Ringtail: Feature Selection for Easier Nowcasting*

Dolan Antenucci
University of Michigan
Dept. of Computer Science
dol@umich.edu

Michael J. Cafarella
University of Michigan
Dept. of Computer Science
michjc@umich.edu

Margaret C. Levenstein
University of Michigan
Institute for Social Research
maggiel@umich.edu

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

Matthew D. Shapiro
University of Michigan
Department of Economics
shapiro@umich.edu

ABSTRACT

In recent years, social media "nowcasting"—the use of online user activity to predict various ongoing real-world social phenomena—has become a popular research topic; yet, this popularity has not led to widespread actual practice. We believe a major obstacle to widespread adoption is the *feature selection* problem. Typical nowcasting systems require the user to choose a set of relevant social media objects, which is difficult, time-consuming, and can imply a statistical background that users may not have.

We propose RINGTAIL, which helps the user choose relevant social media signals. It takes a single user input string (*e.g.*, *unemployment*) and yields a number of relevant signals the user can use to build a nowcasting model. We evaluate RINGTAIL on six different topics using a corpus of almost 6 billion tweets, showing that features chosen by RINGTAIL in a wholly-automated way are better or as good as those from a human and substantially better if RINGTAIL receives some human assistance. In all cases, RINGTAIL reduces the burden on the user.

1. INTRODUCTION

In recent years, social media "nowcasting"—the use of online user activity to predict real-world social phenomena—has become a popular research topic. Researchers have used search queries, Twitter messages, or similar data to estimate flu activity [18], unemployment levels [10], mortgage delinquencies [9], movie ticket sales [19], and more [11, 22, 25].

The motivation behind this work is clear: traditional data-collection methods, such as phone surveys or aggregating different sources of administrative data, are time-consuming and costly. As a result, researchers can ask relatively few questions; the answers to those questions are slow to arrive; and society pays substantial sums to obtain even these unsatisfactory results. 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 [24], and still cannot answer all the questions it would like. Social media nowcasting, which is relatively inexpensive and immediate, is a possible solution with real appeal to domain experts. This paper is the first in a series of collaborations between computer scientists and economists to apply nowcasting to real macroeconomic prediction tasks. (We presented some initial RINGTAIL results at the 2012 Summer Institute of the National Bureau of Economic Research [7].)

A more efficient method of observing social phenomena would be a boon for scientists, economists, and policy experts. Yet as of this writing, social media nowcasting is mainly a scientific curiosity—the subject of many research papers, but barely visible in practice[1]. Research suggests social media nowcasting could be applied to a huge range of applications, so why is it still a matter for research papers instead of a practical tool?

We believe a major obstacle to widespread adoption is the *feature selection problem.* Consider the steps the user follows in most nowcasting projects:

1. Aggregate the relevant objects over time to yield a set of time-varying *signals*—such as the daily frequency of various phrases.

2. Determine whether each signal is *relevant* to the target phenomenon (*e.g.*, perhaps the signal "I feel sick" is relevant for flu levels).

3. Use the selected signals, plus conventional data that describes the phenomenon (such as health system flu statistics), to train a predictive model; later, feed novel social media signals to the model to obtain nowcasting results.

The middle step, in which the user chooses a set of relevant social media objects, is a feature selection problem [8]. The post-aggregation database contains a vast number of candidate signals and the user must choose just a small number to yield a high-quality model in the final step. Most systems have used a human to identify a signal by listing strings that the social media object must contain. For example, a user interested in unemployment might choose a signal cor-

*Named after the ringtail, a "cousin" of the raccoon, known for scavenging morsels from others' garbage—similar to what we do with social media.

[1]Two possible exceptions are the use of Google Flu data by the US Centers for Disease Control and intermittent reports of using social media data by hedge funds [11].

Sixteenth International Workshop on the Web and Databases (WebDB 2013), June 23, 2013 - New York, NY, USA.

responding to all tweets that contain *laid off* and others for *got let go*, *looking for a job*, and *was canned*.

We believe this user-directed feature selection is much more burdensome than it first appears. The difficulty arises because users are only weakly able to choose good signals. For example, we obtained Twitter-derived signals for each of the above four phrases for the time period of mid-2011 to mid-2012 and measured their correlation to official US initial unemployment insurance claims data. To the human eye, each of the four phrases above seems reasonable, but their Pearson correlations with initial claims ranged from a terrific 0.74 (*laid off*) to a terrible 0.14 (*looking for a job*). In the experiments we describe later in this paper, the best three user-chosen signals for each of six different phenomena average a correlation of 0.58, while the worst three average just 0.15. Clearly, humans are unreliable signal-choosers.

