ImpactRank: A Study on News Impact Forecasting

Sukru Tikves
School of Computing, Informatics,
and Decision Systems Engineering (CIDSE)
Arizona State University
Tempe, Arizona 85281
Email: Sukru.Tikves@asu.edu

Hasan Davulcu
School of Computing, Informatics,
and Decision Systems Engineering (CIDSE)
Arizona State University
Tempe, Arizona 85281
Email: Hasan.Davulcu@asu.edu

Abstract

In this paper we developed a framework and a measure for news impact forecasting. We proved the viability of our impact forecasting approach using a SVM based forecaster on six months of NYT corpus - consisting of 16,852 articles. We experimented with different feature selection and ranking algorithms including standard frequency based methods, as well as a new method named ImpactRank. Our ImpactRank based forecaster performed as the best feature ranking technique while providing a graph suitable for browsing and identifying the most influential topics, entities and inter-relationships going into its impact predictions.

1. Introduction

In this paper we worked on measuring and forecasting the impact of news events as they occur on a timeline. An "event" is something that happens at some specific time and at a place[1], e.g. "train bombing in London on July 7th".

In order to forecast the impact of incoming news events, we rely on a model based on partial impact calculations of past news. Partial impact calculations are based on similarity relationships among news, and the events and entities mentioned within them.

Relevant research on ranking of the impact of conference papers rely on citation information among scholarly works. Unlike news, citation information among scholarly publications might be readily available or can be extracted from their text[2].

Bergstrom introduced eigenfactor, which calculates impact factor scores for journals and other scholarly publications[3] based on their citation graphs. This followed Garfield's early work on impact factors[4], which was criticized by [5], [6], [7], since it only relied on one-level immediate citations. ArnetMiner group used a hybrid model for their ranking system[8]. Similarly, in order to measure the quality of the scientific output of an individual researcher, Hirsch[9] proposed the popular *h index*.

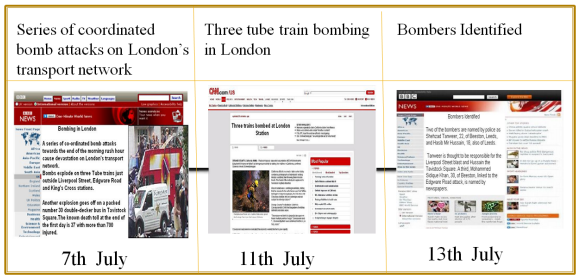

Figure 1. Snapshot of an Example Event Thread

While citation information is readily available in the scholarly publications domain, in the news domain we needed to identify events mentioned in news and rely on similarities among their topics and entities to identify their impact. Fortunately, previous TDT research[10], [1], [11] provides us with the tools and techniques for both (i) First Story Detection (FSD) to identify if a news article is talking about a new story, and (ii) Topic Tracking to relate incoming news stories with the related past stories. In Figure 1 we present a snapshot of the event thread of "train bombings in London" which contains the initial "bombing attacks on July 7th", follow-up coverage on July 11th, and the identification of the bombers on July 13th.

In Section 3 we propose a definition for an impact value measure for news articles, based on the order of number of related articles following it. This measure provides us with a baseline to (i) assign partial impact values to all articles up to a certain date, (ii) experiment

with different feature selection and ranking algorithms to generate feature vectors for news articles, (iii) train a predictive classifier, based on Support Vector Machines utilizing feature vectors of articles and their partial impact scores, and (iv) generate a gold standard of impact measures (with look ahead) to measure the overall accuracy of the SVM classifier [12], [13] for predicting the impact of incoming news based on information gleaned from the past.

We experimented with different feature selection and ranking algorithms (including standard frequency-based *tf*, *tf-idf* methods), as well as a new ImpactRank network model, which is based on the TermRank algorithm introduced by Gelgi [14]. Based on our experiments, this eigen-vector based new measure performed as the best feature ranking technique, and the corresponding ImpactRank network model served as the best method for identifying influential topics, entities and inter-relationships mentioned within an article to explain its predicted impact.

