Intelligent Information Triage.

Conference Paper · January 2001

Source: DBLP

5 authors, including:

Intelligent Information Triage

Sofus A. Macskassy[†]
Haym Hirsh[†‡]
[†]Department of Computer Science
Rutgers University
110 Frelinghuysen Rd
Piscataway, NJ 08854-8019

{sofmac,hirsh}@cs.rutgers.edu

Foster Provost[‡]
Ramesh Sankaranarayanan[‡]
Vasant Dhar[‡]
[‡]Information Systems Department
NYU Stern School of Business
44 W. 4th St
New York, NY 10012

{fprovost,rsankara,vdhar}@stern.nyu.edu

Abstract

In many applications, large volumes of time-sensitive textual information require triage: rapid, approximate prioritization for subsequent action. In this paper, we explore the use of *prospective* indications of the importance of a time-sensitive document, for the purpose of producing better document filtering or ranking. By prospective, we mean importance that could be assessed by actions that occur in the future. For example, a news story may be assessed (retrospectively) as being important, based on events that occurred after the story appeared, such as a stock price plummeting or the issuance of many follow-up stories. If a system could anticipate (prospectively) such occurrences, it could provide a timely indication of importance. Clearly, perfect prescience is impossible. However, sometimes there is sufficient correlation between the content of an information item and the events that occur subsequently. We describe a process for creating and evaluating approximate information-triage procedures that are based on prospective indications. Unlike many information-retrieval applications for which document labeling is a laborious, manual process, for many prospective criteria it is possible to build very large, labeled, training corpora automatically. Such corpora can be used to train text classification procedures that will predict the (prospective) importance of each document. This paper illustrates the process with two case studies, demonstrating the ability to predict whether a news story will be followed by many, very similar news stories, and also whether the stock price of one or more companies associated with a news story will move significantly following the appearance of that story. We conclude by discussing how the comprehensibility of the learned classifiers can be critical to success.

1. INTRODUCTION

Professionals receive increasing amounts of information, some of which is time sensitive and is important for them to consider. Business news provides an interesting illustration: the job performance of financial analysts, attorneys,

business-school professors, market makers, portfolio managers, reporters, and many others would benefit from timely attention to certain business news stories. Bloomberg, Reuters, Bridge, and several other companies have profited greatly selling a variety of instant-access, business information services. However, the volume of business news is so large that few professionals can pay attention to it all, let alone do so in a timely fashion.

Information triage is the monitoring of one or more information sources to provide users with well-filtered, prioritized, and/or categorized information (*cf.*, [18]). Our general information-triage framework consists of monitoring a potentially wide range of online information sources—such as news stories, stock data, weather reports, and other computer-based information feeds—and evaluating each item to assess its importance to a given user. Although information triage does not require multiple sources of information, we explicitly embrace such situations when they can create synergy and improve the information-triage process.

One of the key difficulties in building information-triage procedures is building models of importance that will be used to prioritize information. Ideally we would like to obtain from a user a direct statement of his or her interests. However, in many cases it is not clear that users can do so effectively. Instead information filtering and ranking procedures often rely on indirect statements of interest, such as user-provided, keyword-based profiles [14, 12], or samples of information items whose importance has been assessed by the user via relevance feedback methods [25, 26, 28]. We believe that such methods are crucial components of an effective information triage procedure, but we believe that there are other useful components as well. In this paper we concentrate on prospective indicators.

Often what makes information important is some subsequent occurrence that is directly or indirectly associated with the information. For example, consider the appearance of a news story about a publicly traded company, after which the company's stock value quickly plummets. The importance of the news story is based not solely on the story itself but also on the occurrence of the future event (observed in a separate information feed, in this case stock-market data). In many cases a user will be able to specify what future events would make an information item important—such as a substantial change in the value of a company.

A problem is that this importance criterion can not be evaluated directly at the time the information appears; its

evaluation requires knowledge about the future. However, if there are patterns in the stories—for example, if many stories that are coupled with precipitous drops in a stock price have similar structure or content—we might be able to predict (approximately) that a story will be followed by an important event. As we will show, even an approximate prediction can be quite useful for information triage.

We propose to have a user specify what would make an article important if, in fact, we could perceive the future behavior of this or other relevant information feeds. We then *operationalize* [21] this importance criterion to be evaluable on a given story in the given information feed before we see the future. Key to our approach is the application of the user's specification to label historical documents. These data then form a training corpus, to which inductive algorithms will be applied to build a text classifier. Although we believe this framework to be complementary to learning from labels elicited via relevance feedback (or other manually created labels), it has the advantage that the labeling can be done automatically, and at a very large scale.

This paper describes a four-step process for creating and evaluating such operationalized approximations to a user's non-operational specification.

1. Elicit from the user and encode a specification of what future events would make a current piece of information interesting—for example, a news story would be interesting if, within the hour of the story being published, there is a significant/unusual move in the price of the stock of any company associated with the news story.

2. Use this specification to analyze information feeds received in the past to ascertain whether or not each item was interesting, thereby creating a set of data items labeled by whether or not each was interesting.

3. Apply inductive algorithms to these labeled data to form models that can estimate the extent to which an information item is interesting to a user directly from the item itself, without the need to look into the future.

