Ringtail: A Generalized Nowcasting System

Dolan Antenucci
University of Michigan
dol@umich.edu

Erdong Li
University of Michigan
lierdong@umich.edu

Shaobo Liu
University of Michigan
bobliu@umich.edu

Bochun Zhang
University of Michigan
bochun@umich.edu

Michael J. Cafarella
University of Michigan
michjc@umich.edu

Christopher Ré
Univ. of Wisconsin, Madison
chrisre@cs.wisc.edu

ABSTRACT

Social media nowcasting—using online user activity to describe real-world phenomena—is an active area of research to supplement more traditional and costly data collection methods such as phone surveys. Given the potential impact of such research, we would expect general-purpose nowcasting systems to quickly become a standard tool among non-computer scientists, yet it has largely remained a research topic. We believe a major obstacle to widespread adoption is the *nowcasting feature selection* problem. Typical nowcasting systems require the user to choose a handful of social media objects from a pool of billions of potential candidates, which can be a time-consuming and error-prone process.

We have built RINGTAIL, a nowcasting system that helps the user by automatically suggesting high-quality signals. We demonstrate that RINGTAIL can make nowcasting easier by suggesting relevant features for a range of topics. The user provides just a short topic query (*e.g.*, *unemployment*) and a small conventional dataset in order for RINGTAIL to quickly return a usable predictive nowcasting model.

1. INTRODUCTION

Social media nowcasting—using online user activity to describe real-world phenomena—is an active area of research. Past projects have used social media data like search queries and Twitter messages to describe flu activity [13], unemployment [11], mortgage delinquencies [10], movie ticket sales [14], and other real-world processes.

The potential benefits of nowcasting are enormous. Traditional data collection techniques, such as phone surveys, are time-consuming and expensive. For example, the budget for the US Bureau of Labor Statistics—just one of the US government's statistical bureaus, and responsible for numbers such as the unemployment rate—is over 600 million US dollars each year [21]. If it were possible to vastly lower the cost of collecting data on social phenomena, researchers and policy experts could ask many more questions and get the answers much more quickly.

The direct impact of such data is potentially very large. For example, economist Alan Greenspan correctly predicted the mid-1970s recession would come to a close after analyzing "...ten-day auto sales figures, the weekly retail sales, the data on housing permits and starts, detailed reports coming out of the unemployment-insurance program, and so on..." [15]. Unfortunately, US automakers stopped releasing ten-day auto statistics in 1993 [19], making the modern policymaker's task more difficult. An effective nowcasting system could potentially help make better policy by yielding the auto sales number and many other signals. (Indeed, we are collaborating with two economists and presented an early version of this tool at the 2012 Summer Institute of the National Bureau of Economic Research [9, 7].)

Given their potential impact, we would expect general-purpose nowcasting systems to quickly become a standard tool among non-computer scientists. However, we are unaware of any such tool for sale, let alone in widespread deployment (two arguable exceptions are the use of Google Flu data by the US Center for Disease Control, and intermittent reports of social media data use by hedge funds [12]). Surprisingly, nowcasting has largely remained a research topic.

Nowcasting Today – We believe a major obstacle to widespread adoption is the problem of *feature selection* [8]. Consider the steps followed by most nowcasting research projects to date. First, the user chooses a number of salient phrases or queries (*e.g.*, "I feel sick" for flu prediction, or "I need a job" for unemployment). Second, the user uses the social media data to generate a time-varying signal for each such phrase (*e.g.*, the signal for "I feel sick" would count, for each day covered by the dataset, the number of people who searched for that phrase). Third, the user trains a statistical model that accepts the social media signals as input and outputs its estimate of the phenomenon's true value, such as the number of people with flu. This training procedure also requires a conventional data set, such as flu information collection via the health system.

The first step in this procedure is essentially one of feature selection, in which the user must determine the small handful of strings that should be used to build the time-varying signals. This step is much more burdensome than it first appears, because users are only weakly able to choose good signals. For example, consider the job-loss phrases *laid off*, *got let go*, *looking for a job*, and *was canned*; we derived Twitter signals for these phrases during the period of mid-

Proceedings of the VLDB Endowment, Vol. 6, No. 12

2011 to mid-2012, and measured the signals' correlation to official US initial unemployment insurance claims data. To the human eye, the four phrases seem reasonable, but their Pearson correlations ranged from a terrific 0.74 (*laid off*) to a terrible 0.14 (*looking for a job*).

