IDEA GROUP PUBLISHING
701 E. Chocolate Avenue, Suite 200, Hershey PA 17033, USA
Tel: 717/533-8845; Fax 717/533-8661; URL-http://www.idea-group.com

ITP4608

Speedup Learning for Text Categorization and Intelligent Agents

Jeffrey L. Goldberg, Ph.D. and Matthew L. Jenkins
Analytic Services Inc. (ANSER)
1000 Green River Drive, Suite 202
Fairmont WV 26554
(304) 534-5332, (304) 534-5368
Jeffrey.Goldberg@anser.org, Matthew.Jenkins@anser.org

ABSTRACT

Research in text categorization has been focused on off-line machine learning algorithms: a predetermined set of categories is learned prior to the operation of the system in which they are to be applied. Also, the learning from examples paradigm requires a training session in which a teacher, rather than the end-user, manually labels the training set of example documents. This is labor intensive, particularly for rare categories: and essentially all categories on the Internet are rare. We propose the use of a speedup-learning algorithm in which a user interacts directly with the machine learning algorithm, and thereby greatly reduces the amount of training documents that must be labeled for optimal performance of the system. It also places the training capability directly into the hands of end-users, which opens up new applications, e.g. to track breaking news events on the Internet. Other researchers have previously identified the speedup learning strategy; we extend the concept, implement an algorithm, and apply it to Intelligent Internet Agents and law enforcement.

ORGANIZATION

This paper is organized into eight sections: (1) Introduction; (2) Background, reviewing some relevant technologies; (3) Speedup learning: the algorithm; (4) FasTrac: the algorithm plus the user-interface; (5) Intelligent Agents: How might FacTrac technology be applied in Intelligent Agents (6) Future Work; (7) Conclusions; and (8) References.

1. INTRODUCTION

A lot of research has been done on text categorization. Text categorization is defined as the labeling of documents in accordance with previously specified categories. Several machine learning algorithms have been applied to learn text categorizers automatically from a set of positive and negative example documents. The text categorizers are then used to label new documents. [1]There are two problems with the above approach:

- A teacher, separate from the end-user, must exhaustively label the entire training set. As a rule of thumb, we attempt to maintain at least 50 positive examples in the training set for any category. For rare, or infrequent categories, categories that occur naturally less than ½ % of the time, this requires the user to label more than 10,000 documents.
- The algorithm is a priori. Two phases, separated in time, proceed in sequence as follows:
  - o The training phase: a pre-determined set of categories is learned.
  - o The performance phase: the text categorizers from phase one are plugged into and run as the classifier function in the end-user application.

But, to refine the performance of the categorizers themselves, the end-user is required to augment the training set of documents with additional examples, or to modify the training set with corrected examples, and then start the process from the beginning. It is non-incremental and non-adaptive in the sense that it cannot learn new categories chosen by the end-user at runtime.

2. BACKGROUND

Naïve Bayes Algorithm

Bayesian learning is relevant for our work in speedup learning, because when applied to text categorization, its output is of the form: $P(C_i|D_j)$; or the probability that a category C_i should be assigned given the occurrence of words in a document D_j. Naïve Bayes is a practical Bayesian learning algorithm that has been shown to be particularly effective when applied to natural language documents. The output probabilities are used to produce a ranking of documents according to how "like" they are to the set of positive examples from the training set.

The naïveness is due to its treatment of the features, in our case word roots or stems, and the assumption that they occur mutually independently in documents, which is what permits Naïve Bayes to be computationally feasible. This assumption is obviously false, for example the word "learning" is more likely to be found when the prior word is "machine" than it is at random. However, Naïve Bayes has been proven one of the best for text categorization.