4. Analyze the learned model to assess both whether it appears to be a plausible operationalization of the original criterion and whether it is something that appears trustworthy. If the "native" learned form is not easily interpretable, as is the case in the two studies contained herein, then this may first require applying techniques for obtaining an understandable form of the operationalized criterion.

After providing further details of this process, the paper focuses on two case studies involving two available information feeds, news stories and stock price data. In the first case study we deem a news story important if there are a significant number of subsequent stories that appear similar to it. In the second, a news story is deemed important if in the hour following its appearance the stock-market return of a company associated with the story is more than one standard deviation from its normal hourly return.

2. LEARNING OPERATIONAL INFORMATION FILTERS

We now describe in more detail the four-step process for performing one form of information triage: when an item is deemed interesting because something important subsequently happens—something that can be measured objectively (retrospectively) either in the given information feed or some coupled information feed. This process reifies and extends the process used previously by Fawcett and Provost [11] and by Lavrenko, *et al.* [17], where text documents (news stories) were labeled by referencing subsequent stock-market events.

2.1 Specifying a Non-Operational Criterion

Our first step is to acquire and encode the specification of how an item may be interesting based on possible events that may be observed subsequently. In general this can be a complex process. The non-operational criterion is non-operational only with respect to a world where no knowledge of the future is available. However, it needs to be fully operational when it does have access to the future, as is the case when it is being used to label data from the past. Thus, for example, saying that an article is interesting if it is followed by a substantial movement in a stock lacks important detail. If movement is defined in terms of what is typical, it is necessary to quantify what is typical, as well as to specify how far a value must be from typical. The first step of our process requires that the criterion be stated in unambiguous detail, so it can be applied directly to data.

2.2 Generate and Label Data

The specification of a non-operational criterion will generally presume that a particular primary information feed is the focus of the criterion, and that the criterion will look into the future of either this feed or other coupled feeds to assess the interestingness of items obtained on the primary feed. We thus must access data from these historical feeds. Once available, we can use the non-operational criterion to label elements of this feed for use in the next step of our process. In some cases the criterion will focus solely on the future of the primary feed—such as if a story is interesting because a large number of follow-up stories are subsequently observed—or it may require access to one or more secondary feeds, such as stock price data.

Once the data have been obtained and transformed into suitable form, they can be labeled using the interestingness criterion. This is performed in a straight-forward fashion. For each information item in the data generated for the primary data feed, pretend that it has just appeared. The items that follow it chronologically represent the future that is about to follow the given item. Given access to the item, as well as the other information items that followed it (the item's "future"), it becomes possible to use the user's importance criterion to assess this item. The result is a corpus of information items from the past, each labeled by whether it is deemed important according to the user's criterion.

2.3 Applying Machine Learning

Once the data have been labeled, it is now possible to apply machine learning algorithms to them. Note that all knowledge of the future is embodied in the label associated with each item. The learned model therefore can examine the item—with no information about the future—and can make a prediction about what the non-operational criterion will yield on that item. In other words, the learned model is an operationalized (albeit perhaps approximate) form of

the importance criterion that can be used directly on items obtained from the information feed.

The selection of a learning method depends heavily on the nature of the information feeds. If each item is a collection of numerical values (*i.e.*, attribute/value data), learning methods suitable for such data would be used. In many cases—including those considered in the remainder of this paper—each information item is a text object, and thus text classification methods can be used to form the operationalized importance criterion. The accuracy of any such learning method will be affected by the extent to which the contents of each information item provide clues to what the non-operational criterion may predict. Without at least some correlation of this sort, the operationalization process should perform no better than random prediction. An assessment of the extent to which such correlations exist will usually take place at this stage.

This assessment is affected by the fact that the data are temporal in nature. In particular, any estimates of the expected predictive accuracy of a learned model must be made on data that appeared later in time than the training data. Cross-validation methods are thus not appropriate for use in this context—evaluation must instead guarantee that all test examples appeared chronologically later than all training examples.

2.4 Analysis

Learning a model is only one part of the overall goal of using this framework. Also important is an analysis of the learned model, to gain insight into what actually was learned. This is important for two reasons. First and foremost, an analysis can be used to evaluate whether the learned model actually has learned the criterion and does not reflect less-meaningful artifacts present in the data. Second, it can be used to gain insight into how the criterion works and can be used to *explain* what is happening in the model. The final step of our framework thus consists of analyzing the learned model, both with respect to how well it appears to match up with our intuitions about what the non-operational criterion was encoding, as well as simply with respect to whether it appears sufficiently plausible that the user would be willing to place some trust in it.

Performing such an analysis will depend substantially on the form of the learned model. A number of researchers have used machine-learning methods to extract interpretable models from difficult-to-understand models, such as complicated expert systems [8], neural networks [7], and ensemble classifiers [9]. We also must do so similarly in our case studies here. We use our learned difficult-to-understand models to relabel the given data, thereby forming a corpus that reflects the performance of the learned model (rather than reflecting the original labels). We then use the Ripper rule-learning system [5, 6] to learn, from these relabeled data, a (more) interpretable approximation (explanation) of the original learned model. As we will see, even these approximate models have limited interpretability for our domain experts, leading us to a further stage of analysis.

