A Query System for Social Media Signals

Dolan Antenucci, Michael R. Anderson, Penghua Zhao, and Michael Cafarella
University of Michigan Ann Arbor, Michigan 48109
{dol, mrander, phzhao, michjc}@umich.edu

***Abstract*—Social media *nowcasting*, the process of estimating real-world phenomena from social media data, has grown in popularity over the last several years as an alternative to traditional data collection methods like phone surveys. Unfortunately, current nowcasting methods depend on pre-existing, traditionally collected survey data as an aid to sift through the huge number of signals that can be derived from social media. This dependence severely limits the applicability of current nowcasting techniques. If we could remove this need for conventional data, social media signals could describe a much wider range of target phenomena.**

We have built a nowcasting querying system that estimates real-world phenomena without requiring any conventional data, relying instead upon an interactive exploration with users. Specifically, our system exploits a user-provided multi-part query consisting of *semantic* and *signal* components. The user can explore in real time the tradeoff between these two components to find the most relevant social media signals to estimate the target phenomenon. Our demonstration system lets users search for signals within a large Twitter corpus using a dynamic web-based interface. Also, users can share results with the general public, review and comment on others' shared results, and clone these results as starting points for further exploration and querying.

I. Introduction

Over the last several years, there has been a growing interest in social media *nowcasting*—estimating real-world phenomena with social media data.[1] Examples include flu activity [1], stock market behavior [2], and more [3]–[9].

The goal of a nowcasting system is to consume social media data as input and produce as output a time-varying signal that accurately reflects changes in a real-world phenomenon (i.e., not one-off events like riots or natural disasters). For example, when real-world unemployment rises, so should the nowcasting system's output signal, which is commonly based on the frequency of observing certain salient phrases (e.g., "I lost my job") on social media [1], [3], [8].

Nowcasting has the promise of being both faster and less expensive than traditional survey-based methods, which generally require costly person-to-person interaction. Such inexpensive high-quality datasets would be a boon for many fields—economics, public health, and others—that to a computer scientist appear simultaneously high-impact and data-poor. For example, US government economists use a relatively small number of expensive conventional economic datasets to make decisions that impact *trillions of dollars* of annual economic activity; even a tiny improvement in policy decision making can mean the addition of billions of dollars to the economy.

In spite of demonstrated successes of nowcasting in academic settings—including our ongoing work with several economists [8]—nowcasting has not penetrated deeply into broader use. We believe an important reason is that systems to date have focused on *one-off projects* that reproduce *a known dataset at low cost*.

[1]We use "social media" as an umbrella term for user-generated content, which includes tweets, web searches, blog posts, etc.

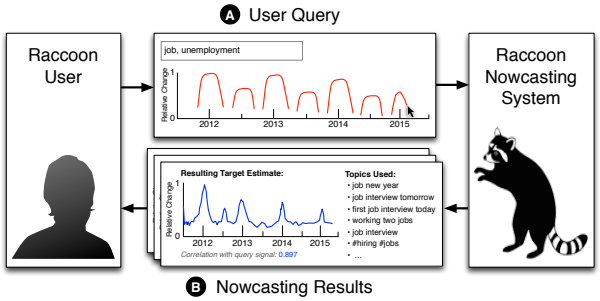


Fig. 1: The user-in-the-loop interaction of RACCOON, where a user can explore the impact of her query components in real time.

In contrast, we have built RACCOON, a prototype user-in-the-loop querying system for discovering novel nowcasting signals for phenomena that lack conventional survey-driven data. The user submits a query that encodes multiple forms of her domain knowledge: a *semantic query component*—a text string that describes the target phenomenon—and *signal query components*—a partial time-varying signal that represents the user's (likely incomplete) domain knowledge about the target phenomenon's signal (Figure 1, label A).

Our system returns a set of *nowcasting results* that best match this domain knowledge. Each nowcasting result consists of two parts: a list of text phrases relevant to the target phenomenon, which we refer to as *topics*, and a time-varying signal that estimates the phenomenon's relative change over time. The set of nowcasting results offer a range of options for the user to explore in real time (Figure 1, label B), each varying the impact each query component has on the result.

