

Text Categorization Models for Retrieval of High Quality Articles in Internal Medicine

Y. Aphinyanaphongs, M.S., C.F. Aliferis M.D., Ph.D.

Department of Biomedical Informatics, Vanderbilt University, Nashville, TN

Abstract

The discipline of Evidence Based Medicine (EBM) studies formal and quasi-formal methods for identifying high quality medical information and abstracting it in useful forms so that patients receive the best customized care possible [1]. Current computer-based methods for finding high quality information in PubMed and similar bibliographic resources utilize search tools that employ preconstructed Boolean queries. These clinical queries are derived from a combined application of (a) user interviews, (b) ad-hoc manual document quality review, and (c) search over a constrained space of disjunctive Boolean queries. The present research explores the use of powerful text categorization (machine learning) methods to identify content-specific and high-quality PubMed articles. Our results show that models built with the proposed approach outperform the Boolean based PubMed clinical query filters in discriminatory power.

Introduction

Evidence Based medicine (EBM) is the clinical application of high-quality medical information. The application of EBM involves 3 distinct steps [2]: (1) the identification of high-quality evidence that pertains to a specific clinical question, (2) evaluation and synthesis of this evidence, and (3) application of the evidence to the problem. This paper addresses the question of how to identify high quality evidence.

One pioneering method to identify high quality articles is the use of the "clinical query filter" (CQF) for PubMed article retrieval. Introduced by Haynes et. al, the method involves the creation of boolean set terms which are used to filter and identify high quality articles in pre-specified content areas. These filters have been shown to have good performance [3] and are featured in the clinical queries link in PubMed [4]. This method requires manual selection of terms and relies on a brute-force learning approach using a non-standard and fairly restrictive classifier (term disjunctions of 4 to 5 terms).

The motivation of the present paper is to contribute to the practice of EBM by exploring methods to automatically construct quality and content filters for article retrieval. We hypothesize that using powerful text categorization techniques and a suitable article collection for training, we can

construct filters superior to the existing ones. Toward these goals, and as a first step, we explore computer models to retrieve high-quality, treatment-related articles in internal medicine.

Methods

1. Definitions

At the core of our efforts lies the selection of a rigorous quality and content gold standard as well as the creation of a document collection that captures this gold standard. Ideally this gold standard should be easy to obtain for large numbers of documents. For these reasons, we chose to use the selections of the editors and reviewers of the ACP journal club as our gold standard [5].

The ACP journal club is a highly-rated meta-publication. It includes no original research articles. Instead, every month experts review the best journals in internal medicine and select the best articles according to specific selection criteria in the article class areas of: *treatment*, *diagnosis*, *etiology*, *prognosis*, *quality improvement*, *clinical prediction guide*, and *economics*. Selected articles are further subdivided into articles that are summarized and abstracted by the ACP because of their clinical importance, and those that are only cited because they meet all the selection criteria but may not pertain to vitally important clinical areas. (In the present paper, the abstracted or cited articles are denoted as ACP+; all other articles not abstracted or cited as ACP-.) Every article is subjected to rigorous review for inclusion. For example, in the article class area of *treatment*, the basic criteria are a random allocation of participants to comparison groups, 80% follow-up of those entering the study, and the outcome to be of known or probable clinical importance [5].

For our first experiments, we chose the *treatment* class area. The ACP journal cites and abstracts this area the most, and a larger proportion of clinical questions are treatment-related [6]. In the discussion section we discuss extensions to all categories.

2. Corpus Construction

We downloaded from PubMed all original articles with abstracts from the journals reviewed by the ACP in the publication period of July 1998 through August 1999. Two conditions motivated this period of time. First, one year provides a large sample for the

treatment category. Second, selecting a period of several years before the start of the present study gave ample time for original articles to be reviewed by the ACP. The ACP journal typically takes several months to review and republish an article. Thus, to ensure that no ACP+ articles are missed, the ACP journal was reviewed from the beginning of the publication period, July 1998 to nearly 1.5 years after the end of the publication period, December 2000.

We identified 49 journals appearing in the review lists of the table of contents of the first ACP journal in July 1998 to the last ACP journal in December 2000. This set of journals thus is guaranteed to be the complete set of journals reviewed by the ACP.

All original articles were automatically downloaded with custom Python scripts using the limit option of the PubMed search interface. Each search was limited to the title of 1 of the 49 journals and set to only retrieve articles during the publication period. The “only items with abstracts” checkbox was marked to ensure that letters and other content were not included in the results. These articles were downloaded in XML format. A custom built XML parser extracted PubmedID, title, abstract, publication type, and MeSH terms. All article information was stored in a relational database (MySQL) [7].

