Learning Boolean Queries for Article Quality Filtering

Yin Aphinyanaphongs[a], Constantin Aliferis[a]

[a]Department of Biomedical Informatics, [a]Vanderbilt University, Nashville, TN

Abstract

Prior research has shown that Support Vector Machine models have the ability to identify high quality content-specific articles in the domain of internal medicine. These models, though powerful, cannot be used in Boolean search engines nor can the content of the models be verified via human inspection. In this paper, we use decision trees combined with several feature selection methods to generate Boolean query filters for the same domain and task. The resulting trees are generated automatically and exhibit high performance. The trees are understandable, manageable, and able to be validated by humans. The subsequent Boolean queries are sensible and can be readily used as filters by Boolean search engines.

Keywords:

Information Storage and Retrieval, PubMed, Medical Informatics, Artificial Intelligence, Text Categorization

Introduction

The pace of research far overcomes the ability of modern health professionals to be up to date about all the recent research developments and current best practices. By one account, a general physician reviewing just 20 clinical journals in adult internal medicine would have to read 19 articles a day for 365 days a year to keep up [1]. Increasingly, physicians are turning to electronic sources for their information needs. Services like MDConsult [2], Up2Date [3], and Pubmed Central [4] evaluate and abstract research articles.

However, the final authority on what constitutes best medical practices and high quality knowledge is provided by the primary sources themselves (i.e the biomedical research literature). Thus there exists a great need for a way to identify the most important of the primary sources, that is the original research, the methodological quality and scope of which are likely to yield the highest benefit to the healthcare professionals.

A primary point of practical significance is the technology and overall process for constructing quality filters to return these primary sources. More specifically, in most cases, filters consist of Boolean queries that were formulated by taking human-derived queries and modifying them, or by stringing together words that are deemed intuitive by human experts for some domain in disjunctions or conjunctions and evaluating their performance [5, 6]. A more structured, yet still, ad-hoc approach was taken to generate Boolean queries to return high quality content related articles in [7] and [8]. In the pioneering study in [8], experts were polled, and words that were deemed relevant to a content area were selected. The exact combination of words was optimized separately for sensitivity and specificity by a brute force search of all disjunctions of the selected words (up to a small number of words per query). The resulting queries perform well, and are featured in the clinical queries (CQF) link in PubMed [4]. Alternately in [7], word frequencies in the abstract were used to identify candidate terms. These terms are individually evaluated for sensitivity and precision, and the terms with the highest (sensitivity * precision) product were combined in a disjunctive Boolean query to find diagnostic studies. The authors report improved performance over the CQF diagnostic filter.

The authors of the present article in [9] address the problem of returning quality articles by running a suite of powerful classifiers on a suitable corpus and not rely on human experts. While the resulting models perform very well, a question remains as to (a) their understandability by humans, and (b) their usability through Boolean based systems such as Pubmed. Even though the Boolean model can capture any set of documents, the process of formulating such queries, especially by humans, can be challenging. Indeed, analysis of search engine logs show that most search engine users avoid Boolean formulations [10].

Flake et. al. recently introduced a hybrid approach that converts corpus-based SVM models to Boolean queries in the web domain [11]. Their method combines in an ad-hoc manner a linear approximation to a polynomial SVM classifier with a modified Adaboost [12] algorithm to convert the original polynomial SVM models to sets of Boolean Queries (also referred to as "query modifications" in the information retrieval literature). The Flake et al. method is highly heuristic and not guaranteed to perform well in specific data and problem domains, however.

Thus the motivation for this paper is how to convert sophisticated machine learning models into usable queries. The application of SVMs to current information retrieval systems is not straightforward and would require a dedicated system built expressly for this purpose. To bridge the gap and give users

applicable technology, we explore the formulation of Boolean queries from a training corpus that includes examples of the high quality content specific articles. Specifically, we ask the question:

Is it possible to automatically construct Boolean queries from a corpus using machine learning techniques such that the Boolean queries have as good classification performance as the SVM models, and are the resulting Boolean queries human-readable, manageable, and simple for use in current search engines?