Contributions — The contributions of this work include:

1) A prototype user-in-the-loop querying system that allows users to choose the most relevant social media signals for a given real-world phenomenon without requiring any conventional ground truth data.
2) A prototype online community built around our querying system to encourage publishing, reviewing, and collaborating on nowcasting results.
3) A demonstration of these prototype systems using over 50 billion tweets (collected from mid-2011 to present), where conference attendees can perform queries in interactive time, view nowcasting results in a dynamic web-based interface, and explore the components of our online community.

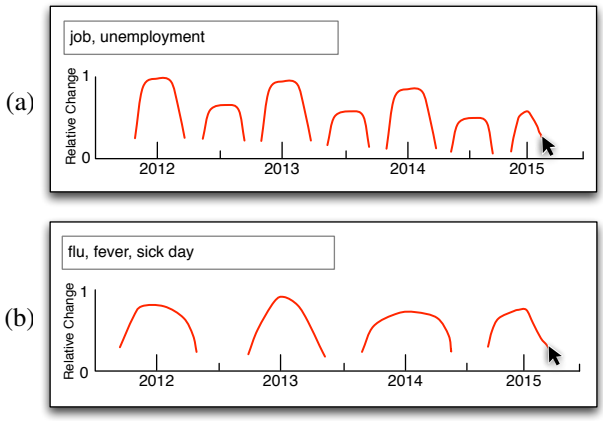


Fig. 2: Sample user queries illustrating the semantic and signal components of the two-part query.

II. Running Example

Our demonstration system is designed to allow conference attendees to understand the capabilities of our signal querying system, as well as to demonstrate the usefulness of nowcasting results. Attendees will be able to explore pre-selected phenomena or phenomena of their own choice. To illustrate a typical user scenario, consider a fictitious economist, Janet, using our system as a means to create inputs into economic models used to advise policy decisions. She wishes to create an estimate of weekly unemployment rates using signals derived from social media. We will refer back to Janet throughout the rest of this paper to help illustrate RACCOON.

III. System Framework

In this section, we give a brief overview of our system's inputs and outputs, our approach to processing a user's query, and our overall system architecture. A fully detailed discussion of these aspects of our system is covered in a separate paper, currently under submission.

A. Social Media Corpus

Prior to processing any user queries, our system requires a corpus of social media data. In particular, our demonstration system uses over 50 billion tweets collected between mid-2011 and present. After filtering out non-English tweets, we enumerate all the topics in the Twitter corpus by pre-processing the remaining 16 billion raw tweets to produce 1- through 4-grams using a Hadoop MapReduce system and store the daily and weekly counts of occurrences of each of these n-grams. As an example, consider the tweet "Happy Holidays." This single tweet would result in three topics: *happy*, *holidays*, and *happy holidays*. Each of these topics appear with varying frequency across the entire corpus of tweets, with *holidays* and *happy holidays* typically peaking in December and *happy* showing a much more consistent year-round trend.

After generating the topics, we remove rare ones that do not occur at least x times throughout our data's time period.[2] The final processed corpus consists of nearly 150 million topics and their associated frequency trends (our *topic-signal pairs*). Each week, we update these signals with the latest social media data using a similar offline process.

[2]We found that $x = 150$ removes many low quality phrases with little impact on result quality for real queries.

B. Query Design Considerations

Several reasonable query models can be proposed for a nowcasting system. In this section, we will discuss the problems facing both textual and signal-based query models and how RACCOON combines the two to overcome these challenges.

Textual Queries — A straightforward way of modeling queries in a nowcasting system is to ask users to describe their target phenomena textually. Using keywords or short phrases, a user has the flexibility to be as descriptive in their input as their domain knowledge allows. Even with lower-information inputs, the system can leverage semantic similarity to find good matches. For example, our user Janet may try estimating unemployment behavior by simply providing "unemployment" as the query's input, and the system would rank candidate topic-signal pairs in the corpus by their semantic similarity to this keyword.

Unfortunately, many topic-signal pairs in the social media database seem to be good semantic matches but are actually poor sources of nowcasting data. As an example, the topic *unemployment*, with its associating signal (derived from the number of tweets containing the word "unemployment"), indicates the genuine level of unemployment on most days, but for certain days of the month reflects a government data release and its subsequent press coverage instead.