As a result, users must repeatedly choose-and-test phrases until the nowcaster's performance is "good enough." Further, because nowcasting is useful in exactly those scenarios where conventional test data is rare, users must also be concerned with statistical issues such as overfitting. As a result, choosing good nowcasting signals is currently a time-consuming process that requires a statistical background. To obtain a popular general-purpose nowcasting tool, we would like a feature selection technique that takes this burden out of the user's hands as much as possible.

Technical Challenge – Feature selection is a well-known problem and has been studied intensively (see Guyon, *et al.* [16] for an overview). The nowcasting domain is unusual in its extreme paucity of conventional data when compared to the potential number of feature candidates.

Our current implementation of RINGTAIL contains roughly 3.2 billion possible features (the *(gram, signal)* pairs described in Section 2). In contrast, our conventional *unemployment* dataset is a government-collected dataset that is released just weekly, giving us about 52 data points that overlap with our year of social media data. Given this disparity in data size, any feature selection technique that chooses signals according to correlation with the conventional data will generate a massive number of spurious correlations. For example, ranking all of our 3.2B social media signals by correlation with the government unemployment data yields a list of suggested features in which the first truly relevant signal does not appear until position *1,376*. Asking a human to clean up such a list would likely be more difficult than simply asking the human for the features in the first place. Moreover, this problem is not limited to the *unemployment* target; nowcasting will always be most useful when conventional data is rare or unavailable.

Scott and Varian [20] described a "spike-and-slab" method for choosing nowcasting features, but it relies on rare conventional data. signals. Konda, *et al.* [17] recently described a system for feature selection in a more traditional structured data setting.

In other work [9], we described how to suggest features using semantic similarity information between the user's query and each candidate signal's label. This domain-independent technique relies on Web corpus statistics, and does not exploit the rare conventional data.

Demonstration – We have built RINGTAIL, a domain-independent nowcasting system that helps the user by automatically suggesting high-quality signals. We will demonstrate that RINGTAIL can make nowcasting easier by suggesting relevant features for a range of topic searches. The user must only provide a short topic query (*e.g.*, *unemployment*) and a conventional dataset in order for RINGTAIL to quickly return a usable predictive nowcasting model. RINGTAIL does not address every nowcasting problem, but does demonstrate that effective feature suggestion is possible.

In the rest of this paper we will provide an overview of RINGTAIL's data pipeline and query-time software architecture (Section 2), describe its user interface (Section 3), and discuss what our live demonstration will include (Section 4).

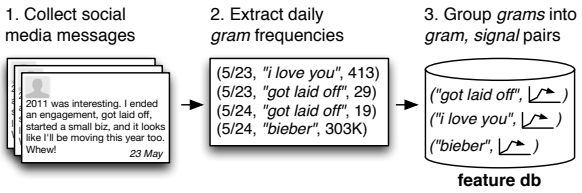

Figure 1: The pipeline RINGTAIL uses to convert a corpus of tweets into a set of *(gram, signal)* pairs.

2. SYSTEM FRAMEWORK

We now describe the basics of RINGTAIL's feature ranking system, the datasets needed to support it, and the software framework that supports user queries.

2.1 Feature Selection

Figure 1 describes RINGTAIL's signal extraction pipeline, which exists in most published nowcasting systems in some form. The system takes a collection of social media messages and transforms them into a series of *(gram, signal)* pairs. Each *gram* represents a string of 4 or fewer consecutive words drawn from social media text. For example, the tweet "lost my job" generates the set of grams, {*lost, my, job, lost my, my job, lost my job*}. Each accompanying *signal* contains a count of the number of times the gram was observed in the social media database in each 24-hour period. A number of social systems might be able to generate such messages; the only strict requirement is that the text have a timestamp.

We collected about 6 billion social media messages over a year-long period, which we transformed into about 3.2 billion *(gram, signal)* pairs with more than 3 occurrences in the data. We preprocessed the tweet corpus to remove punctuation and non-English messages, and to normalize some strings (*e.g.*, all URLs are translated into the generic $<URL>$ token). The task of feature selection is to choose k of these 3.2 billion *(gram, signal)* pairs to show to the user for use in training a statistical model. Because the conventional data can likely not support a large model, the ultimate number of desired signals is likely very small.