Throughout the present paper, we use "word", "term", "feature", and "variable" interchangeably. The choice of word depends on the appropriate context in which it is found.

Methods

Corpus Preparation

We use for the present study a modified version of the corpus in [9]. This corpus uses the ACP journal as a gold standard for both content and quality of articles [13]. The ACP journal is a meta-publication that routinely reviews over a hundred journals for articles that meet its selection criteria. Articles that are abstracted or cited by the ACP are considered positive instances and all other articles in the same journals to be negative. A more detailed description of the gold standard construction methodology can be found in [9]. The criteria for inclusion in ACPJ can be found in [13].

We selected the treatment content area for several reasons. This area had sufficient sample to represent the concepts for a high-quality treatment article. The criteria for selection are simple, and the predominant class of questions asked by physicians is treatment related [14].

The conversion of documents to a format suitable for the machine learning algorithms followed the procedures in [9] closely. The articles in the ACP selected journals were cross-referenced in PubMed, and the title, abstract, and MeSH terms parsed. The processing of the terms differs from [9] in that title and abstract terms were represented separately rather than as one group.

The resulting terms were encoded as binary variables (either appearing in the document or not) in all documents. The final treatment category counts included 397 positive documents and 15407 negative documents with 27891 unique words.

The articles were further split into a training, validation, and test set, with 221 positive / 8998 negative, 76 positive/ 3081 negative, and 82 positive/ 3328 negative documents respectively. A single split was selected because the sample size was large enough, and utilizing a single split simplified the creation of a single Boolean query by removing concerns about how to combine the queries from each split.

Support Vector Machine Classifiers

We used a support vector machine (SVM) from our previous experiments as an empirical "upper bound" on the performance of the binary encoded test set. SVMs function as both linear and non-linear classifiers. They maximize the margin between the instances belonging to different classes. The solution that generalizes best to unseen instances is found by solving a constrained quadratic optimization problem in terms of the patterns that lie on the margin (i.e. support vectors) [15].

We use a Matlab [16] wrapper [17] for Thorsten Joachim's SVM-light [18]. This implementation utilizes a decomposition method to make learning a large number of examples tractable [19]. We use misclassification costs of {0.1, 0.2, 0.4, 0.7, 1.0, 2.0} and degrees of {1, 2, 5} on the validation sets. The best performance combination of degree and cost was used on the test set.

Decision Tree Classifiers

Our primary means to generate Boolean queries is induction of decision trees. The reason for this choice is that the output of a decision tree maps well to Boolean queries. Each leaf of the decision tree corresponds to a path that describes the conjunction of word absence or presence for a classification.

In the text categorization domain, decision trees are a learning method that attempts to partition a training set based on individual words that describe the domain. The extensive work of Apte and Weiss [20], demonstrated that decision trees can produce superior classification performance in text while producing trees that are understandable. Our work extends the findings of Apte in several ways. First, we construct and apply the work to a new task. Second, we introduce new feature selection methods. Third, we analyze the trees in this problem domain to address the understandability and manageability of the resulting queries.

In this paper, we use the CART implementation of decision trees in Release 13 of Matlab with the gini index of diversity [21] to rank the relevant features. The full tree is pruned based on retaining a performance of at least 1% of the maximum performance on the validation set with the smallest tree size. For example, suppose the best tree performs at 92% AUC with 10 nodes and a smaller tree performs at 91% AUC with 5 nodes. We would select the smaller tree as it retains at least 1% AUC of the maximum.

The Flake algorithm was implemented by the first author in Matlab following the description in [11] since public domain code is not currently available.

Feature Selection Algorithms

Decision trees are known to suffer from the curse of dimensionality [22]. As the number of features increases, the increase in sample size must grow exponentially in the worst case, or the decision tree will not generalize well. To overcome this problem, we use several feature selection algorithms with the decision tree.