3. CASE STUDY I: HOT STORY DETECTION

For our first case study we focus on news stories from a set of business wires. Our goal is to recognize stories that are "hot", in the sense that more similar stories follow them than is typically the case. Although similar to the "on-line new event prediction" [4, 34] or "first story detection" [2] problem within the Topic Detection and Tracking (TDT) initiative [1, 30, 31, 33, 32], the problem we are addressing differs in two important ways. First, we do not require a story to be the very first on a topic, but rather that there are more than a normal amount of subsequent stories that are more similar to the story than is typical. Second, it demonstrates a different approach to these sorts of problems than is typically taken in such work. Rather than requiring a human to manually label stories according to one of a fixed number of known topics, we instead elicit and encode a non-operational criterion that can then be used to label arbitrary amounts of past data.

For this case study, we label a story as interesting by focusing on all stories that appeared within the subsequent 24-hour period and that was associated with any of the companies also associated with the current story.[1] If this collection of stories has an average distance that differs substantially from a story's normal distance to subsequent stories, the initial story is deemed interesting. Similarity is based on the cosine of the TFIDF-vectors of the different news stories [27]. The next section discusses this in further detail.

3.1 Specifying the Non-Operational Criterion

Consider the set of news stories associated with a given stock symbol α. Listed chronologically we get a series of stories $S_\alpha = \{s_1, s_2, \ldots, s_i, \ldots\}$. We can step through each story s_i in S_α and assess the extent to which the stories s_{i+j} in S_α for $j = 1, \ldots, t_i$ are similar to s_i, where t_i is the number of stories in S_α that follow s_i within 24 hours. If we use $\text{sim}(s_i, s_j)$ to designate the similarity of two stories using the cosine of the TFIDF vectors of the two stories [27], we can identify this raw total similarity of story s_i, $\text{TSIM}(s_i)$ by adding the similarity of story s_i to all stories that follow it:

$$\text{TSIM}(s_i) = \sum_{j=1}^{t_i} \text{sim}(s_i, s_{i+j}) \qquad (1)$$

This raw measure is unduly influenced by the number of α stories occurring in the following 24-hour period. In order to address this, we compute the average *per-story* similarity value for s_i, $\text{SIM}(s_i)$, by dividing $\text{TSIM}(s_i)$ by t_i, the number of α articles following s_i within 24 hours:

$$\text{SIM}(s_i) = \frac{\text{TSIM}(s_i)}{t_i} \qquad (2)$$

To determine whether this quantity is sufficiently far from normal we first need to define normal. We do this by computing the mean SIM_α and standard deviation ρ_α across all stories s_i in S_α. We then compute t, the average number of stories t_i that follow a story in a 24–hour period, to take into account the number of similar stories (to make sure we don't give unnaturally high scores to stories with only a few nearby, albeit high-scoring stories). We then assign a score

---

[1]While the 24-hour time frame was chosen somewhat arbitrarily, its arbitrariness is exactly in keeping with the spirit of our approach, in which the user provides an intuitively plausible criterion that depends on the future and it is up to the system to do its best with what it's given.
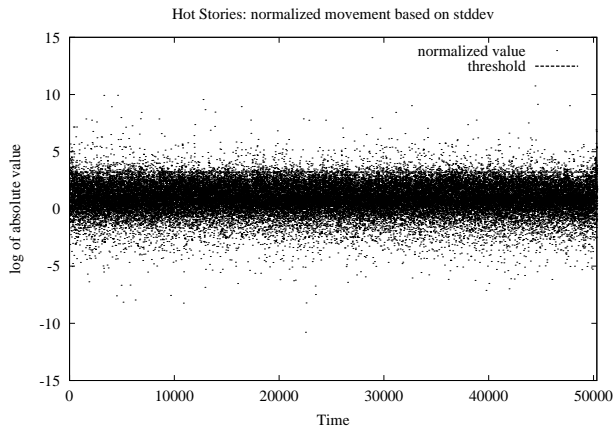
Figure 1: dist scores for news stories

score(s_i) to each s_i as

$$\text{score}(s_i) = \frac{\text{TSIM}(s_i)}{t} \tag{3}$$

Finally, a story is labeled by how many standard deviations, dist, it falls from score(s_i):

$$\text{dist}(s_i) = \frac{\text{score}(s_i) - \text{SIM}_\alpha}{\rho_\alpha} \tag{4}$$

Since multiple companies can be associated with a story s_i, we label s_i using the maximum of all dist scores for all its associated stock symbols. The values observed in our data range from -15 to $10,000$. Figure 1 shows a scatter-plot of the log of the absolute value of these, where the scores are listed left-to-right in chronological order. Final labels were assigned by labeling each s_i as "important" if its dist score was greater than 25.[2]

3.2 Generating and Labeling Data

For this case study, we consider news stories from a set of public newswires (including Business Wire, Canada NewsWire, CCN Disclosure, Internet Wire, PR Newswire, PrimeZone, and Reuters). Each story averages roughly 400–500 words, and each has been analyzed to extract its complete date stamp as well as the stock symbols of all the companies associated with that news story. For the purpose of our experiments, we used a more manageable-sized subset of these data, limited to 50,158 stories that appeared between January 5, 1999 and September 14, 1999, where stories with incomplete time stamps (both date and time of day), duplicate stories, and stories associated with more than eight companies (typical for stories that discuss the market in general rather than a particular stock or segment) were removed. We then applied the non-operational criterion specified in the previous subsection to these data. Using 25 as the cutoff for the density resulted in 1723 of the stories being labeled as important.

