailed description of the proactive query expansion and compare it to the two alternative approaches, the static query expansion and the emergent query expansion. For simplicity, we describe each algorithm with reference to Twitter data, but we note that our method generalizes to other social media streams of similar type. Because of space constraints, the formal definition of each

Method	Word Identification			
	LDA	DEC[1]	VS[2]	CO[3]
Static	✓	-	-	-
Emergent	✓	✓	-	-
Proactive VS	✓	✓	✓	-
Proactive CO	✓	✓	-	✓

[1]Dynamic Eigenvector Centrality, [2]External Vector Space, [3]External Co-Occurrence

TABLE I: Summary of proposed query expansion methods.

algorithm is omitted, but can be found in the extended version of this paper [34]. Table II summarizes the notations that will appear in the algorithms.

S_1	Primary tweet stream
T	A set of topics result from LDA
J	Jaccard similarity
th	Specific threshold
w	Time interval (window)
n	Number of windows
d	Number of top-ranked DEC words
s	A tweet in the primary tweet stream
t	A given topic in T
m	Number of LDA topics
k	Number of top-ranked LDA words
l	Query length
Q	Query result for all topics
q_t	Query result associated with a given topic t

TABLE II: Notation.

We process a data stream by discretizing it into time intervals (called windows), each of length w units (in our implementation we use minutes). We use the DEC metric introduced in [3] to extract the top-ranked emergent words. We trigger the query expansion if the set of emergent words in the current window indicate that an emergent event is occurring. To this end, we calculate the Jaccard similarity (J) of the top d DEC words between the current window and P previous windows. If the Jaccard similarity is less than or equal to some threshold th, we assume an emergent event is occurring and this requires a new search query with new keywords not currently captured in the query that has initiated the current stream. At this point in the stream, we execute Steps 1–6 from Figure 1 to construct the following three queries:

Static (query expansion using LDA words): LDA is used to generate m topics from the current time windows. Each topic represents a ranked list of the words included in the current window which reveals a discussion theme for the topic. Using LDA for query expansion means using each topic estimated with LDA as a new query restricted to the top-ranked words in each topic. The algorithm is described in Algorithm 1 in [34]. After generating a set of topics, we expand the query and return the results that satisfy each expanded query, such that for a given document s from the primary stream S_1 and a topic t from a set of topics T, if we can find any l LDA words in document s, then we add s into the query result (q_t). Finally, the aggregated query results Q for all topics will be returned by the end of the procedure.

Emergent (query expansion using LDA words and DEC words): We propose a query expansion method that combines words extracted with LDA with words identified as emergent based on the DEC method for a specific time window. Combining LDA with DEC to construct new queries overcomes the static nature of LDA and is arguably better suited to construct queries for streaming data where topics might rapidly change. For each topic t and a time window w, we add d top-ranked DEC words that do not appear in the k top-ranked LDA words for topic t. We used this condition to avoid redundant queries because we found that some top DEC words also appear in the top LDA words. By adding the DEC words to the topics, we will guarantee that the emerging topics are included in the query results. After adding the DEC words to each topic, we expand the query and return the results that satisfy each expanded query, such that for a given document s from the primary stream S_1, and a topic t from a set of topics T, if we can find any l LDA and DEC words in the document s, then we add document s into query result q_t. Finally, the aggregated query results Q for all topics will be returned by the end of the procedure. The algorithm is described in Algorithm 2 in [34].

Proactive Vector Space (query expansion using LDA words, DEC words, and vector space): We extend the emergent query by using external data to add words that are correlated with the words identified from the stream but potentially have not yet appeared in the initial stream. This method overcomes the limitation of the LDA-DEC method which only relies on words extracted from the current stream. Using this method allows us to capture future events or words that have not yet appeared in the stream. We used the fastText model [35] to generate a vector space V to find the words' nearest neighbor to each LDA and DEC word. The fastText model is utilized to construct an n-dimensional representation of each word in the external data called word embedding, each embedded word is represented as a vector of n dimension.

After representing each word by a vector, we use the fastText nearest neighbor method to find the closest words in space to a given word. The nearest neighbor method allows us to capture the semantic information of a given word. To find nearest neighbor words for a target word, this method computes the cosine similarity between the target word and all words in the vocabulary using the vector representation of the words.

