Tweelink : Linking Twitter Hashtags to News Articles

Amisha Aggarwal
amisha19016@iiitd.ac.in

Aryan Behal
aryan19026@iiitd.ac.in

Harman Singh
harman19042@iiitd.ac.in

Mayank Gupta
mayank19059@iiitd.ac.in

Yash Bhargava
yash19289@iiitd.ac.in

Yash Tanwar
yash19130@iiitd.ac.in

1. INTRODUCTION

1.1 Motivation and Problem Statement

Twitter is an extremely popular microblogging and social networking site launched in 2006. Today it boasts of hundreds of millions of monthly active users. Users can post, like, tweet and retweet 280 characters at a time. No other social platform comes close to capturing the pulse of the world in real time. Twitter users comprise people from all walks of life. Equally rich is the diversity of topics tweeted by users every second of the day. While a tweet is representative of a person's thought process, a hashtag is the culmination of collective thought of the populace. While the tweets shape the trend of a hashtag, the hashtag shapes the conversation as well, making a self feeding loop.

For better or for worse, Twitter plays a key role in shaping the conversation on topics that affect everyone's lives. From coronavirus to anti-vaccination protests to the Russia-Ukraine crisis, Twitter makes its presence felt everywhere. **Our project aims to capture the sentiment and context of a hashtag and link it to relevant news articles.** The context behind a hashtag is clouded by its informal structure. A hashtag is generally an amalgamation or portmanteau of several words making it difficult to decipher. This will help the uninformed wade through countless tweets to understand the context of the trend while combating misinformation at the same time.

1.2 Novelty

The novelty of our project lies in its uniqueness and ability to retrieve information out of a hashtag.To the best of our knowledge, there isn't any existing research or technology attempting to link hashtags to news articles. Some existing papers have tried to use the tweets containing URLs, but we are not dependent only on URLs within the tweet. Our novelty lies in the fact that we consider external news articles explaining the trending news/controversy. We face the challenge of working with a corpus of extremely short texts on which regular NLP techniques may yield poor results. The text itself may be informal, multi-lingual and full of grammatical errors.

1.3 Dataset Description

We achieve the same by building a dataset of tweets and news articles. We plan to use Twitter API v2 to track a hashtag across the most relevant tweets every hour for 72 hours. Around 30 hashtags will be collected everyday. News articles will be collected by web scraping. For a particular hashtag, after understanding its context through tweets, we plan to collect around 10 news articles within the time frame of its occurrence which consist of both relevant and less-relevant content. The data collection process will span 2-3 weeks.

1.4 Proposed Solution

We ask the user to input a hashtag, time and location. Based on the input, we retrieve relevant tweets from our dataset and create a feature vector which is used to retrieve relevant articles in a ranked order. The workflow of our system can be seen in Figure 1. The ranking is determined as a function of the hashtag, text in relevant tweets, location of tweets, location of articles and difference in time input by the user, time of tweets and date of publication of articles. Time is an important factor in the retrieval of relevant results, as illustrated in Figure 2. We will provide a web interface to facilitate the process or provide a browser extension, which will be decided as the project proceeds.

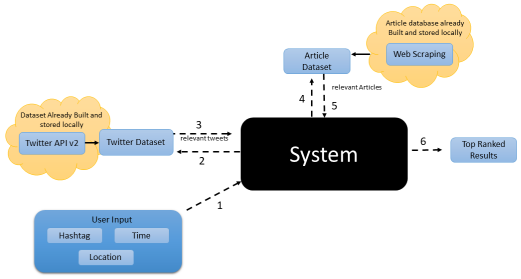

Figure 1: Workflow of our system

2. LITERATURE REVIEW

2.1 Keyword Extraction and Semantic Similarity

Bordoloi and Biswas [1] proposed a modified graph based approach KWC (Keywords from Collective Weights) for automatic keyword extraction from a twitter corpus using frequency, centrality, position, strength of the neighbours and other influencing