3.3 Applying Machine Learning

Given data labeled with our non-operational criterion, we can then proceed to the learning step. To evaluate how well

[2]25 was chosen in a fairly *ad hoc* fashion, by finding a split that limited the number of positively labeled examples.

learning performs we run our learning methods on a per-day basis. For each day we use as training data all stories that appeared before it, skipping all data that appeared earlier in the same day or in the immediately preceding day. (The reason for imposing a gap was to minimize the risk that learning will perform well due to occasional duplicate stories—stories with different headers but identical bodies, something that rarely happens if stories are more than a day apart.)[3]

The criterion we use labels only a small number of stories as important. If testing begins too early in the historical data feed, there is the chance that there may be few or no relevant examples of the minority class to learn from. In order for the learner to have sufficient training data, our evaluations thus begin at the chronological date where at least half of the "important" stories will be in the training set. This left 26,461 stories serving as test data, with 876 of them being labeled "important".

To evaluate the ability of a learning method to form the approximate operationalization of the importance criterion we present our results using ROC curves. ROC analysis is an evaluation technique used in signal detection theory, which has seen increasing use for other types of diagnostic, machine-learning, and information-retrieval systems [29, 23, 22]. ROC graphs plot false-positive rates on the x-axis, and true-positive rates on the y-axis. ROC curves are generated in a similar fashion to precision/recall curves, by varying a threshold across the output range of a scoring model, and observing the corresponding classification performances [24]. Although ROC curves are isomorphic to precision/recall curves, they have the added benefits that they are insensitive to changes in marginal class distribution, and that the area under the ROC curve has a well-defined statistical meaning [13].[4]

Although we used a range of standard text categorization algorithms, all performed roughly comparably. Due to their relatively quick run times we therefore only report on results using the Naive Bayes [10] and TFIDF [27] classification methods.[5] Naive Bayes estimates the *a posteriori* probability that an example belongs to a class given the observed feature values of the example, assuming the independence of the features given the class label. The TFIDF method [27, 16, 28] is based on Rocchio's [25] relevance feedback algorithm. A prototype vector is formed for each class from the examples of that class. A new document can then be compared to each prototype by computing the cosine of the prototype vector with the new document vector, and final scores for each class can be assigned by normalizing the cosine-distance values.

Figure 2 shows the resulting ROC curve for the two methods as well as the method that selects at random between the choices. It shows that, regardless of what ultimately is the appropriate trade-off between false positives and false negatives, it appears that there is sufficient information solely in the news stories themselves to be able to be able to do substantially better than random. Whether this prediction

[3]We also performed experiments without imposing the one-day gap, and observed little effect on the performance of the learned model.
[4]DET curves [19, 3] are used in a similar fashion in the TDT initiative [1, 2, 3, 4, 17, 30, 31, 33, 32], and are isomorphic to ROC curves, differing primarily on rescaling the axes.
[5]We used the versions of these learners found in the publicly available Rainbow package [20].
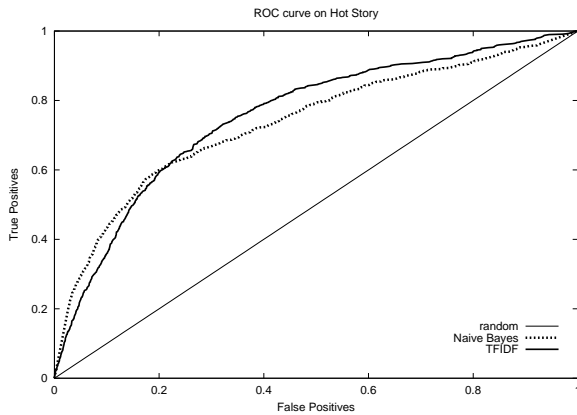
Figure 2: ROC curve for hot story detection

statements net reuters loss
statements press ended current form future risks quarter
statements ended research pm
statements ended pm research
statements release act announces differ
statements press act uncertainties contact include
statements ended pm receivable
statements press ended current form
taxes ended reuters
statements ended risk commission
statements ended loss nasdaq
statements release act announces
statements ended press equipment
statements release announces form
statements release act research
statements release contact made process
statements press act uncertainties

Table 1: Ripper rules for hot story detection

is good enough depends, of course, on how it will be used. Different users have different spans of attention and different needs. The ROC curves show that if the stories were to be ranked solely by this single estimation of importance (which more generally would be a component of a greater definition of interestingness), the top of the ranking would be substantially denser with important stories than would the bottom of the ranking. Any user restricted to examining only a subset of the stories would examine considerably more important stories.

To be specific, each day there are on average about 270 stories (using our corpus, which contains a subset of all the business news), and about 9 of them will be important by the current definition. Without ranking, if a user selected (randomly) 30 stories, 1 would be important. With ranking (using TFIDF), if a user selected the top 30 stories, 7 would be important: an increase in precision of 600%. Of course, the relative increase is smaller as one works down the list, but remains impressive. Without ranking, if a user selected (randomly) 100 stories, 3–4 would be important. With ranking, to get 4 important stories the user would only need to inspect the 17 top-rated stories. Conversely, if the user selected the top 100 ranked stories, 19 would be important; more than 400% improvement in precision.