Reviewing the ACP between July 1998 and December 2000 identified the high quality articles in the publication period of July 1998 to August 1999. Due to the unavailability of complete electronic versions of the ACP for these periods, all table of contents and cited article lists were scanned on a HP Scanjet C9850A and digitized using ABBY FineReader Pro optical character recognition (OCR) software [8]. OCR errors were manually identified and corrected. ACP articles were automatically matched with the titles of articles in the MySQL database and marked in the corpus. In addition, each article was marked as to the article class it belongs.

3. Corpus Preparation For Analysis

The corpus was divided into positive and negative classes. The positive class composed of 396 ACP+ articles in the treatment class. The negative class had 15407 total ACP- articles and ACP+ articles *not in the treatment class*. 20% of the corpus for both classes was left aside as a reserve in the event that we needed an unbiased sample for future analyses. Of the remaining 80%, we created 10 mutually exclusive sets for cross validation. Cross validation is crucial to identify models with good generalization error and estimate that error (i.e. prevent overfitting of the model to the training data) [9].

For each fold, after test set removal, the training set was further subdivided into a 70% train and a 30% validation set. The validation set was needed to

optimize any learning model parameters. The idea is to optimize the model parameters without using the test set since using the test set will likely overfit the learner to the test data [9]. We used maximization of area under the receiver operating curve (ROC) for parameter optimisation [10]. Thus, each fold has a train, validation, and test set with the proportions of each class in each set maintained. *Each set* in the fold is further processed as described in this algorithm:

For each article in this set

Extract mesh terms

precede all terms with 'mh_'

replace all punctuation with '_'

associate main headings with each

subheading ||i.e. Migraine:etiology and Migraine: therapy||

Extract publication types

precede all terms with 'pt_'

replace all punctuation with '_'

Concatenate abstract and title words

convert all words to lowercase

remove all punctuation and replace with ' '

remove MEDLINE stop words

Porter-stem all words

calculate weights for terms //see text for details

calculate raw frequency occurrence of terms

The information provided to the learning algorithms were words in the title and abstract, publication types, and MeSH terms. MeSH and publication types were not encoded as individual terms but instead as phrases. For example, the publication type *Randomized Controlled Trial* is encoded as a single entity.

Stop words are words such as: “the”, “a”, “other”, “each”, “other”, etc. that do not add semantic value to the classification. We used the same stop words that are excluded by the Pubmed search engine [11].

The porter stemming algorithm [12] [13] was used to reduce words to their base forms. Its use is motivated by the observation that word forms may not add additional value to the classification. For example, the terms “randomly”, “randomised”, and “randomisation” all describe similar processes and were reduced to “random” by stemming. Stemming increases the effective sample in this example by encoding a term 3 times rather than 3 terms just once.

Log frequency with redundancy [14] is a weighting scheme used to encode information about the usefulness of a term in making a classification. Intuitively, terms that appear in many articles (i.e. stop words) are not as useful in classifying articles as terms that appear in fewer documents. Many weighting schemes exist [15], but this scheme was

chosen due to its reported superior classification when using support vector machines [14].

The final step was to calculate the raw occurrence of terms in each article. Naïve bayes and the raw input to the boostexter algorithm used frequency rather than weighted terms. Boostexter and support vector machines used weighted terms as input [16].

4. Statistical and Machine Learning Methods

Due to space limitations a thorough review of all machine learning methods applied in the reported research is not possible. Such a review, of how machine learning applies to text categorization and information retrieval, can be found in [17]. In the present section we discuss which methods were applied and with what parameters:

4.1 Naïve Bayes

Naïve Bayes is a common machine learning method used in text categorization with excellent results. The Naïve Bayes classifier estimates the probabilities of a class c given the raw terms w by using the training data to estimate $P(w|c)$. The classification predicted by this classifier is determined by the max a-posteriori class [18].

We coded the algorithm in C as described in Mitchell 1997 [19]. No parameter optimization is necessary for Naïve Bayes.

4.2 Text-Specific Boosting

Another state of the art method for text categorization is boosting. The basic idea behind boosting is that many simple and moderately inaccurate classification rules can be combined into a single, highly accurate rule. The prototypical algorithm for boosting is termed Adaboost.