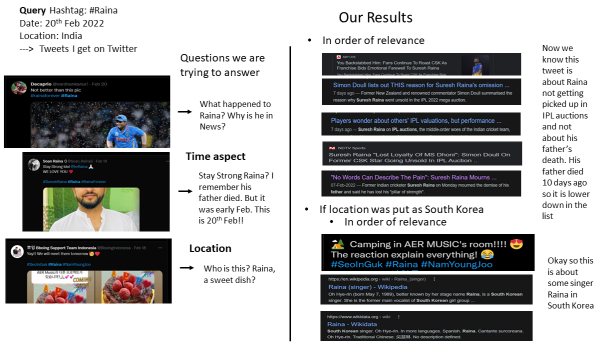

Figure 2: An example scenario of our problem statement

graph measures. Their method was based on node-edge rank centrality with node weight depending on various features. The control flow of their approach followed data preprocessing, textual graph representation, collective node-weight assignment, and keyword extraction, resulting in top keywords as the output. For the textual graph representation part, they represented each token as a vertex and if two tokens co-occurred within the same window (i.e., a tweet), then an edge was created between them. The generated graph revolved around a single topic and thus held many common tokens that were associated with more than one tweet. The edges were weighted according to co-occurrence frequency. The nodes were weighted based on important graph measures like term frequency, selectivity centrality, etc., which made their algorithm unique and more powerful. Finally, keyword extraction was performed using NE-Rank and degree centrality and best n keywords were given as output. Their novel approach outperformed several existing graph based methods. Since our project revolves around similar lines of extracting keywords for a given set of tweets for a hashtag, their superior method might come handy in our methodology.

Mihalcea et al. [2] suggested a method for measuring the semantic similarity of texts by using the information that can be drawn from the similarity of the words. Their main focus was to find the semantic similarity instead of lexical similarity. In addition to the similarity of words, they also took into account the specificity of words, which was determined using the inverse document frequency (idf). Starting with each of the two text segments, and for each word, they determined the most similar word in the other text segment, according to the PMI-IR similarity measure. Next, they combined the word similarities and their corresponding specificity, and determined the semantic similarity of the two texts. They achieved an accuracy of 70.3%, representing a significant 13.8% error rate reduction with respect to the traditional vector-based cosine similarity baseline. Our project also aims at finding the semantic similarity between the tweets and the articles. So, this technique of measuring the similarity using the PMI-IR method will be useful for our work too.

2.2 Similar Work

Uta Losch et al[3] tried to map microblogs to Encyclopedia articles. The idea of their method was to input a search query (a hashtag) and output an RDF document containing the most recent messages that matched the query, the authors of these messages and the most relevant wikipedia entities for this result. They extracted the 100 most recently published tweets using Twitter Search API along with author data, geo-location, publishing date and the URLs posted in the messages. All text data of the tweets and articles hyperlinked by the URLs was put in Wikifier for obtaining the content annotations. This tool took a text as input to return a set of DBpedia entities which were relevant for the analysed text in RDF format. Our project also runs along similar lines except for the fact that we are suggesting articles and also ranking them based on relevance. For making our tweet database from hashtags, we have followed a similar approach as they have used by using Twitter Search API and combining text to search for relevant articles. In contrast to their work, we plan to use our own methods rather than wikifier api for searching relevant articles.

Krestel et al.[4] trained various models to suggest relevant tweets for articles and evaluated their results on the basis of user study. To identify similar content tweets they employed language models (to find word overlap) and topic models (to find concept overlap). To combine the resulting content similarity scores with other features, such as recency or popularity of a tweet, they used logistic regression as well as boosting. For language modeling, they used a document likelihood model to compute the probability that a tweet is generated from a news article instead of a query likelihood model (which assumes that queries are short and documents are long). They used Dirichlet smoothing to smoothen their language model and geometric mean for normalizing tweet lengths. For relevance, they used the document likelihood and latent Dirichlet allocation to find the most relevant tweets. They used logistic regression apart from above techniques to mark a tweet as relevant or irrelevant which was trained, considering 16 additional features, such as publication time, length, follower count, etc and Adaboost and decision stumps to identify less similar tweets. To ensure unique results in the final ranking, they ignored tweets that have a high word overlap with tweets that were already in the set of recommended tweets. In their set-up, the topic model outperformed language model. Adding more features increased the accuracy of the model. In our model, we employ similar techniques of language model and topic model to check for word and concept similarity. For normalizing tweets lengths, we plan to use geometric mean. We also plan to incorporate Adaboost to identify less similar articles (along with latent Dirichlet allocation).