Of course, some users may have to read all the news stories (often subject to other filtering criteria). It seems initially that such users would not benefit from such triage. However, this conclusion ignores the issue of timeliness. At any point, a user will have a queue of news stories pending examination. A triage system would maintain a priority queue of news stories, and even users who eventually must read all the stories may benefit in terms of timeliness of information: important stories are more likely to be inserted higher in the priority queue.

3.4   Analysis

As mentioned previously, it is also important to understand the result of the learning process. The original criterion is specified with respect to future, as yet unseen, information, but its learned form only refers to information present in the given information item. It is important for a user to have confidence that the operationalized criterion matches—even if only in part—the intentions of the original non-operational criterion.

If the learning methods generated interpretable results,

it may be possible to inspect the results directly to understand what aspects of an information item are correlated with the non-operationalized criterion. However, there is no guarantee that interpretable methods will actually be used in practice, for example if the learning method that yields interpretable results runs slowly or is less accurate. Our experiments represent such a case, where we use relatively fast methods that combine scores on words in a holistic fashion, making it difficult to interpret how they behave. (Although we did try to provide explanations of our results by extracting words with high information gain, our domain expert did not feel that it gave him any insight into the results.)

To understand the results of the operationalization process better we approximate the learned classifier using a learning method whose output is more understandable. We step through a collection of data on a day-by-day basis, as was described in the previous section. Each day's data are labeled by the results of learning from the earlier days' data. As a result, on a day-by-day basis, we have the "compiled wisdom" of the learned model, as seen in how it labels the data to which it is applied. These labeled data can then be used as input to a learner that will give more interpretable results.

To demonstrate this approach to analysis we used it to understand the results of the TFIDF classifier. This was done using five steps:

1. For each day we learned a classifier to label that day's data using earlier data.

2. We extract the top 250 stemmed words from all the data using standard entropy-based measures.

3. For each news story, we remove words whose stem is not present in the top 250 words.

4. The resulting labeled data were then given to Ripper [5, 6], a learning system that forms rules, a representation that is perceived by many as being more understandable. It was run with varying Loss-Ratios, ranging from 0.05 through 2.50 in increments of 0.05 in order to get a broad spectrum of rule sets.

5. In order to get rules that represent the learned model well, a pruning step was then applied. A rule was pruned if it had low total coverage (*i.e.*, when applied

in isolation), a low percentage of true positives to its total coverage, or if it was generated by few Ripper runs. In this study, a rule was pruned if it had a coverage of less than 100 stories, had a TP ratio of less than 75%, or if it appeared in fewer than 5 Ripper runs.

Table 1 shows the 17 rules that were produced by our analysis of the data labeled by the original learned model. Ripper learns patterns that common sense suggests could be correlated with the appearance of new stories. More remarkably, however, we find that many of the words found in these rules can be found in the disclaimer at the end of a story or press release—words such as "statements", "information", "contact", "release", "act", and "differ" seem to commonly occur in disclaimers found in the stories matched by these rules. Our ongoing work continues this analysis, to go beyond the qualitative results we've thus far obtained to be able to quantify the extent to which the presence of disclaimers in stories provides a helpful clue to the importance of hot stories.

4. CASE STUDY II: STOCK MOVEMENT

In our second case study we consider a problem that correlates news stories with stock price movement. We began with a problem that has been studied by others [11, 17], labeling a story as interesting if the stock price of any company associated with this news story changes in a way prespecified as being interesting. Rather than inheriting from prior work a definition of an interesting change, to evaluate our four-step approach we "started from scratch", going to an expert on financial information systems to obtain his proposed non-operational criterion for this concept.

4.1 Specifying the Non-Operational Criterion

This case study goes beyond the first case study in one important respect, using a secondary information feed as the basis for assessing the interestingness of a news story. Unlike the previous case, the non-operational criterion can be stated more crisply. For each company's stock, we compute the mean and standard deviation of its one-hour return (relative change in price). We then label a story as "important" if the return of any stock associated with the story in the hour after the story appeared was more than one standard deviation from the norm. Note that this means that stories whose stock dropped as well as stories whose stock rose were included as being "important."

4.2 Generating and Labeling Data

For this case study we continue with the news-story source used in the first case study, but add a stock price news feed. For our experiments we use the same 50,158 used in the first case study. We further take only those that appeared during normal trading hours (excluding those appearing in the final hour of trading), as well as stories that were only associated with one company, leaving 33,326 stories. Thus we remove data that are guaranteed to have no change in stock-price values since normal trading has ceased, as well as stories that have a higher probability of being important solely on the fact that they are associated with multiple companies.[6]

---

[6]We do not consider the many important news stories that appear "after the bell," focusing here only on stories for which we have trading data.
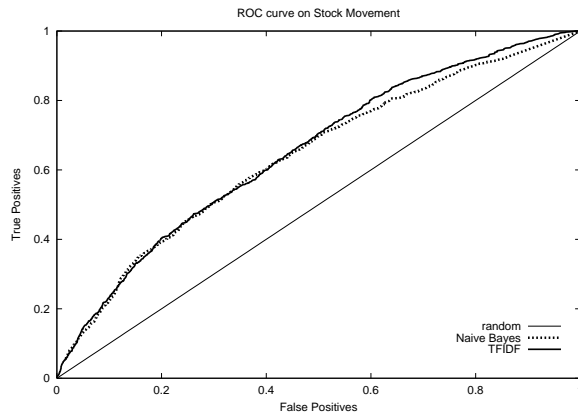