Signal Queries — A slightly less obvious, though still reasonable, model is for users to specify a query as a time-varying signal, intending to find similar signals in the social media corpus. Official government data can be used to find similar signals, but this model is not limited to official data: it is also possible that as with semantic queries, users would only provide signal information for time periods in which they have some confidence. Janet may indicate that the target spikes after the holidays (i.e., when seasonal jobs end) and leave other time periods blank.

Unfortunately, signal queries have the same problem as semantic queries: it is easy to find topic-signal pairs that closely match a user's signal query component, but would be poor choices for building a nowcasting query answer. For example, we found that the time-varying social media signal associated with the topic *pumpkin muffins* is closely correlated with flu activity as reported by the US Center for Disease Control. This is not surprising: Fall baking and influenza trends both grow at the same time of year. However, an especially heavy year for flu will likely not also be a heavy year for pumpkin muffins. No epidemiologist would accept such data.

Our Approach — We address the pitfalls described with both of these query models by allowing a user to provide both types of query inputs and using both to select the candidate topic-signal pairs for the query answer. Results from an incomplete signal component will be improved by the contributions of the semantic component of the query. Further, semantic matches that are obviously irrelevant in the signal domain will be eliminated. This is akin to distant supervision [10], in that we are replacing high-quality ground truth trend data with

lower-quality, user-provided partial signals. By simultaneously using both signal and semantic query components, our system can produce higher-quality results: topics like *pumpkin muffins* and *unemployment* in the above examples would be rejected for not satisfying both parts of the user's query.

C. User Query Model

Figure 2 shows two examples of the two-part user query. In part (a), our economist Janet, who is searching for a signal to model unemployment behavior, enters the phrase "job, unemployment" as the semantic query component and draws from her domain knowledge to create a signal query component showing peaks of unemployment following the winter holidays and a smaller one during the summer months, each representing seasonal job losses. In this example, Janet does not accurately know the actual unemployment trend for the entire time span, so she intentionally leaves several gaps. The system can still process the query with an incomplete signal query component, finding signals that are highly correlated with the portions specified by the user.

In part (b) of Figure 2, a different user attempts to estimate flu activity by using a semantic query component of "flu, fever, sick day" and drawing a signal query component with wide peaks having apexes in mid-winter, based on known behavior of the flu infections. Clearly, there are many possible signal matches in a social media corpus to this query: topics about the winter holidays, for example, will have a very similarly shaped trend. However, they are likely not good candidates for generating a flu signal; if in the future there were a particularly hard or light year for flu, we would not expect tweets about winter vacation to change appropriately. Thus, the semantic query component allows our system to eliminate many spuriously related signals, narrowing the signal search to those that are semantically related to the user's specifications.

As can be seen by these examples, this query model is very flexible, allowing a user to describe many phenomena with as much or as little information as she can provide. Further, it allows experimental and aggressive hypotheses to be tested. Like a search engine, the system does not prejudge whether the query makes any sense. The system will do its best to answer it, letting the user judge the query's utility after the fact. While this experimental approach is familiar to search engine users, it is entirely novel for social science data production.

D. Query Execution

The raw data corpus consists of a large number n of topic-signal pairs, from which a much smaller number k are selected and aggregated into the output signal. (In our demonstration system, n is roughly 150 million and $k = 100$.) To select the k pairs for aggregation, we first compute two scores for each topic-signal pair: (1) a textual relevance score between the topic and the user's semantic query component and (2) the Pearson correlation between the topic's signal and the user's signal query component. Using these scores and a range of weights, we compute a set of weighted harmonic means and rank the topic-signal pairs by these values for each weight. Each ranked list represents a different tradeoff between the relative importance of the two parts of the user's query. The top k pairs for each ranking are selected for aggregation.

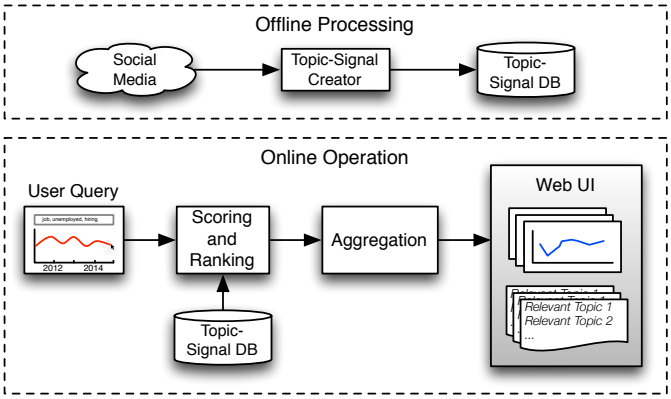


Fig. 3: The system's runtime architecture.