Next Section presents the overall design of the impact estimation procedure. Section 3 formally defines the impact measure. Section 4 provides the details of the SVM based impact forecasting. Section 5 introduces and examines the ImpactRank network model and the TermRank algorithm for feature ranking. Section 6 describes the experimental setup, and in Section 7 we present experimental results. Section 8 concludes the paper.

2. Impact Estimation Procedure

Our architecture for investigating the news impact problem consists of two main components: The first component, runs offline through the entire sequence of news articles, computes, and stores the gold standard values of their impact scores. The details of this components is presented in Section 6.1.1. The second component simulates a real time loop, by continuously testing various impact estimation methods with increasing amount of data, and enables us to assess their accuracies.

In the initial step of the loop, the estimator is supplied with an input of one month of news data and asked to train itself using partial impact scores calculated only using this one month input. After the training is complete it's then asked to estimate the impact scores of the articles for the following two weeks of data. The process for this step is illustrated in Figure 2.

After the initial step, the rest of the dataset is added into the input steam in weekly increments. After each weekly data addition, the partial impact scores are

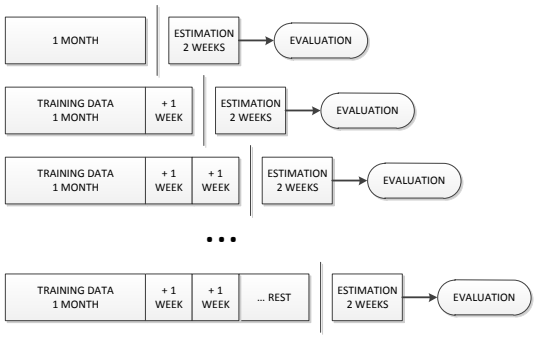

Figure 3. Continuation of the Evaluation Loop

updated, the estimation model is re-trained with the updated partial impact values, and the estimator performance is evaluated for the next two following weeks of news data. These progressive steps are illustrated in Figure 2.

Our choice of 1 month of initial training, 2 weeks of estimations, and weekly increments of the data are not fixed, and chosen for practical results. Any other arbitrary choice of ranges, in weeks, days, hours, or even article-by-article continuations, are all possible.

3. Definition of the Impact Measure

The first challenge in our research is identifying a meaningful definition for measuring the impact of news articles. Since, we don't have explicit references between news articles, the measure should rely on implicit relationships. Additionally, it should be robust, and resistant to small variations.

Towards an impact measure, first we start with a measure of *follow up* length, which will be used in the definition of the impact itself:

Definition 1: Follow up length of an article is defined as the number of news articles that (i) chronologically follow that article, and (ii) lies within a predetermined perimeter of similarity.

Although the above definition is generic, in this paper we will use cosine similarity[15] as the distance metric, and an experimentally determined similarity threshold discussed in Section 6.1.3.

An illustrative news flow configuration is shown in Figure 4. In the figure, for visibility purposes, every back link is assumed to be transitive, while in real life situations they might not necessarily be so.

Using the *follow up* length, it's now possible to define the *impact score* of a news article:

Definition 2: The *impact score* of a news article is defined as the logarithm of the *follow up* length of that