Figure 3: ROC curve for stock movement

The second source of information is trade-level data for over 8000 publically traded companies on the NYSE, AMEX and NASDAQ exchanges. We have data collected since January, 1993, and we use this full data set to calculate the one-hour mean and standard deviation for each stock. Due to the enormous amount of data, we aggregate into 5-minute intervals—for each stock we maintain its price at both the start and end of the interval, as well as its trading volume during that interval. At this point we can apply the non-operational criterion to the data obtained between January 5, 1999 and September 14, 1999 (the dates of the selected stories), resulting in 2615 of the stories being labeled as important.

4.3 Applying Machine Learning

To perform the learning stage of our operationalization approach we followed the procedures laid out in our first case study. We run through the data a day at a time. During training we skip all data from earlier that day and all of the previous data in any evaluations that are performed. We begin evaluations at the chronological date where at least half of the "important" stories will be in the training set. This meant that only 18,165 instances were being tested with 1263 of those being "important."

These results use the same set of learners as above, Naive Bayes and TFIDF. The resulting ROC curves are shown in Figure 3. Here again it shows that, regardless of what ultimately is the appropriate trade-off between false positives and false negatives, it appears that there is sufficient information in the two information sources to be able to predict considerably better than random. Further, as with the previous experiment, any user restricted to examining only a subset of the stories would examine considerably more important stories. To be specific, each day there are on average about 239 stories in our final data set, and about 17 of them will be important by the current definition. Without ranking, if a user selected (randomly) 14 stories, 1 would be important. With ranking (using the TFIDF method), if a user selected the top 14 stories, 3 would be important (all appeared in top 10): an increase in precision of 200%. Of course, the relative increase is smaller as one works down the list, but remains impressive. Without ranking, if a user selected (randomly) 100 stories, 7 would be important. With ranking, to get 7 important stories the user would only need to inspect the top 27 stories. Conversely, if the user selected

share net ended note
statements share net
share net ended average
statements release stock shares
results uncertainties directors
share quarter net
alert nyse
results uncertainties act chief
share record directors
statements actual annual pm markets
statements release stock prnewswire
share net
results statements approximately
results actual approximately
results actual chief risk
statements release stock
results statements chief
results statements future act

Table 2: Ripper rules for stock movement

the top 100 ranked stories, 18 would be important, more than 150% improvement in precision.

4.4 Analysis

Since this problem also concerns news stories and uses the same suite of learning methods, we use the same methodology for evaluating the results of learning as we did in our first case study. Table 2 shows the 18 rules generated in our analysis step.

To obtain insight into our original learned models we would like to go beyond these rules, to understand if there is a more general phenomenon underlying the words in these rules. To answer this question we exploit the existence of a taxonomy from the accounting literature that can be used to label each story with one or more categories from a list of 12 categories [15]. We further expanded this list into 21 categories, which are shown in Table 3.

This made it possible for us to expand on our earlier analysis in light of these categories:

1. We manually label a random sample of the stories into one of the 21 categories. We can use this sample to compare the distribution of the randomly selected stories to the stories that were labeled important by the user importance criterion. These two distributions should be quite different, hopefully pulling out categories that correlate with interesting stories.

2. Select prototypical rules that appeared to have significance based on their coverage of the examples and percentage of true positives.

3. Manually label all *true positive* stories covered by these rules and compare their distributions to the distribution of random stories that were truly important.[7]

We used this analysis here to focus on two prototypical rules that appeared to have some significance in terms of the words within them. In each case we hand-labeled each story that the rule matched with all of the categories that appeared to apply to it.

---
[7]The rules are used to gain insight into the original learned model, and thus looking at false positives would focus on cases that are not part of that model.

Code	Description
PR	Product related
JV	Joint ventures
CMM	Capital Market/Macroeconomeny related*
FA	Forecast/Analysis
NC	Not classifiable*
MR	Management related
EA	Earnings announcements
ACQ	Acquisitions
OTH	Other regulatory and legal actions*
COP	Company operations related*
CAP	Capital/ownership changes
DVD	Dividend announcements*
ASS	Asset changes
MER	Mergers
DIV	Divestiture
LAB	Labor-related*
SPI	Spinoffs
FIN	Financial distress*
DEM	De-merger
ACC	Accounting/corporate*
INC	Income-tax related*

* Our additions to the original 12 categories [15]

Table 3: Category taxonomy used for story analysis

In the case of the first rule we selected, results uncertainties directors ⟶ interesting, our hypothesis was that this rule covers largely analyses and projections about the future as well as earnings announcements and management changes. Indeed, the majority of the matched stories were from these three categories, although a few concerned product-related issues as can be seen in the accompanying distributions (Figure 4). In the case of our second rule, share net ended average ⟶ interesting, our conjecture was that these stories were earnings announcements. Indeed, almost all were so as can be seen in the distributions (Figure 5).