Note that RINGTAIL's primary goal is *not* necessarily to obtain the highest-accuracy nowcasting model. There are many factors that go into the model's success besides the selection of features: the difficulty of the target topic, the quantity of social media data, the choice of statistical model, and so on. Rather, RINGTAIL exists to make *(gram, signal)* suggestions that are close enough to human quality that non-computer-scientist users do not have to focus on this step, but can instead focus on the actual domain-specific implications of the nowcasting information.

2.2 Our Approach

Given a user's topic query q, RINGTAIL scores and ranks candidate features using a three-step process.

Synonym Expansion transforms the topic query into a number of roughly synonymous queries. The goal of this step is to make RINGTAIL feature selection robust to the user's word choice. In other words, queries for *gas prices* and *fuel prices* should yield similar and high-quality results. We simply run each of the t tokens in q through several standard thesauri, yielding t sets of synonyms; we then compute the Cartesian product of these sets. For example, if the topic query *gas prices* has token synonyms *gas = {gas, fuel}* and

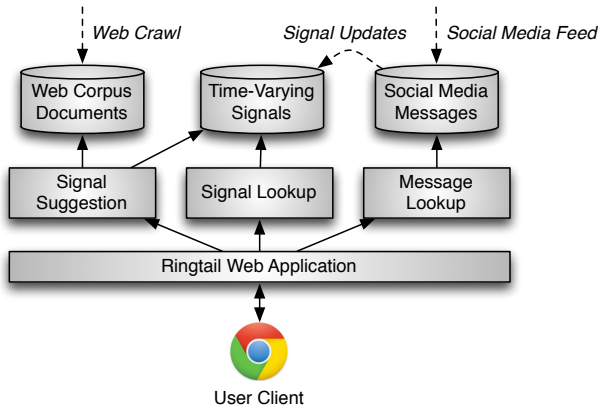

Figure 2: RINGTAIL's runtime software framework.

prices = {prices, costs}, we get expanded topic queries *{gas prices, gas costs, fuel prices, fuel costs}*.

PMI Scoring scores each candidate *(gram, signal)* pair according to the semantic relatedness between the gram and a topic query from the previous step. We then sort all the *(gram, signal)* candidates in descending order of relatedness, and pass the top k to the next step.

Pointwise Mutual Information, or PMI, is used in the Web mining literature to determine the relatedness of two strings. For a string x, $P(x)$ is the probability of seeing x in a corpus of text. $P(x, y)$ is the probability of seeing strings x and y together. PMI is defined as:

$$\text{PMI}(x, y) \equiv \log \frac{P(x, y)}{P(x)P(y)} \quad (1)$$

The resulting equation will yield a large value when x and y occur together more often than random chance. If a *(gram, signal)* pair is a good feature for the user's topic query, then we expect the gram to be highly related to the query string. Critically, we can use Web corpus data to compute PMI, all while not using any of the rare conventional data (*e.g.*, the governmental unemployment statistics). We use PMI to rank *(gram, signal)* candidates for each topic query, then combine them into a unified list sorted by average rank.

In the standard version of the demo, the output from **PMI Scoring** is what we show to the user, who can then choose to retain or eliminate *(gram, signal)* pairs. However, if the user is willing to build a predictive model using signals that do not necessarily have human-understandable grams, we can also pursue the next and final step.

PCA Data Reduction uses principal component analysis to transform the input signals into a space of features that are linear combinations of the original signals. We also obtain a ranking over these new synthetic signals that tells us which account for the most variance in the observed data. These synthetic signals likely capture a substantial amount of information in the entire PMI Ranking output, and our experiments show that these signals are in general higher-quality than those that stop at the previous step; however, the signals are not human-understandable.

2.3 System Architecture

Figure 2 describes the query-time software architecture of RINGTAIL. Several databases (at the top of the figure) are prepared offline before any user submits a query. Naturally,