As an example of this process, consider the top five nearest neighbor words for "curfew" (lockdown, nightfal, impos, riot, loot) and "looting" (vandal, arson, ransack, quiktrip, destruct). These words represent key moments in the primary stream, and they will be used to augmented the static query. For more investigation about the appearance of these words in the

stream, we found that "curfew" appears in the stream at 2015-04-28 05:46:05. The word "lockdown" appears after an hour and 15 minutes in a different time interval of "curfew" after two windows, which means our system can capture relevant words that have not appeared in the stream yet. The word "lockdown" appears at 2015-04-28 06:51:19. Finally, the word "impos" is the stemming of the word imposed appear in the stream after one window of "curfew" at 2015-04-28 06:02:13.

All these examples provide evidence that our system can enrich the static query with words that capture future events. It is worth mentioning that "curfew" is also correlated with the word "violence", which appears after 17 windows from "curfew" at 2015-04-28 10:09:42, The top three nearest neighbor words for "looting" show a similar trend.

Algorithm 3 in [34] describes the proposed method. The algorithm starts by adding the d top-ranked DEC words that do not appear in the top k LDA words to the topic t as explained in Algorithm 2. Then the function nearestNeighbors(V,w_t,i) is used to find the i nearest words closest to each word from topic t in the vector space V. The resulting words are then saved in a list called $nearest$. For each topic t we then attach the words that do not appear in topic t from the $nearest$ list that is saved in W_v. After adding the nearest neighbors words W_v to topic t, we expand the query and return the results such that for a given document s from the primary stream S_1, if we can find any l LDA, DEC, and nearest neighbor words in the document s , then we add document s into query result q_t. Finally, the aggregated query results Q for all topics will be returned by the end of the procedure.

Proactive Co-occurrence (query expansion using LDA and DEC words, and co-occurrence frequency): This method is the same as the previous method except that the most relevant words in the external data are identified as those with the highest frequency. This method returns a set of words that have the highest number of occurrences for a certain LDA and DEC word. We build a dictionary F that consist of bi-grams (pair of adjacent words) from the external source, then we compute the frequency for all the bi-grams in the external data to return the j highest word frequency related to each LDA and DEC word.

Returning to the previous example, consider the top five highest number of co-occurrence words for "curfew" (militari, impos, nationwid, overnight, hour) and "looting" (secur, destruct, extens, sporad, systemat). As mentioned before, the word "curfew" appears in the stream at 2015-04-28 05:46:05. The word "militari" appears after 20 minutes in a different time interval of the word "curfew" after one window. which provides evidence that our system can capture relevant words that have not appeared in the stream yet using the words co-occurrence. The word "militari" appears at 2015-04-28 06:06:53. The other word "impos" is the same word that returned using the proactive vector space, which means this word has the highest words co-occurrence and the highest cosine similarity to the word "curfew". Finally, the word "nationwid" appears in the stream after one window of the word "curfew" at 2015-04-28 06:00:13. The top three highest number of occurrences words for the word "looting" show a similar trend.

Algorithm 4 in [34] describes the the proposed method. The algorithm starts by adding the d top-ranked DEC words that do not appear in the top k LDA words to the topic t as explained in Algorithm 2. The Algorithm then identifies the j highest frequency words for each word in topic t. The Function highestFreq(F,w_t,j) is then used to attach the highest frequency words W_f to each topic t, with the resulting words saved in a list called $freq$. For each topic t, we then attach the words that do not appear in topic t from $freq$ list that is saved in W_v. After adding the highest frequency words W_v to topic t, we expand the query and return the results such that for a given document s from the primary stream S_1, if we can find any l LDA, DEC, and highest frequency words in the document s, then we add document s into query result q_t. Finally, the aggregated query results Q for all topics are returned by the end of the procedure.

V. Evaluation

In this section, we will give an overview of the evaluation of our query expansion methods. First, we describe the data that we use to detect the emergent topics and expand the query. Second, we give a detailed description of the evaluation of our query expansion methods.