Why are these results interesting? First, the classifier is remarkably accurate at associating stories with standard accounting categories. The results suggest that it may be feasible to automate the labeling task with high accuracy instead of requiring humans in the loop to conduct the laborious task of labeling stories.
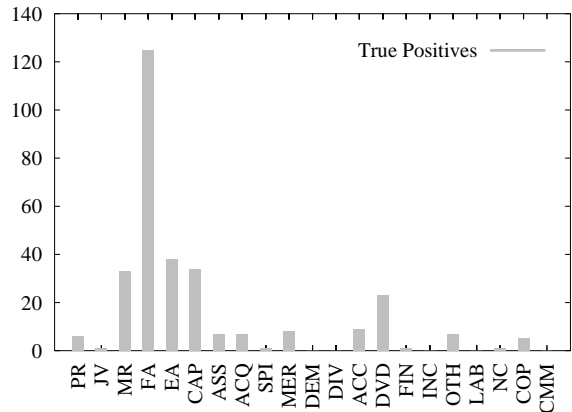

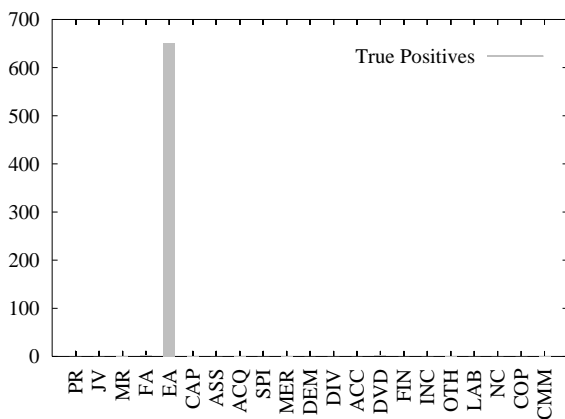**Figure 4: Distribution for "results uncertainties directors"**

Figure 5: Distribution for "share net ended average"

Figure 6: Distribution for "important" and "not important" stories

Second, it may seem odd that the classifier picks up on themes such as earnings announcements as being important. Is it the case that all earnings announcements are important? In this case, the classifier would be essentially be filtering important versus non-important categories. Or is it identifying a subset of earnings announcements that truly are important? To answer this question we looked at the distributions of important versus non-important stories in general, shown in Figure 6. It is interesting that the distributions are not dramatically different, with the major differences occurring in the "not classifiable" (15.8% important versus 10.2% non-important) and product related categories (22.8% important versus 17.8% non-important), with minor differences in the other major categories. This tells us that important stories are *not* dominated by specific categories. In effect, judging by the words in the rule, the classifier is identifying a subset of stories from earnings announcements that are important.

5. FINAL REMARKS

This paper introduced a four-step process for identifying information items that may be important based on their correlation with the occurrence of subsequent events. The paper further presented two case studies of this approach concerning news stories—recognizing "hot stories" that have many similar stories following them, and recognizing stories that are associated with a stock that will have a significant movement in value.

While the first steps of our process are fairly well understood, we have only begun the final step of the analysis to get a better understanding of the resulting models. The analysis presented in this paper presents us with a good set of human-understandable rules that give us a sense of plausibility for the learned models, but still leaves something to be desired with respect to actually being able to *explain* and understand the final model or to gain any insight into what makes the criterion effective. We are currently working on more elaborate techniques to discern the underlying rules and correlations, to get a better understanding of the domain and criteria presented in this paper.
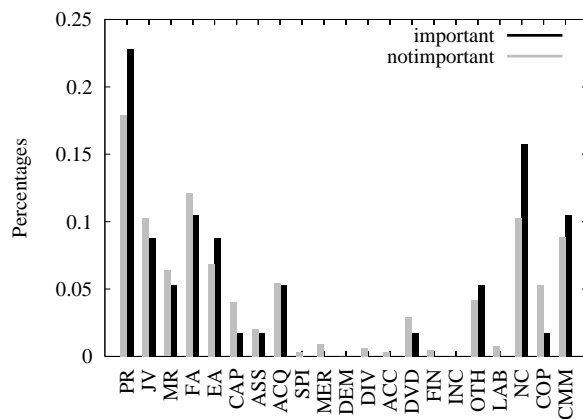
6. REFERENCES

[1] J. Allan, J. Carbonell, G. Doddington, J. Yamron, and Y. Yang. Topic detection and tracking pilot study: Final report. In *Proceedings of the Broadcast News Understanding and Transcription Workshop*, pages 194–218, 1998.

[2] J. Allan, V. Lavrenko, and H. Jin. First story detection in TDT is hard. In *Proceedings of the Ninth International Conference on Information and Knowledge Management*, pages 374–381, 2000.

[3] J. Allan, V. Lavrenko, and R. Papka. Event tracking. CIIR Technical Report IR-128, University of Massachusetts Computer Science Department, 1998.

[4] J. Allan, R. Papka, and V. Lavrenko. On-line new event detection and tracking. In *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 1998.

[5] W. W. Cohen. Fast effective rule induction. In *Proceedings of the Twelfth International Conference on Machine Learning*, 1995.

[6] W. W. Cohen. Learning trees and rules with set–valued features. In *Proceedings of the National Conference on Artificial Intelligence*, 1996.