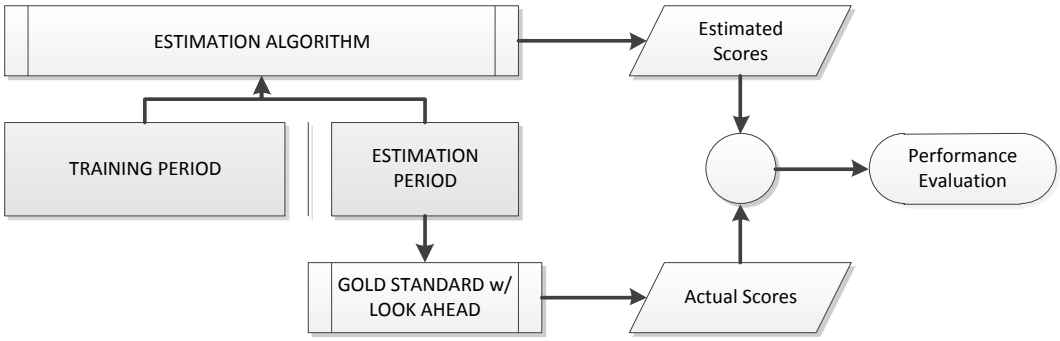

Figure 2. Initial Step of the Evaluation Loop

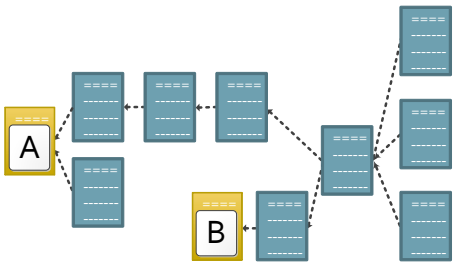

Figure 4. Example impact flow. Here document (A) has a follow up length of 8, and document (B) has a follow up length of 5.

article, in base 10, *restricted* into the [0, 3] range.

For an article a:

$$Impact(a) = min(log_{10}(FollowUp(a) + 1), 3) \quad (1)$$

Employing *follow up* length allows us to have a measurable follow up calculation for impact definition. The choice of logarithmic scale introduces a means resistant to small variations in larger follow up configurations. It builds upon order of magnitude, where impact score around 1 expressing a follow up in tens-of-articles, 2 in hundreds, and 0 indicating singleton events. A limit of 3 in impact score assumes all events having follow up size of thousand or more articles implies highest impact.

4. SVM Based Forecasting

Support vector machines (SVM) have been effectively employed in text processing tasks[12], and they are the primary choice in our system as the estimation algorithms, due to their performance and versatility.

Feature selection and ranking are an important part of support vector machine based classification. In our system, we used term frequency based feature vectors as a baseline. In addition, better feature selection and ranking methods will be presented in Section 4.1.

After each document was tokenized into a stream of words, stop words were removed from that stream, along with any one or two character long entries. The resulting list was then converted into lower case, and the standard TF-formula (2) was used to generate the document term vectors.

$$tf_{i,j} = \frac{ni,j}{\sum_k n_{k,j}} \quad (2)$$

Since we utilized SVM as a discrete learner, target values for the impact scores were binned into 7 categories; each representing a rounding to the nearest multiplier of 0.5. It should be noted that, in case of the uniform distribution of impact scores, this will introduce an expected minimum error around 0.21 even for a perfect estimator.

Our preliminary experiments with standard frequency based term-vectors demonstrated that the support vector machine was indeed capable of learning a model of news impacts, and improve incrementally during additional iterations.

4.1. News Feature Selection and Ranking

In addition to the basic term frequency representation, which was described in section 4, several other feature selection and ranking methods were also implemented.

The following are the list of all methods evaluated in our study:

- **SVM-Basic** The basic method in Section 4, which uses term-frequency for representation, without ranking (i.e. using all terms).
- **SVM-TF** An evolution of this method, which selects only top K ranked keywords.
- **SVM-TF (EV)** Uses keywords as well as recognized named entities, and term-frequency.
- **SVM-TF-IDF (EV)** An iteration of previous method, which uses tf-idf in Formula (3), instead of term-frequency.

$$tf - idf_{i,j} = tf_{i,j} \times log \frac{|D|}{|\{d : d \in D \wedge t_i \in d\}|} \tag{3}$$

- **SVM-Stable** Feature ranking according to the inverse of the variance of per-term impact values, as shown in Formula (6), with term-frequency based vectors. The intuition for this method is based on the fact that $1/variance$ captures the stability of the impact value of a term within the document corpus.

$$D'_i = \{d : d \in D, t_i \in d\} \tag{4}$$

$$\overline{impact_i} = \frac{\sum_{d \in D'_i} Impact(d)}{|D'_i|} \tag{5}$$

$$inv - var_i = \frac{|D'_i|}{\sum_{d \in D'_i} (Impact(d) - \overline{impact_i})^2} \tag{6}$$

- **TermRank** The TermRank algorithm is presented in the following section.