A. Data Description

We use Twitter data collected for one specific public safety event: the 2015 Baltimore protests in response to the death of Baltimore resident Freddie Gray. The death of Gray in police custody caused a series of protests and violence, which led to a whole city curfew on the evening of April 28th.

We purchased archived tweets from Gnip, a company that provides access to the full archive of public Twitter data. We used broad search words to collect tweets in order to create a noisy data stream that covers tweets related to the Baltimore events as well as unrelated events. Our data set comprises 20.5 million tweets covering fifteen days from April 17th to May 3rd, 2015. Because of its noisy nature, the stream is ideal to evaluate our method's ability to detect emergent topics and expand the query.

For our external source, we selected archival news articles published one year before the Baltimore events in order to predict the occurrence of relevant words that have not appeared in the stream yet. We chose the New York Times (NYT) and CNN as our source of external data because they have a public API that can be used to crawl archived news articles. We obtained 30,456 articles from NYT and 14,145 from CNN.

B. Evaluation

We conduct two types of experiments to evaluate our methods using Twitter data from the 2015 Baltimore protests. For both experiments, we simulate real-time stream processing by constructing a primary stream from the full data. This primary stream consists of all tweets that contain the word "police", a total of 5.1 million tweets. We divide this primary stream

into 15-minute time intervals, which we call "windows". On each window, we use the DEC metric to determine if an emergent event occurred. More precisely, we calculate the Jaccard similarity (J) between the top 200 DEC words between the current window and the three previous windows. If this similarity is less than or equal to 15%, we assume that an emergent event has occurred. Using this metric, the primary stream resulted in 373 windows with emergent events out of the 1573 windows. We then used LDA to extract a set of 5 topics in the targeted time window (intervals have emergent events), which we reduced to the top 20 words.

In order to relate our results to actual events that act as ground truth, we identified three key events from timelines published by news outlets [36] to pick time intervals to use in our experiments. Therefore, in addition to the first-time interval triggered by our algorithm (time interval 16), we used the time intervals 155, 781, and 1065 based on the events that happened in Baltimore. Time interval 155 at 7:00 am, April 19 captures the tweets about the death of Freddie Gray. Time interval 781 on April 25 includes tweets about looting, violence, and protest. Time interval 1065 at 10:00 pm on April 28 captures tweets related to the Baltimore curfew. Each experiment is applied at each of these time intervals.

In terms of computational complexity, optimizing the running time of our system is not the goal of this paper, but we note that all individual components of our solution are well studied. To calculate DEC, the most computationally intensive part is calculating eigenvectors, for which highly optimized solutions exist [37]. Similarly, there are highly optimized, parallel solutions for LDA [38]. To integrate external sources, the most computationally intensive part is estimating word embeddings, which can be done before a stream is monitored. Identifying related words in these embeddings via nearest neighbor and co-occurrence requires very little resources.

C. Experiment 1: Quantity and Quality of Retrieved Data

Experiment 1 compares the performance of our proactive query expansion methods – "proactive VS" and "proactive CO" – with respect to the reference methods static and emergent using streaming data quality indicators for a certain time interval. Each method returns a set of tweets called query result (Q). The following quality indicators metrics are:

Volume (measured by tweet count): This metric finds the total number of tweets matching a specific query condition from a certain time interval to the end of the primary stream.

Relevance (measured by hashtag count): This metric finds the total number of hashtags in the tweets matching a specific query condition from a certain time interval to the end of the primary stream.

Conciseness (measured through hashtags clustering): This metric clusters the tweets matching a specific query condition from a certain time interval to the end of the stream. We cluster the tweets based on the hashtags attached to them. Hashtags are used by Twitter users to categorize their messages into meaningful topics, which is why they provide an ideal data source to identify similar tweets. We use k-means [39] to cluster the query results for each method, using the tweets returned by each method as data points and their hashtags as features. Our measure of conciseness is then the optimal number of clusters, with a lower number indicating that a query has returned a concise stream.

To find the optimal number of clusters (k), we used the "elbow method" [40], [41], which means we look for an inflection point ("elbow") in a graph that plots the average distortion score on the y-axis against the number of clusters on the x-axis. The distortion score is the sum of squared differences of each point to its assigned center. In this experiment, the distortion score is computed from k = 2 to k = 15 clusters, and we find the inflection point at k=8, which we select as the optimal number of clusters.

