**Title**: BoosTexter: A Boosting-based System for Text Categorization

**Authors:** ROBERT E. SCHAPIRE, YORAM SINGER

**Year:** 2000

**Introduction:**

* Boosting refers to combining many inaccurate categorizations into a highly accurate categorization rule. The rules are trained sequentially such that each rule is trained on the examples which were most difficult to classify by the preceding rules
* Uses two extensions of AdaBoost.
  + In the first extension, the goal of the classifier is to predict all and only all of the correct labels. Thus, the learned classifier is evaluated in terms of its ability to predict a good approximation of the set of labels associated with a given document.
  + In the second extension, to design a classifier that ranks the labels so that the correct labels will receive the highest ranks.

**Boostexter Description:**

* The boosting algorithm finds a set of weak hypotheses by calling the weak learner repeatedly in a series of rounds. These weak hypotheses are then combined into a single rule called the final or combined hypothesis.
* As boosting progresses, training examples and their corresponding labels that are hard to predict correctly get incrementally higher weights while examples and labels that are easy to classify get lower weights. For instance, for the news classification problem, it might be easy to classify a document as a news item but hard to determine whether or not it belongs to the finance section. Then, as boosting progresses the weight of the label News for that document decreases while the weight of Finance increases. The intended effect is to force the weak learning algorithm to concentrate on examples and labels that will be most beneficial to the overall goal of finding a highly accurate classification rule.

**Evaluation Metrics:**

* **One-error.** This measure evaluates how many times the top-ranked label was not in the set of possible labels. Thus, if the goal of a multiclass system is to assign a single label to a document, the one-error measures how many times the predicted label was not in Y. It measures the probability of not getting even one of the labels correct. The one-error of a hypothesis f is denoted by one-err( f ). A classifier can be defined as H : X → Y that assigns a single label for a document x by setting H(x) = argmax ∈Y f (x, y).
* **Coverage.** While the one-error evaluates the performance of a system for the top-ranked label, the goal of the coverage measure is to assess the performance of a system for all the possible labels of documents. That is, coverage measures how far it is needed, on the average, to go down the list of labels in order to cover all the possible labels assigned to a document. Coverage is loosely related to precision at the level of perfect recall.
* **Average precision.** The above measures are not complete for multi-label classification problems: good (low) coverage can be achieved but suffer high one-error rates, and vice versa. In order to assess the label ranking of a multiclass system as a whole the non-interpolated average precision is used, a performance measure frequently used for evaluation of information retrieval (IR) systems (Salton, 1991). Note, however, that non-interpolated average precision is typically used in IR systems to evaluate the document ranking performance for query retrieval. In contrast, in Boostexter experiments average precision is used for evaluating the effectiveness of the label rankings. This measure evaluates the average fraction of labels ranked above a particular label y ∈ Yi which actually are in Yi . Note that avgprecS( f ) = 1 for a system f which ranks perfectly the labels for all documents so that there is no document xi for which a label not in Yi is ranked higher than a label in Yi.

**Datasets:**

* **Modified Apte (“ModApte”) split.**  Contains 12,902 documents. A cleaned-up version of this dataset, called Reuters-21578, is publicly available from the web page http://www.research.att.com/∼lewis by David Lewis, who originally compiled the collection. Following pre-processing are performed prior to the experiments: All words were converted to lower case, punctuation marks were removed, and “function words” from a standard stop-list were removed. The average length of a document after pre-processing is 82 words. This corpus is divided into categories which in turn are sub-divided into sub-categories. The Reuters corpus has served as the benchmark for many text-categorization studies using various partitions of the corpus. 3-fold cross validation is used in experiments with these partitions. To compare Boostexter to previously published work, experiments were performed with a partition that includes all topics in Reuters that have at least two relevant documents for training. This collection includes 93 topics and was studied extensively by Yang (to appear) and others.
* **AP Titles.** This is a corpus of AP newswire headlines (Lewis & Gale, 1994; Lewis & Catlett, 1994). As for the Reuters corpus, previous work concentrated on binary classification by tagging documents as being relevant or irrelevant to topics like “federal budget” and “Nielsens ratings.” The total number of documents in this corpus is 319,463. The headlines are an average of nine words long, with a total vocabulary of 67,331 words. No preprocessing of the text was done, other than to convert all words to lower case and remove punctuation marks. Two sets of experiments were performed with this corpus based on two different labeling schemes available for this corpus.
* **UseNet data.** This dataset consists of Usenet articles collected by Lang (1995) from 20 different newsgroups. One thousand articles were collected for each newsgroup so there are 20,000 articles in the entire collection. This data was originally treated as single-labeled (see for instance Joachims (1997)). 4.5% of the articles are actually multi-labeled. 544 identical articles which were posted to more than one group. The total number of articles after relabeling the data based on the headers is 19,466 with 20,347 labels.