Adaboost has been applied successfully using various methods to generate the simpler rules. Wilbur uses naive bayes with good results for identifying articles about restriction enzymes [20]. Shapire and Singer boost decision trees with good results on several datasets [21]. Schapire and Singer further extend this work in Boostexter which implements a one level decision tree that evaluates the presence or absence of a word for each category [16]. Because the binaries were readily available, and there is evidence that boosting trees are as good or better than other boosting methods for text (Wilbur, personal communication), we use Boostexter in our initial modeling efforts.

Specifically, we use AdaBoost.MR implemented as part of Boostexter available from AT&T [22]. Parameter optimization was implemented for the number of iterations (i.e. number of simple rules to consider).

4.3 Support Vector Machines (SVMs)

Support vector machines (SVMs) can function as both linear and non-linear classifiers. [23-25].

For the text categorization task, the words are weighted and utilized as features for the linear and polynomial SVMs. We use the implementation of SVMs in Svm-Light [26]. For the linear SVM, we used misclassification costs of {0.1, 0.2, 0.4, 0.7, 0.9, 1, 5, 10, 20, 100, 1000}. For the polynomial SVM, we used the same misclassification costs *minus* {100, 1000} and polynomial degrees of {2, 3, 5, 8}. Combinations of both cost and degree were run on the validation set.

4.4 Clinical query filters

The clinical queries are Boolean queries optimized separately for sensitivity, specificity, and accuracy [3]. The exact queries used in Pubmed were applied to the categorization task. The Boolean query for the sensitivity filter for treatment is “therapeutic use [MeSH subheading] OR drug therapy [MeSH terms] OR randomized controlled trial [ptyp] OR random*[text word].” The Boolean query optimized for specificity for treatment is “placebo [text word] OR (double [text word] AND blind*.)”

Results and Discussion

1. Area under the curve analysis

The average AUCs for the 10 folds for each algorithm are presented in table 1.

Learner	Average AUC over 10 folds	Range over 10 folds	p-value compared to largest
LinSVM	0.965	0.948 – 0.978	0.01
PolySVM	0.976	0.970 – 0.983	1.0
Naïve Bayes	0.948	0.932 – 0.963	0.001
Boost Raw	0.957	0.928 – 0.969	0.001
Boost Wght	0.941	0.900 – 0.958	0.001

Table 1: Average AUC over 10 folds for each learning method.

The high values for the AUCs suggest that the learning methods can distinguish between the target classes. The polynomial SVM model with degree 8 and cost 0.1 has the best performance. We compared the mean of the polynomial model to the other means using the Wilcoxon rank sum test [27]. All p-values for the AUC compared to the polynomial SVM are significant at the 0.01 level. These findings suggest that the problem has non-linear characteristics and is best solved by a polynomial classifier. The distribution of scores (not shown due to space limitations) for the test articles ranked with the polynomial SVM also shows a separation of the articles.

2. 11 point precision-recall

The traditional 11-point precision-recall curve [28] gives the user an idea of what fraction of the articles she would need to peruse to get a certain recall level. For example in Figure 1, at the 0.2 recall level, the precision is 0.68 for the linear SVM (i.e., 68% of the total articles returned are of high quality). Contrast with Naïve Bayes where only 37% of the total articles returned are of high quality. Figure 1 is an average over 10 folds. It is specific to this test set and is a method to compare retrieval/learning methods to one another.

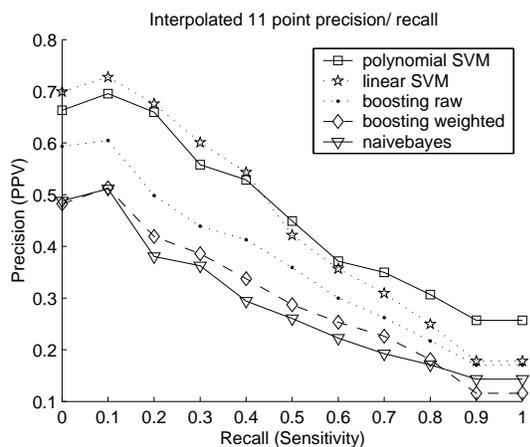


Figure 1: Interpolated 11-point curves

3. Comparison to clinical query filters

We compare now the sensitivity, specificity, precision, and accuracy of the machine learning methods against the clinical query filters optimized for sensitivity and specificity. Recall that the clinical query filters return a Boolean result of whether an article is in the positive class or not. The learning methods return a score. To make the comparison, we use the CQF optimized for sensitivity and vary the threshold for the article scores produced by the learning methods until their sensitivity matches the optimal CFQ sensitivity *for each fold*. The same procedure was run for the optimal CQF specificity. 95% confidence intervals were calculated based on a binomial distribution [27]. Tables 2-3 illustrate these results.