D. Experiment 2: Predictive Power of Retrieved Data

In experiment 2 we test the effectiveness of proactive VS and proactive CO to retrieve data for future events. We again take advantage of the fact that Twitter users attach hashtags to their tweets. We treat these hashtags as a labeled dataset, and we test whether our proactive query expansion methods are better in retrieving future hashtags than the two alternative approaches static and emergent. Applied to our data, we test how well the methods applied to one of the events we identified (time intervals above 16, 155, 781, and 1065) is able to predict hashtags that appear in one of the future time intervals, using precision as our evaluation metric. Because there is a large number of hashtags in each time interval, we picked a random set of the highest and lowest frequency hashtags.

To give an example: one of the high-frequency keywords in interval 781 is #protest. How effective are proactive VS and proactive CO in creating query expansions from time interval 155 that capture tweets that include the #protest hashtag? We answer this question by calculating precision as the count of the hashtag #protest in the stream divided by the count of the hashtag #protest from the start of the time interval 155 to the end of primary stream.

VI. Results

In this section we present the results from our two experiments based on volume, relevance, and conciseness (experiment 1) and hashtag precision (experiment 2). We have conducted each experiment on each of the four time intervals we identified above (16, 155, 781, and 1065). Because of space constraints, we here only report the results for time interval 155. Results for the remaining intervals are similar to those we report here and can be found in report [34].

A. Experiment 1

In terms of volume (i.e, tweet count), Figure 2 shows the number of tweets for the query results per five topics using the four methods at time interval 155. For all time intervals, we find that proactive VS and proactive CO significantly outperforms the emergent and static method: proactive VS and proactive CO return more tweets than the others for each topic,

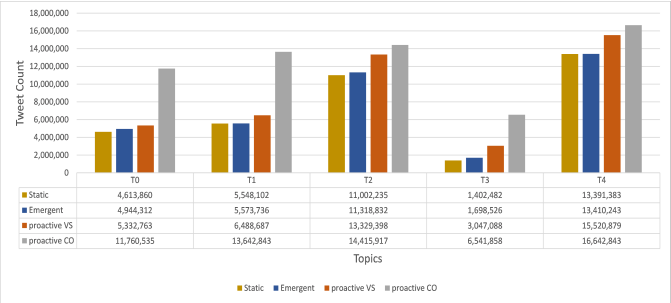


Fig. 2: Tweet count for time interval 155.

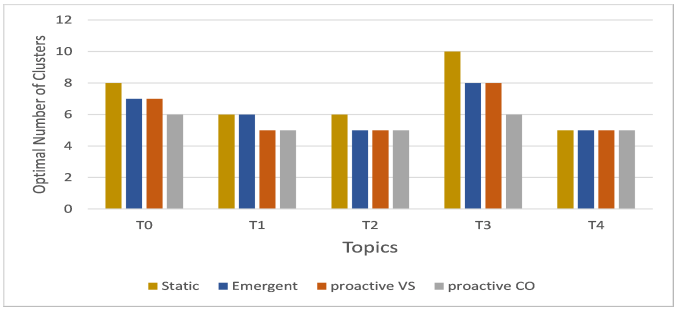


Fig. 4: Optimal number of clusters for time interval 155.

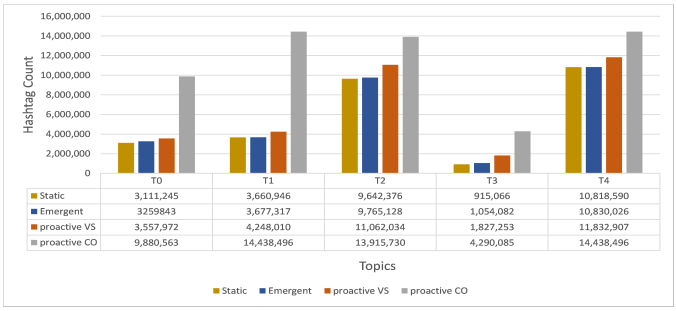


Fig. 3: Hashtag count for time interval 155.