5. Term Rank Algorithm

TermRank is an eigen-vector based ranking algorithm introduced by Gelgi[14] for use on web based document collections. The main advantage of TermRank over traditional td-idf based ranking methods is its ability to distinguish among discriminative, ambiguous and common terms by detecting their various contexts and rank them accordingly as described below:

- Discriminative terms typically strongly relate to a specific high impact context. Topical keywords such as "bombing", "scandals", "beheadings" and named entities such as "Saddam Hussein" belong to this category.
- Ambiguous terms tend to appear in many contexts, however their impact might vary depending on the strength of their association with a certain context. For instance, keywords such as "force" and "increase" may be found in many contexts. However, whenever they are strongly associated with a certain context, their impact should evolve along with the impact of the context.
- Common terms usually appear in many contexts. Therefore, unlike ambiguous terms, common terms only have weak connections with their contexts. Some examples of common terms are "Washington" and "American". The impact of common terms should be lower than discriminative terms with persistent impact, and ambiguous terms with strong associations to high impact contexts.

TermRank algorithm is based on a variation of PageRank[16], [17]. However unlike web graphs, where there are explicit directed links between nodes corresponding to documents, TermRank works on textual term data, where nodes correspond to terms, and by extracting their relationships (i.e. co-occurrence) from the text. PageRank operates on a directed graph where edges have no weights. Whereas, TermRank works on undirected graphs with weighted edges. Hence, all edges are considered to be both incoming and outgoing. Since there are no rank sinks in undirected graphs, a decay factor is not included in the TermRank formula. Given a relationship graph $\mathcal{G}$, TermRank is calculated as follows:

$$TR(i) = \sum_{j \in \mathcal{N}(i)} \frac{TR(j).w_{ij}}{\sum_{k \in \mathcal{N}(j)} w_{jk}} \tag{7}$$

where $\mathcal{N}(i)$ represents the set of neighbors of the node i. The essential difference in the formulas can be summarized as the summation of the weights of edges instead of the number of links. Similar to PageRank, TermRank, can be very efficient approximated with the following iteration method presented in Equation 8:

$$\begin{aligned} TR^{(0)}(i) &= \frac{w_i}{\sum_{j \in \mathcal{V}(\mathcal{G})} w_j} = TF(i) \\ TR^{(t+1)}(i) &= \sum_{j \in \mathcal{N}(i)} \frac{TR^{(t)}(j).w_{ij}}{\sum_{k \in \mathcal{N}(j)} w_{jk}} \end{aligned} \tag{8}$$

5.1. ImpactRank: TermRank for Impact

We define a variation of TermRank graph that operates on news articles' impact values as follows:

$$ImpactRank = TermRank(G(N, E)) \tag{9}$$

Where G is a graph with node set N corresponding to the terms in the document corpus, and a weighted

edge set E, corresponding to term co-occurrence frequencies.

For the TermRank based feature ranking method, a combination of filtered keywords, and recognized named entities were used as the nodes. Additionally sentence boundaries determined the co-occurrence windows to establish the edges. The normalized term co-occurrence frequencies were used to set edge weights. Node weights were initialized with the average impacts of the corresponding terms.