This fact has terrible consequences for the usability of nowcasting systems. Users must preemptively choose a very large number of signals, or must engage in a repetitive choose-and-test loop 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, creating a nowcasting system is currently a time-consuming process that requires a statistical background. It is not surprising that non-computational domain experts have largely failed to embrace them.

Technical Challenge This paper describes RINGTAIL, a nowcasting system, which helps the user choose features. In response to a user's single topic query (*e.g.*, *unemployment*), it yields a number of signals the user can immediately use to build a statistical model. Rather than relying on user expertise or scarce conventional data (*e.g.*, as part of a *variable ranking* approach), RINGTAIL uses statistics from a Web corpus to obtain semantic similarity information between the user's query and each candidate signal's label. The resulting system is domain independent and yields results that in our experiments range from slightly better to roughly break-even with the human-suggested labels (depending on how many suggestions the humans give).

We do not claim that we have found the strongest possible architecture for a nowcasting system; indeed, we believe there are likely to be substantial changes to these designs in the future. Instead, RINGTAIL is designed to roughly emulate the designs embodied in most previous nowcasting systems, while using the feature selection techniques that are this paper's primary contribution.

Contributions This paper is organized as follows. We propose the RINGTAIL architecture for building and selecting nowcasting features (Section 3). We then present several feature selection techniques that do not consume precious conventional data (Section 4). Finally, we evaluate RINGTAIL on six different topics using a corpus of almost 6 billion tweets. Using multiple evaluation criteria, we show that features automatically chosen by RINGTAIL are on average better or as good as those from a human. (Section 5).

We discuss related work in Section 2. We think this paper addresses just the first of a wide range of interesting nowcasting questions, which we discuss in Section 6.

2. RELATED WORK

There are two main areas of work relevant to our research.

Nowcasting There have been many recent projects that have attempted to use social media to characterize real-world phenomena. Most use hand-chosen social media signals and a small amount of conventionally collected data to train a statistical model, which predicts a real-world value. Target phenomena have included mortgage delinquencies [9], unemployment rates [10], auto sales [22], and home sales [25]. Goel, *et al.* [19] used search logs to predict sales of media products. Choi and Varian [13] hand-select categorical search data for some phenomena—such as unemployment in the US—as well as use a statistical approach (spike and slab) for other phenomena, such as consumer confidence. Their work relies on a pre-classified set of features, which we do not have. Additionally, our selection method brings a novel information source to bear on the problem, and combining these approaches is a possible area of future work.

Ginsberg, *et al.*, which predicted flu levels from users' search queries [18], is a notable exception to the standard method of choosing input signals. Their system composes a time-varying signal for each of the 50 million most popular searches, then chooses the 100 that have the best correlation with the target flu signal. This approach was possible because of the long history of both search queries and the conventional flu data; it would not work with vastly more candidate signals, or many fewer conventional data points. Doornik proposed to make flu trend prediction more reliable through a combination of purely statistical techniques and use of a broader set of search queries [17].

Of course, in this paper we are primarily interested in other nowcasting projects for their method of choosing inputs. We do not compare their accuracy against our system's, because accuracy numbers depend not only on feature selection methods, but also on data availability, preparation techniques, and other factors beyond the scope of this paper. That said, the larger RINGTAIL project aims to obtain high-accuracy results to support economists who are interested in accurate nowcasting.

Feature Selection Feature selection is a well-known problem in the statistics literature. Guyon and Elisseeff [20] provided a good survey of the field. They place feature selection techniques into a few broad categories. With *domain knowledge* techniques, a human uses first-hand knowledge of a topic to choose features manually. As we discussed, humans' domain knowledge does not appear to be sufficient when choosing nowcasting features. *Variable ranking* techniques use a scoring mechanism—such as correlation criteria, single variable classifiers, or information criteria—to determine the relative worth of each potential feature (also called a signal or variable). These techniques are popular, but will overfit when the set of candidate signals is large and the conventional data used for computing the score is small; unfortunately, that is the common case for nowcasting applications. (Leinweber [21] provided a vivid example of the pitfalls linked to overfitting and spurious correlations.) Finally, there are *dimensionality reduction* techniques such as clustering, principal component analysis, and matrix factorization. These approaches consume no conventional data points and are applicable to nowcasting applications.