In our first evaluation of the method we employ three variable selection algorithms:

Linear and Approximate-Polynomial Recursive Feature Elimination (RFE_L, RFE_{PA})

RFE builds on the power of SVM classification. The basic procedure can be summarized as follows [23]:

1. Build an SVM classifier using all V features

2. Compute weights of all features and choose the first $|V|*k$ features (sorted by weight in decreasing order, k being a feature set cardinality reduction parameter, typically set to 0.5)
3. Repeat steps 1 and 2 until an empty feature set is produced
4. Choose among all feature subsets created the one that gives the best performance in a validation set

Linear RFE (RFE_L) uses linear SVMs in step 1 as the name implies. In step two features are selected by their weights. In Approximate-Polynomial RFE (RFE_{PA}) a polynomial-kernel SVM is used in step 1 while Step 2 uses, instead of weights, ranking coefficients such that the ranking coefficient of the feature ***i*** is the change of cost function by removing feature ***i***. As a speed-up heuristic, one does not recompute Lagrange coefficients while ranking features. We also note that in the linear case, non-linear RFE is identical to the linear RFE. The exact mathematical formulations and parameter values used for both methods can be found in [23].

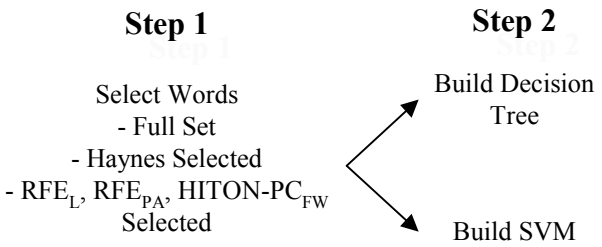
HITON-PC_{FW} (filtered and wrapped HITON PC)

HITON is a feature selection algorithm introduced in [24] that combines induction of Markov Blankets and wrapping (i.e., heuristic search over variables subsets) to identify the smallest variable subset that gives optimal classification performance. It was shown by its authors (a) to be sound given the distributional assumption of faithfulness, universal approximator learners, and a quadratic loss misclassification function (for details please see the original publication); and (b) to have superior variable reduction performance (while maintaining optimal or near-optimal classification performance) to a range of state-of-the-art variable selection methods across a representative sample of biomedical tasks, including text categorization. Given HITON's powerful reduction capabilities we apply it in our experiments.

In order to significantly speed-up the algorithm we modify HITON in two ways: (a) we apply, as a first step, a univariate association-based reduction in the number of terms used (which was shown in [25] to lead to excellent classifiers - but not optimally small ones) and (b) we do not pursue full induction of the Markov Blanket (i.e., parents, children and spouses of the Target category in the Bayesian Network representing the classification tasks) but use an approximation to the Markov Blanket the parents and children only.

The price paid for the resulting speed up is that the modified algorithm is no longer sound even if the original HITON assumptions hold. This is because some members of the Markov Blanket (i.e., parents of children that do not have direct arcs with the target variable) will be omitted; yet they are necessary for optimal classification in the worst case. As we will see this heuristic modification to HITON works well in our experiments.

Figure 1 – Experimental Design Methodology



Experimental Design

The design is a simple 2 step methodology. In step 1, a word set is selected to represent the domain. In step 2, an SVM classifier and a decision tree classifier are trained using this word set. This design is illustrated in Figure 1.

For step 1, we used 3 sets of words as inputs to the decision tree: the word set with the best performance/ feature ratio from each of the 3 selection methods, the full word set, and the word set from the Haynes experts [8].

The decision trees and the subsequent Boolean queries are evaluated quantitatively via a combined sensitivity-specificity measure to the CQF filters of Pubmed; they are also examined qualitatively.

Results

The performance to feature results are shown in Table 1. We use SVMs and examine the area under the receiver operating curve (AUC). The Markov blanket HITON-PC_{FW} algorithm has the best performance-to-feature ratio and is able to reduce from 27891 features to 13.