The evolution of the highest ranked terms in the ImpactRank graph can be observed in Table 1. The first column corresponds to the initial ordering of the top nine terms with respect to their average impact scores, according to Formula (6), for the network extracted for January 14, 2007. Terms starting with the text "LOC/", "ORG/" and "MSC/" refer to recognized named entities. The second column correspond to the output of the TermRank iteration. It can be observed that, many common and ambiguous terms have been replaced with discriminative, and ambiguous but strongly associated terms. The *ORG/Congress* term remains as a stable discriminative term enjoying persistent highest ranking throughout 1/14 - 1/18 period. After each daily iteration, common terms such as *LOC/Washington* and *American* gradually loses their initial higher rankings (i.e. due to their initial higher frequency) against ambiguous but strongly impact associated terms such as *troop* and *increase*.

6. Experimental Setup

6.1. Evaluation Process

The evaluation process was performed in two stages. In the first stage an offline application was utilized to calculate the impact scores with access to the entire data, generating a gold standard for the news impact evaluation with complete follow up information. The second stage simulated a real time execution, by incrementally feeding weekly data to various forecasting algorithms, and recording their performance for plotting the evaluation charts presented in Section 7.

6.1.1. Offline Calculation. In order to provide a test data set, a version of the algorithm presented in Figure 5 was executed with access to the entire corpus. Lines 1-8 iterates through all articles, and updates the follow up lengths. The last three lines performs a second pass on articles, filling in the impact scores by applying Formula (1). The results are then recorded as the gold standard news impact scores.

Figure 5. Impact Score Calculation

1: **for all** incoming article a_i **do**
2: $\quad followup_i \leftarrow 0$
3: $\quad$ **for all** previous article a_j, $j < i$ **do**
4: $\quad\quad$ **if** $Distance(a_i, a_j) < Threshold$ **then**
5: $\quad\quad\quad followup_j \leftarrow followup_j + 1$
6: $\quad\quad$ **end if**
7: $\quad$ **end for**
8: **end for**
9: **for all** articles a_i **do**
10: $\quad impact_i = min(log(followup_i + 1, 10), 3)$
11: **end for**

Figure 6. Online Evaluation

for all article a_i in First Month **do**
$\quad$ ADDTRAININGDATA(a_i)
end for
for all week w_j after First Month **do**
$\quad$ TRAIN
$\quad$ **for all** article a_i in week $w_j + 1$ and $w_j + 2$ **do**
$\quad\quad s\prime_i =$ ESTIMATE(a)
$\quad\quad s_i = Impact(a_i)$ from gold scores
$\quad$ **end for**
$\quad$ Log performance metric for $\vec{s}, \vec{s}'$
$\quad$ Reset $\vec{s}, \vec{s}'$
$\quad$ **for all** article $a_i \in w_j$ **do**
$\quad\quad$ ADDTRAININGDATA(a_i)
$\quad$ **end for**
end for

6.1.2. Online Evaluator. For each estimation method tested, a simulated real time iteration is executed, providing us with an evaluation mechanism. Initially, the forecasters are trained with a month of news data. Then, in following iterations another week of news data is introduced. In each iteration, the forecasters are retrained, and they are asked to forecast the impact scores of the following two weeks of articles. Then, their predictive accuracy is compared against the gold standard impact values and recorded.

Figure 6 shows the evaluation method for a single estimator. It should be noted that the *train* method has access to only *partial* impact scores, calculated by running the algorithm in Figure 5 with only the training data provided. The outer loop is run in parallel for all the estimators tested.

A sample estimator implementation is shown figure 7. This random estimator is used as one of the baseline methods.

6.1.3. Similarity Metrics. The articles were first converted into a representation of mixed term vec-

Table 1. Evolution of the highest ranked ImpactRank nodes.