The basic form of Naïve Bayes used by the speedup algorithm is:
$P(C_i=1|D_j) = P(C_i)P_k\ P(F_k|C_i)$

The formula reads as follows: the probability that a category occurs $C_i=1$ given the evidence in document D_j, equals the product of the prior probability of the category and the product of the conditional probabilities of the occurrence of the features F_k in document D_j given the occurrence of category C_i. In other words: the problem of determining the probability that a category should be assigned is reduced to the product of the conditional probabilities that its features occur given the category. We have made some optimizations to Naïve Bayes for the problem of speedup learning that increase its performance by a few percent. The optimizations are related to higher weighting of the features of recently added examples to the increasing training set, because they are more likely to be near the borderline of the true optimal classifier function.

Speedup Learning

Lewis first introduced the strategy of a speedup learning algorithm for text categorization [2]. It is based on the notion that a machine learning algorithm learns the most from examples that are closest to the border that divides the positive from the negative examples in the training set. The algorithm knows what documents it needs the user to label at each stage of operation, those that are nearest the border defined by the classifier function, and measured by probabilities falling closest to 0.5.

Upon each iteration the current classifier function is applied to the entire unlabeled training set (step [iii] in Figure 3). The output of the Naïve Bayes algorithm is used to produce a ranking. The ranking is ordered by max uncertainty, i.e. closeness to the probability 0.5. The K most uncertain documents are removed from the unlabeled training set and the user is asked to identify the positives, or the documents that match the users criteria for membership in the category. The user can train an optimal categorizer for any category; it need not even correspond to a label, and can be totally abstract. The only necessity is that the user "recognizes" when a document should be included in the category.

3. THE SPEEDUP ALGORITHM

For each desired category:

[i] Index the unlabeled set of documents producing the vocabulary for learning.

[ii] Have the user create an initial test collection by moving N positive documents from the test collection into a folder and any uncovered negatives into another folder(s)

Repeat steps [iii] – [v] until a termination condition is reached: either M total positive documents have been placed in the positive folder; or the accuracy exceeds the performance threshold, L%

[iii] Apply the "current" classifier function to the entire "Unlabeled Set" to produce a ranking ordered by "max uncertainty".
Remove the top K most uncertain documents and put them into the "Candidate Set". The user will now be asked to label these documents.

[iv] Remove from the "Candidate Set" the documents the user identifies as positives, put them into the "Positives" folder of the "Test Collection". Remove the remaining documents from the candidate set, and place them into the "Negatives" folder.

[v] Train a new classifier function based on the "Test Collection"; at the same time use cross validation to produce performance statistics. This is done by dividing the "Test Collection" into two parts; a "Training Set" to produce the classifier; and a "Test Set" to produce the performance statistics. By using the current classifier to predict the labels of the documents in the "Test Set" an estimate of the performance on new/unseen documents can be determined.

Initial suggestions for the parameters M, and N are: M = 200; K = 20; and N = 5, and L is category dependant.

The performance of the current classifier is measured after each iteration and compared to several termination conditions: either the overall performance objective is met, or the maximum number of iterations the user is willing to execute is met.

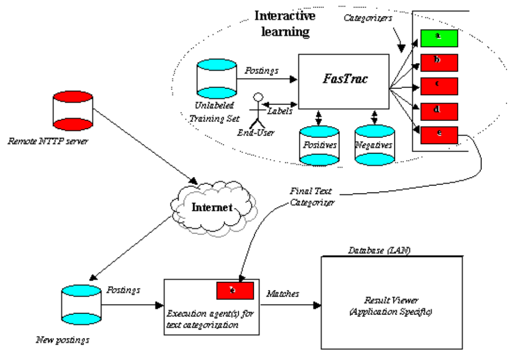
Performance

The performance of the speedup algorithm on two categories related to terrorism is shown in Table 1 below. After 10 iterations the user has labeled 200 documents and the performance is already reasonable, around 90%. After 20 iterations the performance of the speedup algorithm is very high, around 95%. The performance of the Naïve Bayes algorithm on the entire training set was about this level. It is of concern that the performance of the speedup algorithm actually continued to increase beyond the "optimal performance." We are doing more extensive testing to determine what caused this unexpected result, the speedup algorithm is supposed to converge to optimal performance more quickly than the straightforward learning from examples approach, but it is not expected to exceed the performance on the fully labeled training set. Despite this, the effectiveness of the speedup algorithm is clear from the comparison to the algorithm that selects the candidate set documents randomly, rather than by max uncertainty.