Table 1 – Feature Selection Performance

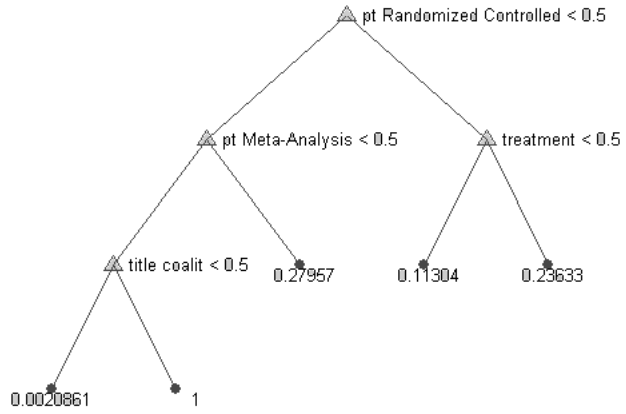
HITON-PC_{FW} (13 Features)*				0.92 AUC		
RFE						
Features	**28000**	**1743**	**871**	**217**	**54**	**13**
RFE_L	0.95	0.85	0.96	0.97	0.86	#
RFE_{PA}	0.83	0.95	0.94	0.95	0.92	0.91

* - HITON-PC_{FW} returns a single set.

- RFE_L did not converge to a solution

The performance of decision trees with varying inputs are shown in Table 2. The best decision tree (best AUC performance) is illustrated in Figure 2.

Figure 2 – Decision tree produced by HITON/ DT



The triangles are decision nodes. The left branch corresponds to the word being absent, and the right branch to the word being present. The leaves indicate the probability of a high quality treatment related document.

Table 2 – Decision Tree Performance on Test Set

Method	AUC	Words in pruned tree
Full Feature Set (27891 features)		
- SVMs	0.98	N/A
- DT	0.94	2
HITON-PC_{FW} Feature Set (13 features)		
- SVM	0.95	N/A
- DT	0.95	4
Haynes Feature Set (747 features)		
- SVM	0.94	N/A
- DT	0.93	2

Table 2 shows that the best performing decision tree is using the HITON-PC_{FW} feature set. The other decision tree methods follow closely. The words in the trees differ. Using the full feature set, the terms "publication type (pt) randomized controlled trial (RCT)"(top node) and "pt meta-analysis" are returned. Using the Haynes feature set, the terms "pt RCT" (top node) and "mesh heading RCT" are returned. The terms using the HITON-PC_{FW} feature set are in Figure 2.

Table 3 compares the CQF filters with the decision trees. For each constructed decision tree we measure the sensitivity and specificity for the given task. These statistics provide two related measures for comparing two algorithms. None is sufficient by itself. This is because an algorithm may achieve perfect sensitivity by classifying all samples as positive or perfect specificity by classifying all samples as negative. Thus, a combined measure is required. The measure we used is the proximity of the sensitivity and specificity of the algorithm to perfect sensitivity and specificity expressed as [26]:

$$dist = \sqrt{(1-sens)^2 + (1-spec)^2}$$

Note, that we cannot use AUCs or fix the measures. First, AUC's cannot be generated for the CQF filters because the documents are not ranked. Either the query is satisfied or not. Second, fixing sensitivity and specificity as used in [9] cannot be used because of the limited thresholds output by the decision trees. Equivalent matches cannot be generated.

The decision tree methods (bolded) outperform both optimized CQF filters and have the best tradeoff between sensitivity and specificity.

Table 3 –Decision Trees Compared to CQF filters

Method	Distance
CQF filter – optimized for sensitivity	0.23
CQF filter – optimized for specificity	0.50
Full feature set/ decision tree	**0.11**
HITON features set/ decision tree	**0.11**
Haynes feature set/ decision tree	**0.11**

In additional experiments, we ran the Flake method on this dataset. We found that the classifier performance was poor and selected counter-intuitive terms. Since the Flake method is highly heuristic and not designed for this domain, we did not pursue it further.