3. SYSTEM DESIGN

In this section we present RINGTAIL's basic architecture, including the *feature preparation* pipeline that most now-

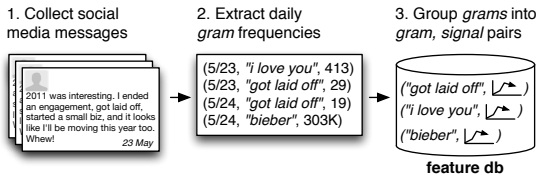

Figure 1: The pipeline RINGTAIL uses to convert a corpus of tweets into a set of *(gram, signal)* pairs.

casting systems have in common, plus the *feature selection* process that is unique to RINGTAIL.

3.1 Feature Preparation

Figure 1 illustrates the data preparation pipeline that most nowcasting systems have to some rough degree. The process starts with a large corpus of social media messages. In our experiments, we used almost 6 billion tweets collected between July 2011 and July 2012. Other implementations could use individual web search queries or similar messages. Each message might have various metadata (*e.g.*, username, source IP address, *etc.*) but the only strict requirement is that it has a timestamp.

In the next step, we aggregate these messages to obtain a large number of time-varying signals. This process is straightforward for very short texts like web searches: for each unique search, count the number of appearances in each 24-hour period over the collection's timespan.

For longer messages like tweets, the process is slightly more involved. For each tweet, RINGTAIL enumerates each sequence of t or fewer words into *grams*. For example, the tweet "lost my job" generates the grams {*lost, my, job, lost my, my job, lost my job*}. RINGTAIL then computes a signal for each of these unique grams observed in the entire corpus of tweets. Because a single tweet almost always contains multiple grams, it can contribute to multiple signals.

After processing the social media messages, we obtain the raw input to the nowcasting feature selection step: a huge number of *(gram, signal)* pairs—3.2 billion in our experiments when grams are 4 or fewer words. The user will eventually use a small number of these signals, plus a relatively small amount of conventional data, to train a statistical model, which predicts the target phenomenon. RINGTAIL's task is to choose the k signals from this massive set that will yield as accurate a model as possible.

3.2 Feature Selection

RINGTAIL's basic architecture is described in Figure 2. Note that in order to sidestep some computational issues that are outside the scope of the current work, we currently precompute some of the work in Step **D**, though we intend this work to be computed entirely at query-time—this is an active area of our research, which we describe in Section 6.

Using the pipeline described in Figure 1, Step **A** precomputes the candidate signals. The rest of the steps begin when the user enters a single *query* in step **B**. This includes a *topic query* describing the target—such as "unemployment"—and a conventional dataset—such as the US government's weekly data on unemployment insurance claims. The next step, *feature selection*, is where our contributions lie.

RINGTAIL feature selection comprises three steps. To start, we expand the user's *topic query* into a large number of gram candidates as pictured in step **C**. This uses an off-the-shelf thesaurus to expand each of the query's words and can yield

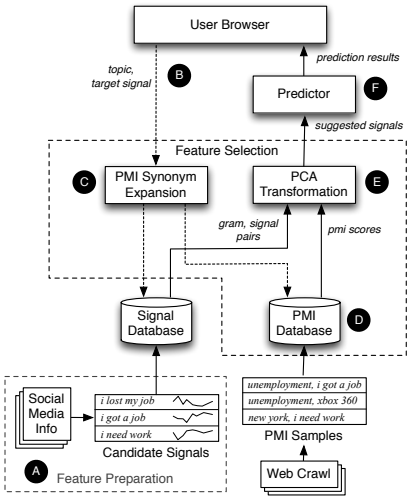

Figure 2: RINGTAIL's overall architecture.

up to several dozen *topic synonyms*. In **D**, we look up each topic synonym in the web-derived Pointwise Mutual Information (PMI) database; PMI is a method for measuring the semantic relatedness of two strings. Ranking candidate grams by PMI score can yield thousands of *synonym-PMI grams* that are related to the topic query. This set likely contains a large number of strings that should be semantically linked to the user's topic. We have not used any conventional data so far.