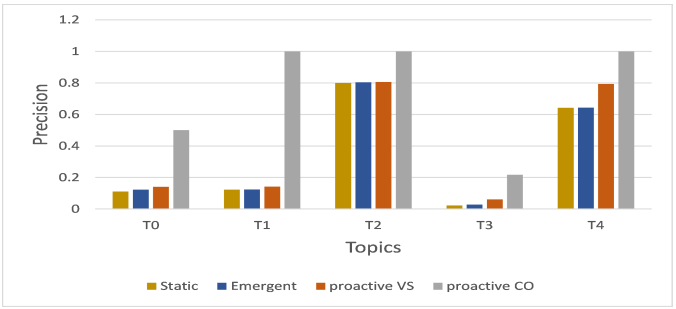


Fig. 5: Predicting events in time interval 781 from interval 155 using hashtag "protest".

which means adding new words from external sources will create more data.

In terms of relevance (i.e, hashtag count), Figure 3 shows the number of hashtags for the query result associated with each of the five topics and generated using the four methods for the time intervals. The two figures show that a higher number of hashtags is associated with proactive VS and proactive CO compared to static and emergent.

In term of conciseness (i.e, quality of hashtag clustering), Figure 4 shows the optimal number of clusters k for the query result using the k-means elbow method for each of the five topics and each of the four methods. For all topics and time intervals, proactive VS and proactive CO return more concise tweets despite the fact that they return more tweets than the other two methods, with our method outperforming the other methods by, on average, 1-2 clusters.

B. *Experiment 2*

In this experiment we test the effectiveness of our proposed methods to predict future emerging events. A precision near or equal to one indicates the effectiveness of our method to predict future events in terms of the hashtags retrieved by the method. Figure 5 summarizes precision for results for hashtag #protest selected from interval 781 when expanding the queries in interval 155 (see [34] for additional results). The precision of proactive VS and proactive CO is higher than the other methods for all topics.

VII. Conclusion

We introduced the proactive query expansion, a novel approach that suggest to dynamically adapt stream querying using keywords from the external data sources. Two major experiments were performed: (1) we compared the performance of our proposed query expansion methods: Proactive VS and Proactive CO (query expansion using LDA, DEC, and external data) with the reference methods (Static: query expansion using LDA) and Emergent (query expansion using LDA and DEC). The performance of the proposed approach is quantified by quality indicators of the streaming data which are the tweet count, hashtag count, and hashtag clustering. (2) We tested the effectiveness of our proposed methods to predict future emerging events or to predict the conversations from previous time intervals using a different set of hashtags. Our experiment performed on 20.5 million tweets (primary stream) covers fifteen days from April 17–May 3, 2015. For our external source, the external data (secondary stream), we collected news from CNN and New York Times which covers one year before the event happened (2014). Generally, the proactive query expansion methods (Proactive VS and Proactive CO) improve the performance of the information retrieval and achieve higher performance compared with Static and Emergent. Additionally, the experiments indicate that our approach can enhance the quality of the results for all the topics. Besides, our proposed methods are more concise comparing to Static and Emergent. Finally, the proposed methods play a key role in enhancing the performance of the search query, which means providing the user with more relative and concise results of interest. As possible future research direction, we suggest evaluating the proactive query expansion within domains of interest (such as health or social activity) combined with domain specific text preprocessing such as elimination of domain related stop words [42].

REFERENCES

[1] E. D'Andrea, P. Ducange, A. Bechini, A. Renda, and F. Marcelloni, "Monitoring the public opinion about the vaccination topic from tweets analysis," *Expert Systems with Applications*, vol. 116, pp. 209–226, 2019.

[2] E. Martínez-Cámara, M. T. Martín-Valdivia, L. A. Urena-López, and A. R. Montejo-Ráez, "Sentiment analysis in twitter," *Natural Language Engineering*, vol. 20, no. 1, pp. 1–28, 2014.

[3] N. Avudaiappan, A. Herzog, S. Kadam, Y. Du, J. Thatcher, and I. Safro, "Detecting and summarizing emergent events in microblogs and social media streams by dynamic centralities," in *2017 IEEE International Conference on Big Data (Big Data)*. IEEE, 2017, pp. 1627–1634.