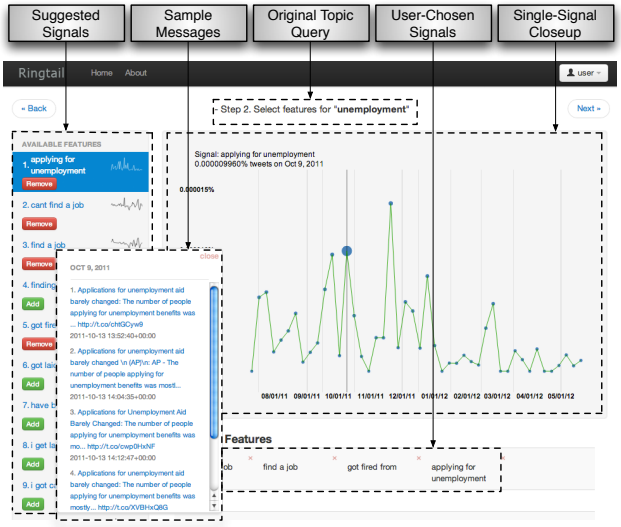

Figure 3: The RINGTAIL search result page shows **Suggested Signals** results in the left-hand column.

this offline phase includes recording a stream of **Social Media Messages** and transforming them into a set of **Time-Varying Signals**, or *(gram, signal)* pairs. We store the resulting signals in a distributed database (Apache HBase) for retrieval at query-time, and periodically update them with new data points. We also crawl **Web Corpus Documents** for use in the PMI step described in the above section.

Much of the intellectual contribution of RINGTAIL lies in the **Signal Suggestion** component. The **Signal Lookup** and **Message Lookup** components mainly serve interface ends, as we describe below in Section 3.

Much of the systems-building challenge of RINGTAIL lies in efficiently calculating PMI scores for each topic query. A single query requires a huge number of PMI scores (one for each candidate), a single computation could integrate frequency counts from many Web pages, and PMI is difficult to precompute (since in principle there are as many possible PMI scores as the square of the number of *grams*, which lies in the billions). RINGTAIL has three techniques for obtaining PMI scores. The first method is accurate but induces huge latency: for each query it runs a MapReduce job across a cluster of servers. The second method approximates the PMI scores by using a random sampling of the Web corpus. Some accuracy is lost, but speedup is near linear with corpus reduction. Finally, we have experimented with using matrix completion methods [18] to very efficiently approximate PMI scores even when we have relatively few samples of PMI scores. This last approach again trades accuracy of the scores for improved runtime performance.

3. USER INTERFACE

RINGTAIL's external presentation is that of a web application with three primary pages. The first page is a simple search box, which prompts the user to provide a topic query ("unemployment") and a conventional dataset (*e.g.*, the US weekly initial unemployment claims data for the past year). This is processed by RINGTAIL as described in Section 2.

The second page is shown in Figure 3 and is where the user spends most of his or her time. The **Suggested Signals** are the main output of the system, given in response to the

Target Phenomenon	Source	Potential User Label
Box Office Sales	B.O. Mojo [2]	movie tickets
Flu Activity	CDC [3]	flu rates
Gas Prices	U.S. EIA [6]	gas prices
Mortgage Refinancings	MBA [4]	mortgage refinance
E-commerce Traffic	Alexa [1]	online shopping
US Unemployment	US DOL [5]	unemployment

Table 1: Datasets the user can use to train a model.

user's **Original Topic Query**. The suggested signals can be highlighted by clicking on them. When a gram is clicked, the **Single-Signal Closeup** shows the corresponding signal data, as a percentage of social media messages over time. When the user clicks on a specific date within that signal, RINGTAIL displays some **Sample Messages** drawn from that date. Finally, the user can retain or reject RINGTAIL's suggestions to build the set of **User-Chosen Signals** that are sent to the last phase.

The final page allows the user to process the signals chosen in the previous step. It offers a few basic model training methods (*e.g.*, a simple linear regression, autoregressive model); these methods are offered as a convenience rather than a deep contribution of the system. The user can also download the signal data for use with an external tool.

4. DEMONSTRATION DETAILS

Our demonstration will allow conference attendees to use RINGTAIL to query for arbitrary topics and manage the resulting list of signals. We will put the system into a special "demonstration mode" that allows users to query for a topic even when a conventional dataset is not available; however, in this mode RINGTAIL can only suggest features, not train a working model. We will have several conventional datasets (listed in Table 1) for attendees to use; they are also welcome to upload and test any datasets of their own.