In **E**, we use principal component analysis to distill this still-large candidate set into a small number of synthetic signals we can return to the user. This process may cause us to lose signals that are "eccentric but correct" and so we must implicitly trust that any real-world phenomenon whose signal we want to observe will be captured by multiple social media grams. We have still not used any of the user's conventional data. Finally, in **F**, we use the conventional data to compute a statistical model and return it to the user. In the next section we examine RINGTAIL's feature selection steps—**C**, **D**, and **E**—in detail.

4. FEATURE SELECTION

RINGTAIL's central goal is to choose a small number of relevant features from the massive set of candidates we can compute from a social media corpus. Feature selection has a vast literature; we will explain that most standard techniques do not apply in the nowcasting setting, with its relative paucity of conventional data. We then describe our proposed solution, which uses as little conventional data as possible.

4.1 Dead Ends

We now describe a few common feature selection techniques that appear reasonable at first, but will not work in our target nowcasting setting. In the course of preparing this paper we tried each of them and met with no success (with the partial exception of subset selection, a limited form of which we incorporate into our final technique).

Domain knowledge relies on a knowledgeable human to suggest good features. This approach is not terrible in terms of result quality, and is widely used by other nowcasting systems. But in practical terms, a domain knowledge approach

yields a system that is difficult to use for all the reasons we described in Section 1: users have only weakly-accurate ideas about what signals are effective, and so engage in a repetitive choose-and-test loop that is both tedious and prone to overfitting.

Feature selection with signal data scores each feature according to a data-driven similarity metric based on the target phenomenon—such as Pearson correlation or R^2 from a single-variable regression. The underlying problem with this approach is the steep imbalance in size of our two datasets (*e.g.*, 3.2 billion *(gram, signal)* pairs vs. 52 conventional samples). Given so many candidate signals, it is easy to find variables that score highly through sheer chance. For example, the gram whose signal has the highest in-sample Pearson correlation with our unemployment dataset is *out this facebook* (0.991). It is unlikely that this gram has any predictive power for the unemployment level. Of the 100 most highly correlated grams, *none* of them are plausibly connected to unemployment.

Variable ranking with a human filter uses data-driven scoring to obtain an initial ranking, then asks a human to manually remove spurious correlations. This is a plausible technique when the number of spurious correlations is small; however, the number of spurious correlations in nowcasting settings is so large that simply examining this list amounts to a substantial burden on the user. For example, in our correlation-ranked list of variables, the first one that is plausibly connected to *unemployment* is *ski job*, which does not appear until position 1,376—and this may still not carry much information. RINGTAIL will not be very usable if each nowcasting query requires the user to manually examine thousands of candidates.

Subset selection with signal data attempts to find not just a ranking of variables, but the best possible combination of them. Forward selection starts from an empty set of variables and grows a high-quality set, while backward selection gradually removes variables from a full set. Unfortunately, subset selection methods suffer from the same data imbalance as the above methods. Spurious subsets of variables will appear to yield high performance only because of the limited size of our test data.

All of the above techniques are either labor-intensive or suffer because our conventional dataset is so small in comparison to the potential number of grams. In principle, variable ranking against the conventional signal data should work, if only we had sufficient data to avoid spurious correlations. Unfortunately, nowcasting systems are most useful exactly in those settings where conventional data is hard to obtain; we will never have as much as we would like.

It is easy to overlook that finding a source of this conventional data is only half the problem. We also need conventional data that *overlaps in time* with the social media data. Since no social media data source is more than a few years old, the best way to evaluate the relative scarcity of these data sources is how quickly we expect them to arrive in the future. By that measure, the quantity of social media data will always swamp conventional data. A small conventional dataset is not simply an artifact of our experiments, but rather has to be considered a basic challenge for usable nowcasting. Thus, the feature selection techniques we apply below are designed to use no conventional data at all.

4.2 Unsupervised Feature Selection

The central observation behind our approach is that while the signal part of each *(gram, signal)* pair is constrained by the availability of conventional data, the gram portion is not. When choosing signals by hand, humans are able to examine the gram alone and still yield good, though imperfect and burdensome, results. We attempt to build a system that similarly exploits common sense about the gram.

We use two techniques to expand the user's topic query into a large family of potential grams.