01/14 Initial: average term impact	01/14	01/15	01/16	01/17	01/18
LOC/Deep Blue Highway	ORG/Congress	ORG/Congress	ORG/Congress	ORG/Congress	ORG/Congress
MSC/Safe Port Act	republican	republican	nation	$\boxed{\text{troop}}$	republican
shipping	scandal	nation	$\boxed{\text{troop}}$	republican	senate
commerce	LOC/Washington	LOC/Washington	senate	$\boxed{\text{increase}}$	$\boxed{\text{increase}}$
MSC/Year To Keep	force	force	American	nation	nation
90s	senate	$\boxed{\text{troop}}$	LOC/Washington	senate	$\boxed{\text{troop}}$
forth	nation	senate	republican	American	leader
masterwork	Saddam Hussein	leader	$\boxed{\text{increase}}$	leader	American
ORG/Democrats	complete	American	force	rush	LOC/Washington

Figure 7. Example Baseline Estimator: Random

procedure ADDTRAININGDATA(article a)
  nothing
end procedure
procedure TRAIN
  nothing
end procedure
function ESTIMATE(article a)
  $Estimate \leftarrow Random_0^3$
end function

Figure 8. Second Baseline Estimator: Average

procedure ADDTRAININGDATA(article a)
  Add a to set A
end procedure
procedure TRAIN(partial impact scores $\overline{impact}$)
  $Average \leftarrow \sum_{a \in A} \overline{impact_a}/|A|$
end procedure
function ESTIMATE(article a)
  $Estimate \leftarrow Average$
end function

tors, which go through basic cleanup, and then enhanced with the inclusion of recognized named entities. As a similarity metric between two vectors, cosine similarity[15] was used with an experimentally determined threshold of 0.80.

6.1.4. Evaluation Metric. We employed a mean square error based evaluation metric. This allows penalizing higher losses, while minimizing the impact of small variations. Since the system was evaluated empirically, residual sum of squares divided by the number of articles were calculated according to the Equation (10) as an approximation of the actual MSE values.

$$mse \approx \sum_{a \in A} (Impact(a) - Estimate(a))^2/|A| \quad (10)$$

6.2. Experiment Data

Our experimental data was extracted from The New York Times (NYT) Annotated Corpus[18]. The corpus itself consists of over 1.8 million articles from January of 1987 to June 2007, for a total of 20.5 years.

In our experiments, we used the news corpus for the first 6 months of 2007.

In order to focus on events, articles were filtered using the category information found within the NYT corpus. This allowed us to remove irrelevant categories, such as "Obituaries" and "Reviews".

The resulting evaluation dataset consisted of 16,852 articles.

6.3. Baseline Methods

The first choice as a baseline is random score generation, which was presented in Figure 7 earlier. This method randomly assigns scores in the [0,3] range to any article, without ever looking at the training dataset.

The second, and stronger baseline is based on using the average score of the articles seen so far for all newer articles. As presented in Figure 8, the algorithm uses the partial impact scores for calculating this average. It should be noted that since those scores are not generated with access to the entire dataset, partial scores might significantly vary from the actual ones.

6.4. Software Setup

The evaluation system utilized Stanford NER library[19] for named entity recognition and sentence boundary detection, Apache Cassandra[1] server for caching and data storage, LibSVM[13] for Support Vector Machine implementation, AT&T graphviz

1. http://cassandra.apache.org

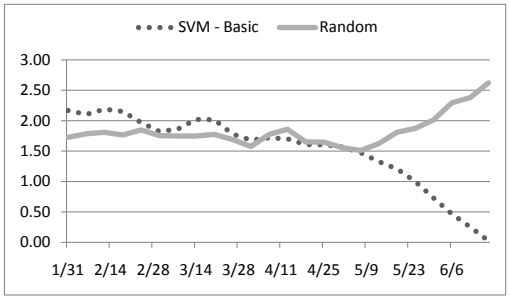

Figure 9. Comparison of the MSE values for **SVM-Basic** and **Random** estimators

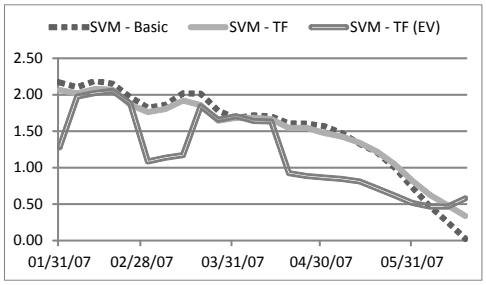

Figure 10. Comparison of the MSE values for Term Frequency based methods

software[20] for visualization, and Silverlight Framework[2] for web access.