Table 1: Performance of the Speedup Learning Algorithm

Iteration	**Random's** Errors = fp + fn	**Random's** Breakeven Performance	**Speedup's** Errors = fp + fn	**Speedup's** Breakeven Performance
1	565	71.03	576	70.8
2	460	76.63	467	76.0
3	352	80.16	408	78.9
4	533	72.11	372	80.5
5	467	74.17	314	83.4
10	656	63.37	194	89.2
15	447	73.57	138	91.8
20	281	82.34	81	94.9
25	257	82.76	42	97.2
30	205	85.43	46	96.9

Figure 1: The Architecture of An Intelligent Agent using Text Categorization via FasTrac



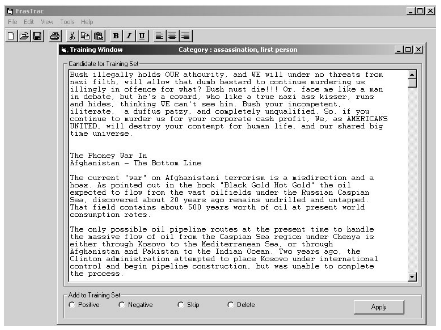
4. FASTRAC

FasTrac is the user-trainable, reusable software module that uses the speedup learning algorithm for text categorization. Its dataflow diagram appears below in Figure 3: Appendix A. The steps of the algorithm were given in section 3. The algorithm is the smart one in this model of learning, as it knows what documents it wants the user to label at every step, and the user just does what it is told to do: provide the labels for the documents in the candidate set. The point of the speedup learning is twofold: to learn quickly, and to facilitate user-trainable categories.

In the upper right hand corner of Figure 1, the interactive learning component, FasTrac is shown. This produces the classifier functions, or text categorizers. For each desired category, the user produces an optimal text categorizer by interactively labeling the "Candidate Sets" of documents, extracted from the "Unlabeled Set" of documents by the speedup algorithm according to "maximal uncertainty." The output is the optimal (final) text categorizer for the category. Currently, the algorithms being applied for use with speedup learning are Naïve Bayes [3], Support Vector Machines (SVM) [4]. The optimal categorizer is then used by execution agents to recognize new documents.

In Figure 2, FasTrac's Candidate Document Window is shown. A document has been selected by FasTrac for labeling by the user according to the desired category. The user may choose the options in the panel at the bottom of the window labeled "Add to Training Set" to indicate the document is either an example of the category, by choosing "positive", or not, by choosing "negative". Alternatively, if the user wishes he or she may defer judgement on the document by choosing "skip", which returns it to the "Unlabeled Set"; or may remove the document from the system by choosing "delete."

Figure 2: FasTrac's User Interface



Training Window Category : assassination, first person

Candidate for Training Set

```
Bush illegally holds OUR athourity, and WE will under no threats from
nazi filth, will allow that dumb bastard to continue murdering us
illingly in offence for what? Bush must die!!! Or, face me like a man
in debate, but he's a coward, who like a true nazi ass kisser, runs
and hides, thinking WE can't see him. Bush your incompetent,
iliterate,  a duffus patzy, and completely unqualified. So, if you
continue to murder us for your corporate cash profit, We, as AMERICANS
UNITED, will destroy your contempt for human life, and our shared big
time universe.


The Phoney War In
Afghanistan - The Bottom Line

The current "war" on Afghanistani terrorism is a misdirection and a
hoax. As pointed out in the book "Black Gold Hot Gold" the oil
expected to flow from the vast oilfields under the Russian Caspian
Sea, discovered about 20 years ago remains undrilled and untapped.
That field contains about 500 years worth of oil at present world
consumption rates.

The only possible oil pipeline routes at the present time to handle
the massive flow of oil from the Caspian Sea region under Chenya is
either through Kosovo to the Mediterranean Sea, or through
Afghanistan and Pakistan to the Indian Ocean. Two years ago, the
Clinton administration attempted to place Kosovo under international
control and begin pipeline construction, but was unable to complete
the process.
```