Synonym Expansion (SYN) In the first step we expand the user's single *topic query* into several roughly synonymous queries. A good topic query (*unemployment*) may be different from a good gram (*I need a job*), and the goal of this step is not to find good grams. Rather, our goal is to make the system robust to different word choices on the part of the user. Finding good grams is the next step.

Synonyms for a given topic query are generated from three sources: WordNet, Thesaurus.com, and Big Huge Thesaurus. The user's topic query is split on whitespace into tokens, and each individual token is run through these three services. Out of the resulting words and phrases returned, all permutations from the different sets are used. For example, if the input label "gas prices" returns the sets *gas* = {*gas, fuel*}; *prices* = {*prices, costs*}, then the final list of *synonym topic queries* would be {*gas prices, gas costs, fuel prices, fuel costs*}. This is a fairly naive expansion algorithm, but as we will see in Section 5.3, its performance is roughly comparable to human-generated synonyms.

PMI Scoring (PMI) For each topic query synonym, we now want to find all the related phrases that could embody useful signal data. With each query, we want to sort all grams in descending order of *relatedness* with it, then pick the top k. Fortunately, the information retrieval and Web search research communities have developed a straightforward technique for computing the relatedness of two strings.

Pointwise Mutual Information, or PMI, is often used to check for an association between words or phrases in a particular corpus [14]. For example, Turney used PMI to build a system that performed well enough to pass the synonym portion of a English language test [23].

Given two phrases x and y where $P(x)$ and $P(y)$ are the respective probabilities of each phrase occurring in the corpus, and $P(x, y)$ is the probability of the phrases occurring together, PMI is defined as

$$\mathrm{PMI}(x, y) \equiv \log \frac{P(x, y)}{P(x)P(y)} \tag{1}$$

If there is an association between the two, then $P(x, y)$ will be much larger than $P(x)P(y)$, thus yielding a high PMI score. If there is no association between the two, the PMI computation will yield something close to zero.

Computing these probabilities is a critical ingredient to PMI. We can calculate them using any large text corpus. We used the ClueWeb09 English web crawl dataset [15]. This contains 500 million English web pages crawled during 2009. We processed this corpus using Hadoop MapReduce. We define two grams as "related" if they occur within 100 tokens of each other.

To combine these n synonyms' PMI rankings into one list, we take the top 1,000 grams with the highest PMI for each topic query synonym. In principle, this gives us $n \times 1,000$

potential signals; however, many of these signals appear in multiple lists, so the number of unique signal phrases is less. We rank these signal phrases by the number of lists on which they appear. Where there are ties, we rank signals by their average rank across synonym lists.

Data Reduction (PCA**)** The first two steps expand the user's query, operating strictly at the text level. The final step performs data reduction on the actual signal data.

Once the features have been ordered, we then need to select features for use in nowcasting. Since there is a limited number of data points—just 52 when we have a year of social media data and the conventional data is a weekly signal—we can only select a handful of features to avoid overfitting when training the nowcasting predictor (step **F** in Figure 2).

We first explore *k-thresholding*, which involves selecting the top k features from each ranking method—our experiments in Section 5 set $k = 100$. While k-thresholding is a common practice, there may be a feature at *k+1* that carries important information, so we explored transforming a larger number of features with principal component analysis (PCA). PCA is used in a range of tasks that require unsupervised feature transformation, such as computer vision [16] and asset pricing [12]. The top j signals from each ranking mechanism are passed into the PCA algorithm—our experiments have $j = 500$. This yields a set of transformed signals, ordered by the amount of variance in the data explained by each. As before, we apply k-thresholding to the resulting list of signals.

5. EXPERIMENTS

The central experimental claim of this paper is that we can use automated methods to choose signals that are at least as good as those chosen by hand. In other words, we are concerned with the quality of the signals from step **E**.

Note we do *not* claim this method always obtains a high-accuracy nowcasting system (the results of step **F**). Nowcasting accuracy is dependent on many factors besides feature selection, such as the amount and type of social media data, the exact phenomenon being predicted, and so on. Poor prediction accuracy might be due to any of these causes.

5.1 Experimental Setup

Our initial design of RINGTAIL uses roughly 6 billion tweets collected between July 2011 and July 2012. We transformed these into roughly 3.2 billion candidate signals (Figure 1 and Step **A** of Figure 2) using a series of MapReduce jobs. We evaluated RINGTAIL on six phenomena (listed in Table 1) that are past or plausible future nowcasting applications. Each target has a label and a conventional dataset associated with it, which the user provides in Step **B**.