Ahuja et al.[5] have made a framework which downloads a list of news-related relevant tweets from twitter, extracts URLs associated with those tweets and infers the significance of those URLs in Twitter. They collected tweets using the REST and Streaming API of Twitter, and then extracted URLs from these tweets. Further, the tweets have been ranked based on relevancy. They used a threshold of 15% similarity between headline and the tweet, thus finding news-related relevant tweets and the corresponding news topics. We can use this work for extracting URLs from tweets and then using the corresponding news articles. But we are not dependent only on the URLs. We are also using the tweets that don't have URLs, and are mapping them to relevant news articles from the articles database.

2.3 Ranking Evaluation

In an ideal scenario, our system would link the hashtag directly to the most relevant article present on that particular topic. However, there could be multiple articles that are closely related with only slight variations amongst them. Corso et. al.[6] proposed a ranking algorithm that takes into account a number of desirable properties of each piece of news and subsequently assigns them a

rank based on these properties. They evaluated the consistency of their algorithm based on its behaviour for two important limiting cases, the mean rank of independent news articles should be independent of the time and size of observation window and two sources, where one acts as a "mirror" source, should have a similar rank. The authors introduced two classes of algorithms: Non-Time-aware ranking algorithms and Time-Aware ranking algorithms. Non-Time-aware ranking algorithms dealt with a static dataset of news articles rather than a continuous stream of news flow. Both algorithms, however, violated the limiting cases and coupled with the loss of temporal information associated with the fixed time-window scheme, Time-aware ranking algorithms were made necessary. According to Corso et. al.[6],the importance of a piece of news and the time of its emission were strictly related. To account for this, the author introduced a "freshness" decay parameter "α" that is obtained from the half-life decay time. Moreover, these algorithms took into account another parameter "β" which could be varied to tune the change in the rank of a news source based on the arrival of a fresh piece of news. Although an improvement from the Non-Time-aware algorithms, each of the proposed Time-aware algorithms also failed to satisfy one desirable property and failed to pass the second limiting case. However, for the purpose of ranking articles, Time-aware algorithms sufficed.

Yilmaz et. al.[7] studied the shortcomings of traditional rank correlation coefficients such as Kendall's τ and Spearman rank correlation coefficient for two ranked lists and proposed a new coefficient that penalized errors at high rankings more than the errors at low rankings and had an underlying probabilistic interpretation that was easy to understand. The AP rank correlation coefficient introduced by the authors possessed two important properties when compared to Kendall's : its equality to Kendall's τ when the errors are uniformly distributed over the entire ranking and the difference in value when the error in ranking is concentrated more at the top/bottom of the list.

3. PLAN OF WORK

Tentative Timeline

1. 3 Feb 2022 - 13 Feb 2022: Formulating the problem statement and doing literature review

2. 14 Feb 2022 - 27 Feb 2022: Data collection using Twitter API v2 and finding relevant articles

3. 28 Feb 2022 - 06 March 2022: Article collection completion

4. 15 March 2022 - 28 March 2022 - Preprocessing the data (tweets and articles) and producing baseline results

5. 29 March 2022 - 11 April 2022: Modeling the data to improve results: summarization/keyword extraction and similarity matching to find relevant articles

6. 11 April 2022 - 17 April 2022: Evaluating the results and using ranking related metrics

7. 18 April 2022 - 30 April 2022: WebApp creation, final report completion

4. REFERENCES

[1] Bordoloi, M. and Biswas, S.Kr. (2018). Keyword extraction from micro-blogs using collective weight. Social Network Analysis and Mining, 8(1).

[2] Mihalcea, R., Corley, C. and Strapparava, C. (2017). Corpus-based and Knowledge-based Measures of Text Semantic Similarity. [online] Available at: https://www.aaai.org/Papers/AAAI/2006/AAAI06-123.pdf

[3] Lösch, U. & Müller, D. (2011). Mapping microblog posts to encyclopedia articles. GI-Jahrestagung.

[4] Krestel, R., Werkmeister, T., Wiradarma, T.P. and Kasneci, G. (2015). Tweet-Recommender. Proceedings of the 24th International Conference on World Wide Web.

[5] Ahuja, S. (2015). Discovering significant news sources in Twitter. [online] IEEE Xplore. Available at: https://ieeexplore.ieee.org/document/7434278 [Accessed 26 Feb. 2022].

[6] Del Corso, G.M., Gullí, A. and Romani, F. (2005). Ranking a stream of news. Proceedings of the 14th international conference on World Wide Web - WWW '05.

[7] Yilmaz, E., Aslam, J.A. and Robertson, S. (2008). A new rank correlation coefficient for information retrieval. Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval - SIGIR '08.