If conference attendees prefer to avoid writing their own queries and instead let the authors run a standard demonstration, we will show how an economic policymaker "Ben" might use the tool:

1. Ben is concerned about the unemployment rate. He visits the first page of RINGTAIL and enters "unemployment" into the search box. He also provides a recent weekly dataset: a year's worth of unemployment insurance claims. He then clicks on the **Go** button.
2. RINGTAIL quickly shows a result similar to that seen in Figure 3. Ben can see a number of signal grams proposed by the system: *I need a job*, *I got laid off*, and so on. When Ben clicks on a gram, the relevant time series is shown in the central window. He can also click on an individual data point to list tweets that contain the relevant gram on the day in question; this feature is often useful when trying to explain sudden and surprising moves in the data.
3. Ben adjusts the list of grams until he is satisfied. He then clicks **Next** to proceed to the final page of the application. On this page, he can train a simple regression model based on the chosen signals and the uploaded conventional data. The central window shows the conventional data series overlaid with a predicted signal generated by the trained model.

In addition, we will keep a running tally of past queries to suggest interesting demonstration ideas to potential users.

5. CONCLUSIONS

RINGTAIL is a new social media nowcasting system with a signal suggestion system that makes it easier to use than current published systems. It is enabled by a variable ranking mechanism that, critically, does not rely on conventional data to obtain high-quality suggestions. We believe the ideas behind RINGTAIL will make nowcasting systems a much more realistic proposition for typical domain experts and policymakers.

6. ACKNOWLEDGMENTS

This project is supported by National Science Foundation DGE-0903629, IGERT-0903629, IIS-1054009, IIS-1054913; Office of Naval Research N000141210041 and N000141310129; a gift from Yahoo!; and Ré's Sloan Foundation Fellowship.

7. REFERENCES

[1] Alexa Web Information Service.
[2] Box Office Mojo Weekly Box Office Index.
[3] CDC Influenza Surveillance Data via Google.
[4] Mortgage Bankers Association's Weekly App. Survey.
[5] US Department of Labor - Unemployment Insurance Weekly Claims Data.
[6] US Energy Information Administration - Weekly Retail Gasoline and Diesel Prices.
[7] Creating Measures of Labor Market Flows using Social Media. 2012 NBER Summer Institute.
[8] M. Anderson, D. Antenucci, V. Bittorf, M. Burgess, M. Cafarella, A. Kumar, F. Niu, Y. Park, C. Ré, and C. Zhang. Brainwash: A Data System for Feature Engineering. In *CIDR*, 2013.
[9] D. Antenucci, M. J. Cafarella, M. C. Levenstein, C. Ré, and M. D. Shapiro. Ringtail: Feature Selection for Easier Nowcasting. In *WebDB*, 2013.
[10] N. Askitas and K. F. Zimmerman. Detecting Mortgage Delinquencies. Technical report, Forschungsinstitut zur Zunkuft der Arbeit, 2011.
[11] N. Askitas and K. F. Zimmerman. Google Econometrics and Unemployment Forecasting. Technical report, Forschungsinstitut zur Zunkuft der Arbeit, 2011.
[12] J. Bollen, H. Mao, and X. Zeng. Twitter Mood Predicts the Stock Market. *Journal of Computational Science*, 2(1):1–8, 2011.
[13] J. Ginsberg, M. H. Mohebbi, R. Patel, L. Brammer, M. S. Smolinksi, and L. Brilliant. Detecting Influenza Epidemics Using Search Engine Query Data. *Nature*, February 2009.
[14] S. Goel, J. Hofman, S. Lehaie, D. Pennock, and D. Watts. Predicting Consumer Behavior with Web Search. *Proceedings of the National Academy of Sciences*, 2010.
[15] A. Greenspan. *The Age of Turbulence*. Penguin Press, 2007.
[16] I. Guyon and A. Elisseeff. An Introduction to Variable and Feature Selection. *Journal of Machine Learning Research*, 3:1157–1182, 2003.
[17] P. Konda, A. Kumar, C. Ré, and V. Sashikant. Feature Selection in Enterprise Analytics: A Demonstration using an R-based Data Analytics System. In *VLDB Demo 2013*, 2013.
[18] B. Recht, C. Ré, S. J. Wright, and F. Niu. Hogwild: A Lock-Free Approach to Parallelizing Stochastic Gradient Descent. In *NIPS*, pages 693–701, 2011.
[19] M. R. Rogers. *Handbook of Economic Indicators, 2nd Edition*. McGraw-Hill, 1998.
[20] S. Scott and H. Varian. Bayesian Variable Selection for Nowcasting Economic Time Series. Technical report, UC Berkeley School of Information, 2012.
[21] US Department of Labor, Bureau of Labor Statistics. The 2013 President's Budget for the Bureau of Labor Statistics, June 2012.