We lightly preprocess the tweets, removing punctuation and discarding non-English messages. We also translate "one-off" text strings such as URLs and reply indicators (*e.g.*, *@carlos*) into generic tokens (*e.g.*, *<URL>* and *<REPLY>*).

Each gram consists of a sequence of four or fewer tokens from a tweet, which are generated as described in Section 3.1. These are then aggregated together into *(gram, signal)* pairs, where the signal represents the daily frequency for each gram between July 2011 to July 2012. Finally, we normalize the resulting signals to account for growth in total tweets. Each conventional dataset is weekly. For correlation and R^2 metrics, we convert the target signal and *(gram, signal)* pairs into four-week moving averages.

Target Phenomenon	Source	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: Target phenomena used for testing. US Unemployment refers to the weekly number of initial unemployment insurance claims.

5.2 Feature Quality

We now show that RINGTAIL can obtain social media features that are at least as good as those given by a human expert. We first describe our evaluation metrics, then the benchmarked techniques.

Evaluation Metrics – We evaluate the quality of the emitted set of signals using three metrics.

- **Average Correlation** – For each signal in the emitted set, we measure the Pearson correlation with the conventional data signal. Each emitted signal and the conventional data signal consist of the entire conventional data timespan—in our case, 52 data points. We average over all the emitted signals. Values are the absolute value of correlation, ranging 0 to 1, higher being better.

- **Average** R^2 – For each signal in the emitted set, we perform a linear regression computation with the signal as a predictor variable and the conventional data signal as the response variable. We compute the R^2 error quantity that arises from the regression. This is a standard error metric in economics and other fields. Again, we use the entire 52-week dataset and average over all emitted signals. Values range 0 to 1, with higher being better.

- **Average Mean Absolute Error (MAE)** – Finally, we compute the accuracy of the predictor in step **F**. This is the only metric that evaluates our performance using held-out conventional data. For each signal in the emitted set, we build a series of linear regression predictive models using subset selection (max three features per model). Each model uses 30 data points from both the emitted signal and the conventional data series. We then ask the model to predict the value for the *next* data point in the conventional series. The difference between the prediction and the held-out data is the error. We average this error over $52 - 30 = 22$ rolling predictors. Finally, we average the MAE from each of the models. MAE is described in percentages, and smaller values are best.

Benchmarked Techniques – We tested several feature selection methods, all different combinations of the three techniques described in Section 4.2: SYN, PMI, and PCA. We compared these approaches against two baseline methods. The first, RANDOM, is simply a set of 100 signals drawn randomly from the overall signal database. The second, HUMAN, is the result of asking seven graduate students to each suggest 10 relevant grams for each target phenomenon (*e.g.*, *I need a job* for US unemployment). RINGTAIL's goal is to match or beat HUMAN's quality with an entirely automated method.

Metric	**Random**	**Human**	**PMI**	**PMI+Syn**	**PMI+PCA**		**PMI+Syn+PCA**	
					$k = 100$	$k = 3$	$k = 100$	$k = 3$
Average Correlation (larger is better)	0.2448	0.3630	0.2567	0.2543	0.1669	**0.4496**	0.1637	0.3993
Average R^2 (larger is better)	0.0912	0.1809	0.0986	0.0970	0.0474	**0.2406**	0.0449	0.2176
Average MAE (smaller is better)	0.1661	**0.1277**	0.1282	0.1308	0.1293	0.1313	**0.1278**	0.1349

Table 2: Evaluation of different feature selection mechanisms, averaged over the tasks in Table 1. Best ones, or close to best, are bolded.

5.3 Experimental Results

Our experimental results are summarized in Table 2. Not surprisingly, HUMAN outperforms RANDOM on all three metrics. In addition, at least one of RINGTAIL's techniques can beat or essentially tie HUMAN. For *Average MAE*, the best RINGTAIL technique is the one that uses all three approaches from Section 4.2 (PMI+SYN+PCA).