We use an Apache Spark cluster to process each query in roughly interactive time (with a typical query taking around 30 seconds to process on a seven-node cluster). Most of this processing time is spent on computing the semantic and signal relevance scores, both of which parallelize well. Our full processing pipeline certainly has room for optimizations—and indeed, our full paper (currently under submission) shows how to optimize runtimes to sub-second levels—but such work is not a focus of this current paper.

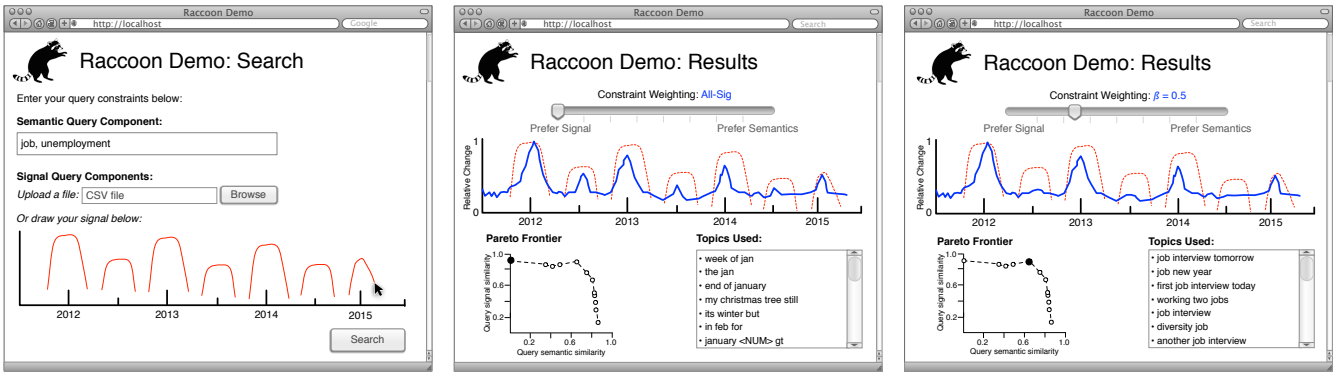
Our system combines the signals from the top k pairs using a PCA-based method [8] and then filters the topics to remove near-duplicates and reduce the number to a more user-comprehensible representative sample. Once complete, the system returns the resulting aggregated signal and associated topics to the user for further exploration.

E. System Architecture

A high-level view of our system's architecture is shown in Figure 3. The system has two main modes of operation. In offline processing, the topic-signal pairs are extracted and updated from the raw social media corpus as discussed in Section III-A. During online operation, a web-based interface accepts the user's query and launches multiple processes in parallel across a cluster of servers, which perform the scoring, ranking, and aggregation described in Section III-D. The aggregated results are then returned to the user's web browser and presented in a format allowing interactive exploration and evaluation (described further in the next section).

IV. User Interface

The user interface for our querying system consists of two main parts. First, the user enters her query using an intuitive two-part form (Figure 4a). The semantic query component is entered as a comma-separated series of words or phrases in a simple text field. The signal query component can either be uploaded via a CSV file or drawn directly on a time series-type graph using an interactive JavaScript-based widget. Partial or fragmented signals can be entered here, allowing the user to encode as much (or as little) domain knowledge she possesses about the actual historical behavior of her target phenomenon. Second, after submitting the form, our system processes the query and returns the results to the user on a page similar to Figures 4b and 4c. Here the user can use interactive controls to investigate the contributions of individual topic-signal pairs to the nowcasting result.



(a) Query Page **(b)** Search Result (All Signal) **(c)** Search Result (Chosen)

Fig. 4: User interface of the demonstration system.

Further, the user can explore *in real time* a tradeoff between the influence of her query's semantic and signal components via a slider widget. This tradeoff can be visualized in a "Pareto Frontier" plot (Figures 4b and 4c), where the x-axis and y-axis indicate each result's similarity with the user's semantic and signal query components, respectively. Each score generally decreases as the other's influence increases, where the "best" tradeoff can vary from query to query—and from user to user—because for any given query, a user may have more confidence in her signal query component than in her semantic query component, or vice versa. By letting the user adjust this balance, our system offers more flexibility across a range of phenomena.