Add to Training Set

○ Positive ○ Negative ○ Skip ○ Delete

5. INTELLIGENT AGENTS WITH TEXT CATEGORIZATION

The initial application for FasTrac is to allow an Intelligent Internet Agent to be trained to track categories of immediate topical interest. We have several prototype Intelligent Agents in various stages of development: Newshound, Chathound, and Webhound. Newshound is most highly developed, is in actual use by law enforcement, and so we'll now describe its capabilities. Newshound, and the two agents in development, are described in more detail in a separate publication [5].

Newshound Requirements

The original purpose of Newshound is to look for specified, trainable content in Usenet newsgroups. By specified and trainable, we mean that given a set of (positive and negative) example postings, it must be able to discern a classifier function and find new postings that are 'like' the positive examples. Newshound is to operate as an intelligent agent. It must allow a human agent to specify the parameters of operation, including the news server, the newsgroups in which to look, and the categories of what to be on the look out for. After having the parameters of operation selected, the intelligent agent must then operate autonomously, only requiring interaction whenever the user desires to check the results of what has been matched so far, or to change the parameters of operation. Once a Newshound agent has found postings that match its category(s), the human agent instructs the Newshound agent as to which of the results are correct and which are not. This last requirement is called user-feedback and retraining and allows the originally learned text categorizers to be refined and personalized.

Newshound Implementation

Newshound is an intelligent Internet agent that recognizes postings of interest to a human user from Usenet newsgroups. It uses text categorization technology to train a classifier function for each desired category based on a set of examples. The classifiers, or text categorizers, are then used to recognize documents (Usenet postings), which are like the positive examples. It has been employed in a Pilot Program with an organization of the federal government and is being operationally tested by Special Agents.

Applying FasTrac to Newshound

Since text categorization has already been applied to Newshound, it already has an interface that allows a text categorizer to be plugged-in. The Intelligent Agent may now be used to recognize breaking events by simply archiving an adequate number of newspostings from Usenet news and running FasTrac. The user then can use FasTrac to train Newshound to retrieve new documents of the desired category. Since Usenet News has a high daily throughput, the amount of time to archive the required "Unlabeled Set" of documents required by FasTrac should be limited to hours or days for many typical categories. Applying Newshound with FasTrac to track categories related to breaking terrorist events would be of interest, but it remains to be seen how quickly useful Newshound Agents can be trained, and how well they will work.

6. FUTURE WORK

The infrastructure is in place, but the hard work has just begun. Requiring the end-user to be the trainer of an Intelligent Agent places a burden on both the user and the user-interface. If the user interface is too complicated for the prospective end-users to achieve acceptable performance; then the speedup algorithm is moot. Getting the right combination of technology and simplicity for use by non-technical, or at least non-programming end-users will require great skill beyond the realm of programming. All of this assumes that the Intelligent Agent technology is flawless, and that FasTrac is flawless, but they are currently prototypes and are therefore not flawless. Obviously, the user cannot be expected to tolerate difficulties with robustness in such a highly interactive, and dynamic system.

How to best apply the technologies of Newshound and FasTrac to monitor Usenet news will also be challenging. One question is whether it may be possible to apply text categorization to detect "first person" categories. These are categories where a perpetrator incriminates his or herself directly. And whether correlations can be found between threats made on the Internet and actual classes of criminal events or attacks? Also, how much "third person" information can be found in Usenet newsgroups or the rest of the Internet that is of possible interest in intelligence in the hours or days immediately following breaking events?