Within each fold, we statistically compare the learning methods to the CQF using McNemar's test [27]. 10 p-values are generated for each method. The p-values vary with some folds being highly significant and others not as significant. To reconcile these irregularities, the 10 p-values are averaged and a vote of significance by each fold is taken. At fixed sensitivity, the mean p-values of the 10 folds for all the methods are significant at the 0.05 level except

for Boostexter with raw input at the 0.14 level and the linear SVM at 0.06 level. Using votes, a 2/3 majority of folds in each method are significant at the 0.05 level.

	Sensitivity	Specificity
ClinSensOpt		0.736(0.713, 0.758)
SensLinSVM	0.967(0.823, 0.999)	0.791 (0.77, 0.811)
SensPolySVM		0.875(0.857, 0.891)
SensNaiveBay		0.768(0.747, 0.789)
SensBtextwght		0.635(0.611, 0.659)
SensBtextraw		0.786(0.765, 0.806)
		Precision
ClinSensOpt	0.07 (0.046, 0.101)	0.754(0.732, 0.775)
SensLinSVM	0.107(0.076, 0.146)	0.81(0.79, 0.829)
SensPolySVM	0.169(0.119, 0.229)	0.893(0.877, 0.908)
SensNaiveBay	0.091(0.064, 0.126)	0.787(0.766, 0.807)
SensBtextwght	0.065(0.046, 0.089)	0.655(0.63, 0.678)
SensBtextraw	0.099(0.069, 0.136)	0.804(0.784, 0.824)

Table 2: Learning methods at fixed sensitivity

For most learning method, at score thresholds equal to optimal CQF sensitivity, the values for both specificity and precision are superior to the CQF values.

	Sensitivity	Specificity
ClinSpecOpt	0.367(0.199, 0.561)	0.958(0.947, 0.968)
SensLinSVM	0.767(0.577, 0.901)	
SensPolySVM	0.80(0.614, 0.923)	
SensNaiveBay	0.60(0.406, 0.773)	
SensBtextwght	0.60(0.406, 0.773)	
SensBtextraw	0.667(0.472, 0.827)	
	Precision	Accuracy
ClinSpecOpt	0.15 (0.115, 0.191)	0.948(0.935, 0.958)
SensLinSVM	0.264(0.176, 0.37)	0.955(0.943, 0.964)
SensPolySVM	0.281(0.191, 0.386)	0.955(0.944, 0.965)
SensNaiveBay	0.22(0.136, 0.325)	0.951(0.94, 0.962)
SensBtextwght	0.22(0.136, 0.325)	0.951(0.94, 0.962)
SensBtextraw	0.238(0.152, 0.344)	0.953(0.941, 0.963)

Table 3: Learning methods at fixed specificity

At fixed specificity, the mean p-values of the 10 folds are all significant at the 0.05 level. Using votes, a 2/3 majority of folds in each method are also significant at the 0.05 level.

Fixing specificity yields superior sensitivity and precision. Sensitivity rises from the base of 37% to 80%, and precision rises from 15% to 28%.

Discussion

To summarize the results, the learning methods in these experiments exhibit high discriminatory performance as measured by the AUC; the resulting models outperform the Boolean-based CQF; Polynomial SVMs have the best performance as measured by the AUC and the 11 point curve.

The choice of the gold standard is important for the external validity of these results. It is possible that ACP journal reviewers may have accepted or rejected some paper inadvertently relative to the ACP journal stated criteria. However, the international authority of

the ACPJC, as a premier international meta-publication with dedicated experts selecting articles by thorough review of the literature based on explicit and well-defined criteria, provides a strong basis for a gold standard; we propose that, as a result, the external validity of this group's selections meet and exceed the validity of most, if not all, ad-hoc groups of experts brought together for the sole purpose of rating articles manually to construct CQFs. In addition, the ACPJ's more recent selections are readily available in electronic form for automatic corpus construction.

We also note that we attempted to obtain the original gold standard used by Haynes et al for additional comparisons with those models [3]. Unfortunately this data is no longer available (Haynes, personal communication).

We are currently exploring extensions to all article class areas. The methodology is identical, and preliminary results suggest comparable performance.