[4] J. P. Guidry, Y. Jin, C. A. Orr, M. Messner, and S. Meganck, "Ebola on instagram and twitter: How health organizations address the health crisis in their social media engagement," *Public relations review*, vol. 43, no. 3, pp. 477–486, 2017.

[5] Y. V. Bolotova, J. Lou, and I. Safro, "Detecting and monitoring foodborne illness outbreaks: Twitter communications and the 2015 us salmonella outbreak linked to imported cucumbers," *arXiv preprint arXiv:1708.07534*, 2017.

[6] C. C. David, J. C. Ong, and E. F. T. Legara, "Tweeting supertyphoon haiyan: Evolving functions of twitter during and after a disaster event," *PloS one*, vol. 11, no. 3, p. e0150190, 2016.

[7] N. Hubig, P. Fengler, A. Züfle, R. Yang, and S. Günnemann, "Detection and prediction of natural hazards using large-scale environmental data," in *International Symposium on Spatial and Temporal Databases*. Springer, 2017, pp. 300–316.

[8] C. Vaccari, A. Valeriani, P. Barberá, R. Bonneau, J. T. Jost, J. Nagler, and J. A. Tucker, "Political expression and action on social media: Exploring the relationship between lower-and higher-threshold political activities among twitter users in italy," *Journal of Computer-Mediated Communication*, vol. 20, no. 2, pp. 221–239, 2015.

[9] K. Keib, I. Himelboim, and J.-Y. Han, "Important tweets matter: Predicting retweets in the# blacklivesmatter talk on twitter," *Computers in human behavior*, vol. 85, pp. 106–115, 2018.

[10] H. Schütze and J. O. Pedersen, "A cooccurrence-based thesaurus and two applications to information retrieval," *Information Processing & Management*, vol. 33, no. 3, pp. 307–318, 1997.

[11] J. Zhang, B. Deng, and X. Li, "Concept based query expansion using wordnet," in *2009 International e-Conference on Advanced Science and Technology*. IEEE, 2009, pp. 52–55.

[12] H. K. Azad and A. Deepak, "Query expansion techniques for information retrieval: a survey," *Information Processing & Management*, vol. 56, no. 5, pp. 1698–1735, 2019.

[13] J. Xu and W. B. Croft, "Query expansion using local and global document analysis," in *Acm sigir forum*, vol. 51, no. 2. ACM New York, NY, USA, 2017, pp. 168–175.

[14] Y. Jing and W. B. Croft, *An association thesaurus for information retrieval*. Citeseer, 1994.

[15] Y. Du, "Streaming infrastructure and natural language modeling with application to streaming big data," Ph.D. dissertation, Clemson University, 2019.

[16] K. Massoudi, M. Tsagkias, M. De Rijke, and W. Weerkamp, "Incorporating query expansion and quality indicators in searching microblog posts," in *European Conference on Information Retrieval*. Springer, 2011, pp. 362–367.

[17] Z. Saeed, R. A. Abbasi, I. Razzak, O. Maqbool, A. Sadaf, and G. Xu, "Enhanced heartbeat graph for emerging event detection on twitter using time series networks," *Expert Systems with Applications*, vol. 136, pp. 115–132, 2019.

[18] X. Chen, X. Zhou, T. Sellis, and X. Li, "Social event detection with retweeting behavior correlation," *Expert Systems with Applications*, vol. 114, pp. 516–523, 2018.

[19] M. Adedoyin-Olowe, M. M. Gaber, C. M. Dancausa, F. Stahl, and J. B. Gomes, "A rule dynamics approach to event detection in twitter with its application to sports and politics," *Expert Systems with Applications*, vol. 55, pp. 351–360, 2016.

[20] Q. Chen, W. Wang, K. Huang, S. De, and F. Coenen, "Multi-modal generative adversarial networks for traffic event detection in smart cities," *Expert Systems with Applications*, p. 114939, 2021.

[21] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," *the Journal of machine Learning research*, vol. 3, pp. 993–1022, 2003.

[22] X. Wei and W. B. Croft, "Lda-based document models for ad-hoc retrieval," in *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, 2006, pp. 178–185.