Discussion

Every decision tree method produces a tree that is manageable, readable, and can be validated by humans. The simplicity of the solutions is not surprising since, for this proof-of-concept study, we purposely chose a task that had simple guidelines.

Specifically, the decision tree in Figure 2 has words that are intuitive to the treatment class. Publication type (pt_) randomized controlled trial and pt meta analysis seem appropriate considering the criteria of the ACP journal [13]. The ACPJ criteria for treatment 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.

The Boolean queries at each leaf appear to be equally sensible. For example, the leaf obtained with pt randomized controlled trial with the word treatment in the abstract has a 24% probability of being a good document. Human experts could develop this Boolean query intuitively.

The next highest leaf is the article is *not* a pt randomized controlled trial, but is a pt meta analysis, then we are 28% sure that the article is of high quality in the treatment class. This query is less intuitive since it says that meta-analysis qualify as high quality treatment related articles.

In light of these Boolean queries, how easy would it be for an expert to construct them? The first query seems straightforward and is an example of a disjunctive query that experts excel at constructing. It follows closely the intuitive notion of what content bearing words would indicate a high quality treatment related study. We argue that second query is more difficult for an expert to construct. Experts can readily explain, in specific instances, what would make a good document, but when it comes to generating an efficient query, especially with "not" qualifiers, the problem of selecting the appropriate words becomes very hard [27, 28].

Given the good performance of the decision trees, it makes sense to ask why the feature selection process is necessary if running a decision tree using the full feature set produces good results? The answer is not apparent in this data set. More complex tasks may *require* feature selection. Towards this we ran the same experimental design in the etiology category area, and preliminary results show that a decision tree approach on the full feature set does not produce as good results as the feature selection/ decision tree method. HITON-PC_{FW} achieves a better performance in etiology while reducing the

Table 4 – Treatment/ Etiology Decision Tree AUCs

Category	Full/DT	Features	HITON DT	Features
Treatment	0.94	27891	0.95	13
Etiology	0.80	27891	0.90	9

number of features from ~28000 to 9 features.

The methodology in the present paper reduces labor by learning the needed words from a corpus rather than asking experts to define words that represent the treatment area. This approach has two advantages. It reduces variability in term selection and bypasses the need for appropriate experts. While the opposite method of [8] produces results comparable to the corpus-based methods (Table 2), the resulting tree is limited in its interpretation. For example, the best Boolean query is "not pt randomized controlled trial and mesh heading randomized controlled trial." This query essentially represents the same concept, and, in comparison to the second query, misses the pt meta analysis concept. The Boolean query construction of Haynes is sub-optimal for this category as seen by the distance measures in Table 3, and lower AUC in Table 2. The possibility of missing words that describe the content is an weakness in the methodology. Similarly human cognitive biases equating co-occurrence with association hinder the construction of effective Boolean queries by experts [28].

Conclusions

The contribution of this paper is 4-fold. First we have presented a combined feature selection/decision tree method that can produce decision trees that perform as well as the best text classifiers and outperform methods currently available for this task. Second, these decision trees are understandable, manageable, and amenable to validation by humans. Third, these trees and queries are generated automatically from a corpus hence the process can be readily repeated many times in similar domains/tasks. Fourth, the Boolean queries discovered can be readily applied in existing search engines.

Our future research will also explore this method in more difficult categories with broader criteria for ACP inclusion such as diagnosis, prognosis, and etiology to further delineate the limits of the methodology presented here as well as potential improvements.

Acknowledgments

Yin Aphinyanaphongs is supported from NLM training grant: T15 LM07450-01. The authors would like to thank Ioannis Tsamardinos and Alexander Statnikov for help with the feature selection methods and useful discussions.