The mapping of such models to EBM retrieval is straightforward. Most authors assume so (implicitly) in their research and focus on the harder problem of finding the high quality articles. For example, one setup is to run a Boolean query of the clinical question and then run the high quality treatment filters on the Boolean results (or vice-versa). We recognize that the work presented here is a first step for addressing the high-quality, content-specific classification problem. This work paves the way for an applied system to realize the clinical utility of these techniques.

References

1. Sackett, D.L., et al., *Evidence Based Medicine: How To Practice and Teach EBM*. 1998, Edinburgh: Churchill Livingstone.
2. Bigby, M., *Evidence-based medicine in a nutshell. A guide to finding and using the best evidence in caring for patients*. Arch Dermatol., 1998. **123**(12): p. 1609-18.
3. Haynes, B., et al., *Developing Optimal Search Strategies for Detecting Sound Clinical Studies in MEDLINE*. JAMIA, 1994. **1**(6): p. 447-458.
4. <http://www.ncbi.nlm.nih.gov/PubMed/>
5. *Purpose and Procedure*. ACP Journal, 1999. **131**(1): p. A-15 - A-16.
6. Jerome, R.N., et al., *Information Needs of clinical teams: analysis of questions received by the Clinical Informatics Consult Service*. Bull Med Libr Assoc, 2001. **89**(2): p. 177-184.
7. <http://www.mysql.com/>
8. <http://www.abbyy.com/>
9. Kulikowski, C.A. and S. Weiss, eds. *Computer Systems That Learn*. ed. M. Kauffman. Jan 1991.

10. Metz, C.E., *Basic Principles of ROC Analysis*. Sem in Nuc Med, 1978. **8**(4): p. 283-298.
11. www.princeton.edu/~biolib/instruct/MedSW.html
12. <http://www.tartarus.org/~martin/PorterStemmer/>
13. Porter, M.F., *An algorithm for suffix stripping*. Program, 1980. **14**(3): p. 130-137.
14. Leopold, E. and J. Kindermann, *Text Categorization with Support Vector Machines. How to Represent Texts In Input Space?* Machine Learning, 2002. **46**: p. 423-444.
15. Salton, G. and C. Buckley, *Term Weighting Approaches in Automatic Retrieval*. Info Process and Management, 1988. **24**(5): p. 513-523.
16. Schapire, R.E. and Y. Singer, *Boostexter: A Boosting-based System for Text Categorization*. Machine Learning, 2000. **39**(2/3): p. 135-168.
17. Duda, R., P. Hart, and D. Stork, *Pattern Classification*. 2nd ed. ed. J.W. Sons. 2001.
18. Joachims, T. *A probabilistic analysis of the Rocchio Algorithm With TFIDF for text categorization*. in *14th International Conference on Machine Learning*. 1997. Nashville, TN: Morgan Kaufman.
19. Mitchell, T.M., *Machine learning*. 1997, New York: McGraw-Hill. xvii, 414.
20. Wilbur, W.J. *Boosting Naive Bayesian Learning on a Large Subset of MEDLINE*. in *AMIA*. 2000. Los Angeles, CA: Hanley & Belfus, Inc.
21. Schapire, R.E. and Y. Singer, *Improved boosting algorithms using confidence rated predictions*. Machine Learning, 1999. **37**(3): p. 297-336.
22. www.cs.princeton.edu/~schapire/boostexter.html
23. Cristianini, N. and J. Shawe-Taylor, *An Introduction to Support Vector Machines and other Kernel-based Learning Methods*. 2000: Cambridge University Press.
24. Burges, C., *A tutorial on support vector machines for pattern recognition*. Data Mining and Knowledge Discovery, 1998. **2**: p. 121-167.
25. Vapnik, V., *Statistical Learning Theory*. 1998: Wiley.
26. Joachims, T., ed. *Making Large-Scale SVM Learning Practical. Advances in Kernel Methods - Support Vector Learning.*, ed. B. Scholkopf, C. Burges, and A. Smola. 1999, MIT-Press.
27. Pagano, M. and e. al, *Principles of Biostatistics*. 2000: Duxbury Thompson Learning.
28. Baeza-Yates, R. and B. Ribeiro-Neto, *Modern Information Retrieval*. 1999, Harlow, England: Addison-Wesley.

Acknowledgements

The first author is funded by Vanderbilt University Funds. The authors wish to thank Drs. John Wilbur, Bill Hersh, and Brian Haynes for valuable comments and suggestions.