[23] J. Xu and W. B. Croft, "Cluster-based language models for distributed retrieval," in *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, 1999, pp. 254–261.

[24] W. Li and A. McCallum, "Pachinko allocation: Dag-structured mixture models of topic correlations," in *Proceedings of the 23rd international conference on Machine learning*, 2006, pp. 577–584.

[25] X. Yi and J. Allan, "Evaluating topic models for information retrieval," in *Proceedings of the 17th ACM conference on Information and knowledge management*, 2008, pp. 1431–1432.

[26] D. Zhou and V. Wade, "Latent document re-ranking," in *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, 2009, pp. 1571–1580.

[27] Z. Ye, J. X. Huang, and H. Lin, "Finding a good query-related topic for boosting pseudo-relevance feedback," *Journal of the American Society for Information Science and Technology*, vol. 62, no. 4, pp. 748–760, 2011.

[28] K. Christidis, G. Mentzas, and D. Apostolou, "Using latent topics to enhance search and recommendation in enterprise social software," *Expert Systems with Applications*, vol. 39, no. 10, pp. 9297–9307, 2012.

[29] E. Zhuravskaya, M. Petrova, and R. Enikolopov, "Political effects of the internet and social media," *Annual Review of Economics*, vol. 12, pp. 415–438, 2020.

[30] E. A. Vogels, L. Rainie, and J. Anderson, "Experts predict more digital innovation by 2030 aimed at enhancing democracy." *Pew Research Center*, 2020.

[31] M. Serizawa and I. Kobayashi, "A study on query expansion based on topic distributions of retrieved documents," in *International Conference on Intelligent Text Processing and Computational Linguistics*. Springer, 2013, pp. 369–379.

[32] A. Saha and V. Sindhwani, "Learning evolving and emerging topics in social media: a dynamic nmf approach with temporal regularization," in *Proceedings of the fifth ACM international conference on Web search and data mining*, 2012, pp. 693–702.

[33] J. Benhardus and J. Kalita, "Streaming trend detection in twitter," *International Journal of Web Based Communities*, vol. 9, no. 1, pp. 122–139, 2013.

[34] F. Alshanik, A. Apon, Y. Du, A. Herzog, and I. Safro, "Proactive query expansion for streaming data using external source," *arXiv preprint arXiv:2201.06592*, 2022.

[35] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov, "Enriching word vectors with subword information," *Transactions of the Association for Computational Linguistics*, vol. 5, pp. 135–146, 2017.

[36] E. Ortiz. (2015) Freddie Gray: From Baltimore Arrest to Protests, a Timeline of the Case. NBC News. [Online]. Available: https://www.nbcnews.com/storyline/baltimore-unrest/timeline-freddie-gray-case-arrest-protests-n351156

[37] O. E. Livne and A. Brandt, "Lean algebraic multigrid (LAMG): Fast graph laplacian linear solver," *SIAM Journal on Scientific Computing*, vol. 34, no. 4, pp. B499–B522, 2012.

[38] Z. Liu, Y. Zhang, E. Y. Chang, and M. Sun, "Plda+ parallel latent dirichlet allocation with data placement and pipeline processing," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 2, no. 3, pp. 1–18, 2011.

[39] J. MacQueen *et al.*, "Some methods for classification and analysis of multivariate observations," in *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, vol. 1, no. 14. Oakland, CA, USA, 1967, pp. 281–297.

[40] B. Bengfort, R. Bilbro, N. Danielsen, L. Gray, K. McIntyre, P. Roman, Z. Poh *et al.*, *Yellowbrick*, 2018. [Online]. Available: http://www.scikit-yb.org/en/latest/

[41] S. Susan and J. Malhotra, "Learning interpretable hidden state structures for handwritten numeral recognition," in *2020 4th International Conference on Computational Intelligence and Networks (CINE)*. IEEE, 2020, pp. 1–6.

[42] F. Alshanik, A. Apon, A. Herzog, I. Safro, and J. Sybrandt, "Accelerating text mining using domain-specific stop word lists," in *2020 IEEE International Conference on Big Data (Big Data)*. IEEE, 2020, pp. 2639–2648.