While Janet is interactively exploring these results, she sees a nowcasting result (Figure 4b) that heavily favors her signal query component and that has a very high similarity score with her query signal; however, the topics used to create this signal have no relevance to her task and instead deal with the seasonal time period (e.g., "end of january"). Moving the constraint weighting slider to more heavily favor her semantic query component results in a signal generated from topics that are now very relevant to the unemployment task (e.g., "job interview tomorrow") and only cause a slight reduction of the similarity score with her signal query component (Figure 4c). Satisfied, she exports this result for use in her economic modeling, and then shares the result for other users to explore.

Sharing Results — One of the goals of our system is to create an online community where users can share interesting results with the public, as well as get feedback on potentially questionable results. If a user makes an outlandish claim supported by data generated by RACCOON, others can review the components that went into creating the data, respond with commentary, or "clone" the result as a starting point for their own exploration. This type of analysis and reproducibility is especially important for economists, where researchers at the US Federal Reserve recently showed they were unable to reproduce over 50% of selected published economic results—and that was even with the original authors' help [11].

In addition to allowing users to review, comment, and clone shared results, RACCOON can keep shared results updated regularly with new social media data, thus making it easy for anyone to monitor for changing trends. We envision economists and other researchers using this community as a means for better nowcasting collaboration, discovery, and debugging.

V. CONCLUSION

Our nowcasting querying system demonstrates the power of being able to extract time-varying estimates of real-world phenomena from data sources like Twitter, all without requiring any conventional ground truth data. By allowing users to describe their query in two parts in an interactive, user-in-the-loop manner, our system can build useful and trustworthy signal estimates. These estimates correlate well with the desired phenomena and, importantly, are semantically related to them as well. We believe a query system like this will allow the power of nowcasting to be brought to bear on many diverse real-world phenomena.

ACKNOWLEDGEMENTS

This project is supported by National Science Foundation grants 1054913 and 1131500; Yahoo!; and Google.

REFERENCES

[1] J. Ginsberg, M. H. Mohebbi, R. S. Patel, L. Brammer, M. S. Smolinski, and L. Brilliant, "Detecting influenza epidemics using search engine query data," *Nature*, vol. 457, no. 7232, pp. 1012–1014, 2009.

[2] J. Bollen, H. Mao, and X. Zeng, "Twitter mood predicts the stock market," *Journal of Computational Science*, 2011.

[3] H. Choi and H. Varian, "Predicting the present with Google Trends," Google, Inc., Tech. Rep., 2011.

[4] A. Sadilek, H. A. Kautz, and V. Silenzio, "Predicting disease transmission from geo-tagged micro-blog data," in *AAAI*, 2012.

[5] X. Zhu, J.-M. Xu, C. M. Marsh, M. K. Hines, and F. J. Dein, "Machine learning for zoonotic emerging disease detection," in *ICML*, 2011.

[6] N. Askitas and K. F. Zimmerman, "Detecting mortgage delinquencies," Forschungsinstitut zur Zunkuft der Arbeit, Tech. Rep., 2011.

[7] L. Mitchell, M. R. Frank, K. D. Harris, P. S. Dodds, and C. M. Danforth, "The geography of happiness: Connecting Twitter sentiment and expression, demographics, and objective characteristics of place," *PloS one*, vol. 8, no. 5, 2013.

[8] D. Antenucci, M. Cafarella, M. C. Levenstein, C. Ré, and M. D. Shapiro, "Using social media to measure labor market flows," National Bureau of Economic Research, Working Paper 20010, March 2014.

[9] D. Antenucci, M. J. Cafarella, M. Levenstein, C. Ré, and M. Shapiro, "Ringtail: Feature selection for easier nowcasting." in *WebDB*, 2013.

[10] M. Mintz, S. Bills, R. Snow, and D. Jurafsky, "Distant supervision for relation extraction without labeled data," in *ACL-IJCNLP*, 2009.

[11] A. C. Chang and P. Li, "Is economics research replicable? sixty published papers from thirteen journals say "usually not"," 2015.