[7] M. W. Craven and J. W. Shavlik. Extracting tree-structured representations of trained networks. In *Advances in Neural Information Processing Systems*, pages 24–30, 1996.

[8] A. Danyluk and F. Provost. Small disjuncts in action: Learning to diagnose errors in the telephone network local loop. In *Proceedings of the Tenth International Conference on Machine Learning*, 1993.

[9] P. Domingos. Knowledge acquisition from examples via multiple models. In *Proceedings of the Fourteenth*

International Conference on Machine Learning, pages 98–106, 1997.

[10] P. Domingos and M. Pazzani. Beyond independence: Conditions for the optimality of the simple Bayesian classifier. In *Proceedings of the 13th International Conference on Machine Learning*, pages 105–112, 1996.

[11] T. Fawcett and F. Provost. Activity monitoring: Noticing interesting changes in behavior. In *Proceedings of the Fifth International Conference on Knowledge Discovery and Data Mining*, 1999.

[12] P. W. Foltz and S. T. Dumais. Personalized information delivery: An analysis of information filtering methods. *Communications of the ACM*, 35(12):51–60, Dec. 1992.

[13] D. J. Hand. *Construction and Assessment of Classification Rules*. Chichester:John Wiley and Sons, 1997.

[14] E. M. Houseman and D. E. Kaskela. State of the art of selective dissemination of information. *IEEE Transactions on Engineering Writing and Speech*, 13(2):78–83, 1970.

[15] R. B. T. II, C. Olsen, and J. R. Dietrich. Attributes of news about firms: An analysis of firm-specific new reported in the *wall street journal index*. *Journal of Accounting Research*, 25(2), 1987.

[16] T. Joachims. A probabilistic analysis of the Rocchio algorithm with TFIDF for text categorization. In *Proceedings of the Fourteenth International Conference on Machine Learning*, 1997.

[17] V. Lavrenko, M. Schmill, D. Lawrie, P. Ogilvie, D. Jensen, and J. Allan. Language models for financial news recommendation. In *Proceedings of the Ninth International Conference on Information and Knowledge Management*, pages 389–396, 2000.

[18] C. Marshall and F. Shipman. Spatial hypertext and the practice of information triage. In *Proceedings of the '97 ACM Conference on Hypertext*, pages 124–133, Apr 1997.

[19] A. Martin, G. Doddington, T. Kamm, , M. Ordowski, and M. Przybocki. The DET curve in assessment of detection task performance. In *Proceedings EuroSpeech*, volume 4, pages 1895–1898, 1997.

[20] A. K. McCallum. Bow: A toolkit for statistical language modeling, text retrieval, classification and clustering. `http://www.cs.cmu.edu/∼mccallum/bow`, 1996.

[21] D. J. Mostow. Machine transformation of advice into a heuristic search procedure. In *Machine Learning: An Artificial Intelligence Approach*, pages 367–403. Morgan Kaufmann, 1983.

[22] K.-B. Ng and P. Kantor. Predicting the effectiveness of naive data fusion on the basis of system characteristics. *Journal of American Society for Information Science*, 51(13):1177–1189, 2000.

[23] F. Provost and T. Fawcett. Analysis and visualization of classifier performance: Comparison under imprecise class and cost distributions. In *Proceedings of the Third International Conference on Knowledge Discovery and Data Mining*, pages 445–453, 1997.

[24] F. Provost and T. Fawcett. Robust classification for imprecise environments. *Machine Learning*, 42:203–231, 2001.

[25] J. Rocchio. Relevance feedback in information retrieval. In Salton, editor, *The SMART Retrieval System: Experiments in Automatic Document Processing*, chapter 14, pages 313–323. Prentice–Hall, 1971.

[26] G. Salton and C. Buckley. Improving retrieval performance by relevance feedback. *Journal of the American Society for Information Science*, 41:288–297, 1990.

[27] G. Salton and M. J. McGill. *Introduction to Modern Information Retrieval*. McGraw-Hill, Inc., 1983.

[28] R. Schapire, Y. Singer, and A. Singhal. Boosting and Rocchio applied to text filtering. In *Proceedings of ACM SIGIR*, pages 215–223, 1998.

[29] J. Swets. Measuring the accuracy of diagnostic systems. *Science*, 240:1285–1293, 1988.

[30] F. Walls, H. Jin, S. Sista, and R. Schwartz. Probabilistic models for topic detection and tracking. In *IEEE International Conference On Acoustics, Speech and Signal Processing*, 1999.

[31] J. P. Yamron, L. Gillick, S. Knecht, S. Lowe, and P. van Mulbregt. Statistical models for tracking and detection. In *Working notes of the DARPA TDT-3 Workshop*, 2000.

[32] Y. Yang, T. Ault, T. Pierce, and C. W. Lattimer. Improving text categorization methods for event tracking. In *Proceedings of ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 65–72, 2000.

[33] Y. Yang, J. G. Carbonell, R. D. Brown, T. Pierce, B. T. Archibald, and X. Liu. Learning approaches for detecting and tracking news events. *IEEE Intelligent Systems*, 14(4):32–43, 1999.

[34] Y. Yang, T. Pierce, and J. G. Carbonell. A study on retrospective and on-line event detection. In *ACM SIGIR Conference on Research and Development in Information Retrieval*, 1998.