For *Average Correlation* and *Average* R^2, the result is somewhat different. When $k = 100$—as is the case with the other benchmarks—PMI+PCA and PMI+SYN+PCA perform worse than RANDOM on average, yet the top signals ($k = 3$) easily outperform HUMAN on average. We suspect this is due to the the way PCA ranks the signals by the amount of variance of PCA inputs that they explain. The bottom ones carry little of the information that the top ones do, so they perform quite poorly, thus lowering the overall average. While the seemingly low correlation and R^2 values in Table 2 may cause one to question the quality of RINGTAIL, we emphasize that we make no claim the signals are perfect, only that we can essentially match a human on average (0.1277 for HUMAN, vs 0.1278 for RINGTAIL). Although correlation and R^2 are useful metrics, MAE is likely the metric the typical nowcaster will want to maximize.

To test if there is room for improvement with our synonym combining heuristic, we asked a human choose the synonyms manually instead of using an automated thesaurus-driven process. Results show this algorithm performs 5% better than HUMAN using average MAE, compared to the tie that arises from our automatically generated synonyms.

6. FUTURE WORK AND CONCLUSION

We have demonstrated that RINGTAIL can suggest features as well as a human on a range of nowcasting tasks. We believe this approach is a critical step in making nowcasting a practical tool rather than a research curiosity.

Our immediate future focuses on efficiently computing PMI values (step **D**), which we believe would also have an impact on other Web research projects; we are currently exploring approximation techniques for PMI. RINGTAIL avoids conventional data for most of the feature selection process, but still uses it to train a statistical model (bottom row of Table 2). Nowcasting would be most useful in cases where conventional data does not exist at all. Indeed, it would be terrific if the predictive model could be built using social media data exclusively. Finally, improving nowcasting accuracy will likely continue to be a lively research area.

7. ACKNOWLEDGMENTS

This project is supported by National Science Foundation DGE-0903629, IIS-1054009, IIS-1054913 and SES-1131500; Office of Naval Research N000141210041 and N000141310129; a gift from Yahoo!; and the Sloan Foundation Fellowship to Ré. The project is part of the University of Michigan node of the NSF-Census Research Network.

8. 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 (MBA) Weekly Applications 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] N. Askitas and K. F. Zimmerman. Detecting Mortgage Delinquencies. Technical report, Forschungsinstitut zur Zunkuft der Arbeit, 2011.
[10] N. Askitas and K. F. Zimmerman. Google Econometrics and Unemployment Forecasting. Technical report, Forschungsinstitut zur Zunkuft der Arbeit, 2011.
[11] J. Bollen, H. Mao, and X. Zeng. Twitter Mood Predicts the Stock Market. *Journal of Computational Science*, 2011.
[12] G. Chamberlain and M. Rothschild. Arbitrage, Factor Structure, and Mean-Variance Analysis on Large Asset Markets, 1984.
[13] H. Choi and H. Varian. Predicting the Present with Google Trends. Technical report, Google, Inc., 2011.
[14] K. W. Church and P. Hanks. Word Association Norms, Mutual Information, and Lexicography. *Computational Linguistics*, 16(1):22–29, 1990.
[15] The ClueWeb09 Dataset, via Lemur Project.
[16] F. De la Torre and M. J. Black. Robust Principal Component Analysis for Computer Vision. In *Computer Vision, 2001. ICCV 2001. Eighth IEEE International Conference*, volume 1, pages 362–369. IEEE, 2001.
[17] J. A. Doornik. Improving the Timeliness of Data on Influenza-Like Illnesses using Google Trends. Technical report, Oxford University, 2010.
[18] 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.
[19] 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.
[20] I. Guyon and A. Elisseeff. An Introduction to Variable and Feature Selection. *Journal of Machine Learning Research*, 3:1157–1182, 2003.
[21] D. Leinweber. Stupid Data Miner Tricks: Overfitting the S&P500. *The Journal of Investing*, 16(1):15–22, 2007.
[22] Y. Qingyu, P. Geng, L. Ying, and L. Benfu. A Prediction Study on the Car Sales Based on Web Search Data. In *International Conference on E-Business and E-Government (ICEE)*, 2011.
[23] P. D. Turney. Mining the Web for Synonyms: PMI-IR versus LSA on TOEFL. In *ECML*, pages 491–502, 2001.
[24] US Bureau of Labor Statistics. The 2013 President's Budget for the Bureau of Labor Statistics, June 2012.
[25] L. Wu and E. Brynjolfsson. The Future of Prediction: How Google Searches Foreshadow Housing Prices and Sales. Technical report, MIT Sloan School of Management, 2009.