References

1. Davidoff, F., et al., *Evidence Based Medicine: A New Journal To Help Doctors Identify the Information They Need.* BMJ, 1995. **310**: p. 1085-6.
2. www.mdconsult.com
3. www.up2date.com
4. http://www.ncbi.nlm.nih.gov/PubMed/
5. Shojania, K.G. and L.A. Bero, *Taking Advantage of the Explosion of Systematic Reviews: An Efficient MEDLINE Search Strategy.* Effec Clin Prac, 2001. **4**(4): p. 157-159.
6. Robinson, K.A. and K. Dickersin, *Development of highly sensitive search strategy for the retrieval of reports of controlled trials using PubMed.* Int Epi Assoc, 2002. **31**: p. 150-153.
7. Bachmann, L., et al., *Identifying Diagnostic Studies in MEDLINE.* JAMIA, 2002. **9**(6): p. 653-658.
8. Haynes, B., et al., *Developing Optimal Search Strategies for Detecting Sound Clinical Studies in MEDLINE.* JAMIA, 1994. **1**(6): p. 447-458.
9. Aphinyanaphongs, Y. and C.F. Aliferis. *Text Categorization Models for Retreival of High Quality Articles in Internal Medicine*. in *AMIA*. 2003. Wash, D.C.
10. Silverstein, C., et al. *Analysis of a Very Large Web Search Engine Query Log*. in *SIGIR FORUM*. 1999.
11. Flake, G., et al. *Extracting Query Modifications from Nonlinear SVMs*. in *Int WWW Conf*. 2002. Honolulu, HA.
12. Schapire, R.E. *Theoretical views of boosting and applications*. in *Tenth International Conference on Algorithmic Learning Theory*. 1999.
13. *Purpose and Proc.* ACP Journal, 1999. **131**(1): p. A15.
14. Jerome, R.N., et al., *Info Needs of clinical teams: analysis of questions received by the Clinical Informatics Consult Service.* Bull Med Libr Assoc, 2001. **89**(2): p. 177-184.
15. Burges, C., *A tutorial on support vector machines for pattern recognition.* Data Mining and Knowledge Discovery, 1998. **2**: p. 121-167.
16. www.mathworks.com
17. http://www.cis.tugraz.at/igi/aschwaig/software.html
18. 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.
19. Osuna, E., R. Freund, and F. Girosi. *Training support vector machines: an application to face detection*. in *Conf on Computer Vision and Pattern Recognition*. 1997.
20. Apte, and Weiss, *Data Mining with Decision Trees and Decision Rules.* Future Gener Computer Systems, 1997.
21. Murthy, S., *Automatic Construction of decision trees from data: A multi-disciplinary survey.* Data Mining and Knowledge Discovery, 1997.
22. Duda, R., P. Hart, and D. Stork, *Pattern Classification*. 2nd ed. ed, ed. J.W. Sons. 2001.
23. Guyon, I., et al., *Gene Selection for Cancer Classification using Support Vector Machines.* Machine Learning, 2002. **46**: p. 389-422.
24. Aliferis, C.F., I. Tsamardinos, and A. Statnikov. *HITON: A Novel Markov Blanket Algorithm for Optimal Variable Selection*. in *AMIA*. 2003. Washington, DC.
25. Yang, Y. and J. Pederson. *A comparative study on feature selection in text categorization*. in *14th International Conference on Machine Learning*. 1997: M. Kauffman.
26. Tsamardinos, I., C.F. Aliferis, and A. Statnikov. *Time and Sample Efficient Discovery of Markov Blankets and Direct Causal Relations*. in *Proceedings of the 9th ACM SIGKDD International Conference on KDD*. 2003.
27. Bourne, L.E., Jr. and D.E. Guy, *Learning Conceptual Rules: II The role of positive and negative instances.* Journal of Experimental Psychology, 1968. **77**: p. 488-494.
28. Plous, S., *The Psychology of Judgement and Decision Making*. 1993, McGraw-Hill Inc: New York.

Address for correspondence

Yin Aphinyanaphongs, ping.pong@vanderbilt.edu.