And then, how can Newshound be combined with ANSER's other protocol agents, e.g. Webhound, and Chathound, to gather information on the same topic? How to answer questions like these using the technology of text categorization and Intelligent Internet Agents will be very challenging.

7. CONCLUSIONS

Speedup has been shown to work, it increases the rate of learning in text categorization by roughly one order of magnitude or greater. Unlike a pure learning from examples paradigm, it allows the training to be done by the end-user. It opens up new possible applications for text categorization, and some are highly relevant to the new war on terrorism..

8. REFERENCES

[1] J.L. Goldberg, "CDM: An Approach to Learning in Text Categorization," *International Journal on Artificial Intelligence Tools,* Vol 5, Nos. 1 & 2, pp. 229-253, July 1996.

[2] D.D. Lewis, "A Sequential Algorithm for Training Classifiers," In Proceedings of SIGIR'94 the 17th ACM International Conference on Research and Development in Information Retrieval, pp. 3-12, July 1994.

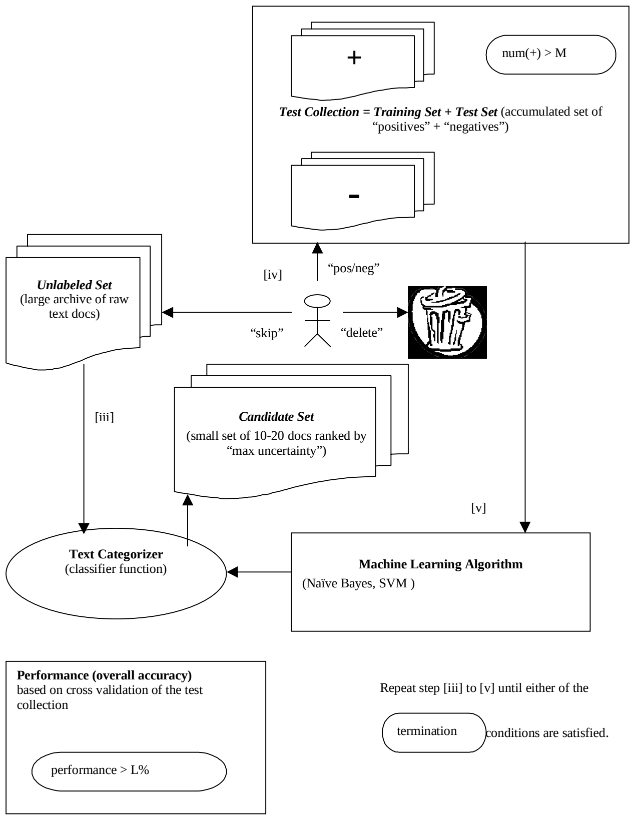
[3] David D. Lewis, "Naive (Bayes) at forty: The independence assumption in information retrieval," In *European Conference on Machine Learning*, 1998.

[4] N. Cristianini and J. Shawe-Taylor, *An Introduction to Support Vector Machines (and other kernel-based learning methods).* Cambridge University Press, 1999.

[5] J.L. Goldberg, and S.S. Shen, "Newshound Revisited: The Intelligent Agent that Retrieves News Postings," In V. Sugumaran, editor, *Intelligent Support Systems*, Idea Group Publishing, Hershey PA, In Press for Spring 2002.

APPENDIX A

Figure 3: Speedup Data Flow Diagram


+
num(+) > M
Test Collection = Training Set + Test Set (accumulated set of "positives" + "negatives")
-
[iv]
"pos/neg"
Unlabeled Set (large archive of raw text docs)
"skip"
"delete"
[iii]
Candidate Set (small set of 10-20 docs ranked by "max uncertainty")
[v]
Text Categorizer (classifier function)
Machine Learning Algorithm (Naïve Bayes, SVM)
Performance (overall accuracy) based on cross validation of the test collection
performance > L%
Repeat step [iii] to [v] until either of the termination conditions are satisfied.