7. Experiment Results

The first set of experiments conducted proved the viability of the SVM based method, by comparing its mean average error performance against the Random baseline algorithm. As seen on Figure 9, the **SVM-Basic** method overtakes **Random** with four months of training data.

After determining the viability of the **SVM-Basic** estimator, we evaluated the performance of the various feature selection and ranking schemes. Figure 10 shows the term vector only, and term vector plus named entity recognition based features, demonstrating

2. http://www.silverlight.net

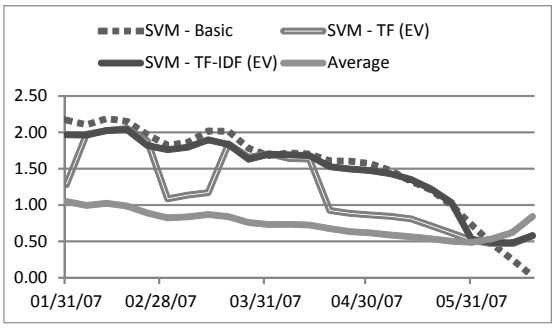

Figure 11. Comparison of MSE values for **SVM-TF (EV)**, **SVM-TF-IDF (EV)** features, and the **Average** baseline

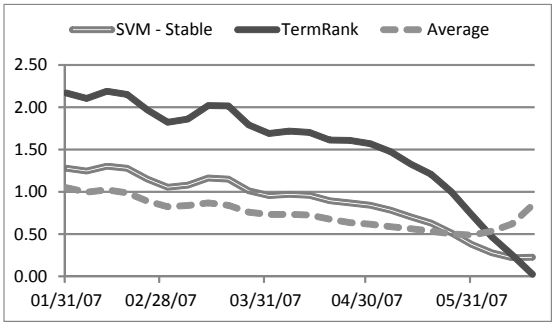

Figure 12. Comparison of MSE values for final set of estimators

the benefit of including named entities in the feature vectors. Figure 7 includes **SVM-TF-IDF (EV)** and **Average** (baseline) against the previously mentioned methods, showing the need for 5 months of training data for SVM to properly surpass the performance of the simple moving **Average** based baseline, and the poor performance of the tf and tf-idf based feature selection and ranking methods against the same baseline. Final set of evaluation among the simple moving **Average** based baseline, **SVM-Stable** and **TermRank** methods are presented in Figure 12. These results confirm our previous intuition about the appropriateness of the inverse-of-the-variance as a stable feature ranker. **TermRank** based feature ranking methods can outperform all other rankers in the longer run - within 6 months of training data. It also performs on the same level of the **SVM-Basic** with all features, with significantly lesser number of features.

8. Conclusion

In this paper we developed a framework and a measure for news impact forecasting. We proved the viability of our impact forecasting approach using a SVM based forecaster on six months of NYT corpus subset - consisting of 16,852 articles. We reported the results of our experiments with six different feature selection and ranking algorithms alongside two baseline methods, as well as results of a new ImpactRank technique. In our future work we plan to experiment with longer time-frames at multiple granularities of training and forecasting (i.e. yearly, monthly, weekly, daily, hourly etc.) and work with multiple news sources.

References

[1] Y. Yang, J. Carbonell, R. Brown, T. Pierce, B. Archibald, and X. Liu, "Learning approaches for detecting and tracking news events," *Intelligent Systems and their Applications, IEEE*, vol. 14, no. 4, pp. 32 –43, 1999.

[2] C. L. Giles, K. D. Bollacker, and S. Lawrence, "Citeseer: an automatic citation indexing system," in *DL '98: Proceedings of the third ACM conference on Digital libraries*. New York, NY, USA: ACM, 1998, pp. 89–98.

[3] C. Bergstrom, "Eigenfactor: Measuring the value and prestige of scholarly journals," *C&RL News*, vol. 68, no. 5, 2007. [Online]. Available: http://www.ala.org/ala/mgrps/divs/acrl/publications/crlnews/backissues2007/may07/eigenfactor.cfm

[4] E. Garfield, "The impact factor," *Current Contents*, vol. 34, no. 25, pp. 3–7, 1994.

[5] F. Hecht, B. K. Hecht, and A. A. Sandberg, "The journal 'impact factor': A misnamed, misleading, misused measure," *Cancer Genetics and Cytogenetics*, vol. 104, no. 2, pp. 77 – 81, 1998.

[6] L. L. Lange, "The impact factor as a phantom: Is there a self-fulfilling prophecy effect of impact?" *Journal of Documentation*, 2002.

[7] B. D. Ogden TL, "The ups and downs of journal impact factors," *The Annals of occupational hygiene*, 2008.

[8] ArnetMiner, "Arnetminer conference rank," online, 2008. [Online]. Available: http://www.arnetminer.org/page/conference-rank/html/Databases.html

[9] J. E. Hirsch, "An index to quantify an individual's scientific research output," in *Proc. Natl Acad. Sci. USA*, Nov. 2005, pp. 16 569–16 572, published as Proc. Natl Acad. Sci. USA, volume 102, number 46.

[10] J. Allan, J. Carbonell, G. Doddington, J. Yamron, Y. Yang, and U. Amherst, "Topic detection and tracking pilot study final report," Jul. 28 1998. [Online]. Available: http://citeseer.ist.psu.edu/425405.html;http://www.nist.gov/speech/publications/darpa98/ps/tdt2040.ps

[11] E. M. Voorhees, "Overview of TREC 2002," in *TREC*, 2002. [Online]. Available: http://trec.nist.gov/pubs/trec11/papers/OVERVIEW.11.pdf

[12] T. Joachims, "Text categorization with support vector machines: Learning with many relevant features," Universität Dortmund, Dortmund, Germany, Tech. Rep. LS VIII-Report, 1997.

[13] C. C. Chang and C. J. Lin, *LIBSVM: a library for support vector machines*. Online, 2001. [Online]. Available: http://www.csie.ntu.edu.tw/~cjlin/libsvm

[14] F. Gelgi, H. Davulcu, and S. Vadrevu, "Term ranking for clustering web search results," in *WebDB*, 2007. [Online]. Available: http://gemo.futurs.inria.fr/events/WebDB2007/Papers/p52.pdf

[15] G. Salton and C. Buckley, "Term-weighting approaches in automatic text retrieval," in *INFORMATION PROCESSING AND MANAGEMENT*, 1988, pp. 513–523.

[16] L. Page, S. Brin, R. Motwani, and T. Winograd, "The pagerank citation ranking: Bringing order to the web," Stanford Digital Library Technologies Project, Tech. Rep., 1998. [Online]. Available: citeseer.ist.psu.edu/page98pagerank.html

[17] M. Gori and A. Pucci, "Itemrank: A random-walk based scoring algorithm for recommender engines," in *IJCAI*, 2007, pp. 2766–2771.

[18] E. Sandhaus, "The new york times annotated corpus," Linguistic Data Consortium, Philadelphia, 2008. [Online]. Available: http://www.ldc.upenn.edu/Catalog/CatalogEntry.jsp?catalogId=LDC2008T19

[19] J. R. Finkel, T. Grenager, and C. D. Manning, "Incorporating non-local information into information extraction systems by gibbs sampling," in *ACL*. The Association for Computer Linguistics, 2005. [Online]. Available: http://acl.ldc.upenn.edu/P/P05/P05-1045.pdf

[20] AT&T, "Graphviz," open Source Graph Drawing Software. [Online]. Available: http://www.research.att.com/sw/tools/graphviz/