Categorization of Sentence Types in Medical Abstracts

Larry McKnight, MD[1], and Padmini Srinivasan, Ph.D[2]

[1]Department of Medical Informatics, Columbia.University, New York, NY
[2]National Library of Medicine, Bethesda, MD

ABSTRACT

This study evaluated the use of machine learning techniques in the classification of sentence type. 7253 structured abstracts and 204 unstructured abstracts of Randomized Controlled Trials from MedLINE were parsed into sentences and *each sentence* was *labeled as one of four types (Introduction, Method, Result, or Conclusion). Support Vector Machine (SVM) and Linear Classifier models were generated and evaluated on cross-validated data. Treating sentences as a simple "bag of words", the SVM model had an average ROC area of 0.92. Adding a feature of relative sentence location improved performance markedly for some models and overall increasing the average ROC to 0.95. Linear classifier performance was significantly worse than the SVM in all datasets. Using the SVM model trained on structured abstracts to predict unstructured abstracts yielded performance similar to that of models trained with unstructured abstracts in 3 of the 4 types. We conclude that classification of sentence type seems feasible within the domain of RCT's. Identification of sentence types may be helpful for providing context to end users or other text summarization techniques.*

INTRODUCTION

With increasing availability to large stores of online text data, automated text categorization continues to be an important area of research. As this field matures, several important questions about classifier performance have been answered. However, while several articles have been published on text categorization based on content[1,2,3], few have reported on the categorization of contextual information (e.g., whether the text is found in the title, abstract or a particular section of the article).

The fact that information about the text context is related to the content is of no surprise. Many have noted, for example, the common practice of busy clinicians reading the last few sentences of the abstract first to determine if the rest of the article is worth reading. The determination of the context from content has great benefit since in many situations the contextual information may be difficult to obtain, ambiguous, or just plain missing.

Yet, some programs depend on these contextual clues for adequate performance. For example, beyond simple classification, work is proceeding at the National Library of Medicine (NLM) to abstract conceptual relations from text with a tool SemRep[4]. SemRep uses natural language processing techniques and does well when given conclusion type sentences, but has more difficulty with any general sentence. Therefore information about the type of sentence would be helpful as a preprocessing step.

This study addresses the possibility of using machine-learning techniques to identify contextual information directly from free text. Specifically, we examine the feasibility of classifying sentences into general high-level categories that would be found in a structured report. This classification task differs from other reports of text categorization in that the class labels are fewer, but the determinates of a given class label may be broader and more abstract. Additionally, previous reports of text classification have generally involved larger units of analysis such as abstracts or documents. The use of sentences as the unit of analysis leads to increased sparseness in an already sparse data

vector. This makes feature limiting more problematic.

METHODS

To test the efficacy of machine learning techniques on sentence classification, we designed an experiment to train SVM's and linear classifiers on a random sampling of medical abstracts of randomized control trials (RCTs). We limited our studies to RCTs to build on previous work[5]. In the past RCTs had been chosen because of their high volume, relative consistency.

Approximately 12 million MedLINE abstracts from 1976 to 2001 were evaluated for criterion matching the Haynes[6] filter for Randomized Control Trials on medical therapy along with the subheads "drug therapy" and "therapeutic use." This yielded 37,151 abstracts. From this sample, 7253 abstracts were marked as "structured" based on the presence of structure labels. A preliminary study identified 362 structure labels though iterative passes with regular expression matches looking for short phrases (1-4 words) followed by a colon or dashes. Reviewing this technique on independent structured and unstructured abstracts indicated an error rate of less than 1% for this technique.

Both structured and unstructured abstracts were then parsed into 334,623 sentences (90,655 structured, 243,958 unstructured) using Adwait Ratnaparkhi's mxTerminator[7]. MxTerminator is an entropy based sentence tagger that is trainable. The results reported here used mxTerminators default model, trained on a large corpus of text from the New York Times. We found accuracy of sentence breaks using this model to be 93.5% on the MedLine abstracts in our experiment. Retraining mxTerminator using the corrected sentences generated in this evaluation (1532 sentences) and testing on 400 fresh sentences, accuracy increased to 97%. However, this retrained sentence model was generated after the experiments were performed, and therefore could not be used for results presented here.

From the 30,182 unstructured abstracts, 204 (1629 sentences) were randomly chosen and manually reviewed and classified as one of the following four types: 'introduction', 'method', 'result', or 'conclusion'. Criteria were set for identifying these types as follows. An 'introduction' sentence was defined as a sentence that describes the need for the study or prior work. A 'method' was defined as something the investigators did. A 'result' was defined as something the investigators found in the study. A 'conclusion' was defined as a statement of fact that applies to cases outside of the study population, or a sentence that was specifically states itself to be a conclusion. In ambiguous cases the later type was assigned. For example, if a sentence had both a result and a conclusion the sentence was marked as a conclusion. In the process of labeling, we also identified and fixed many sentence break errors. This reduced the total number of labeled sentences to 1,532.

Next, we labeled the sentences in the structured abstracts based on the most immediate structure label. The vast majority of the 362 labels were categorized into one of the 4 types and all sentences from the label to next label (or end of the abstract) were marked with that category. Some of the labels were ambiguous however, leaving approximately 4% of the sentences not categorized.

The labeled sentences were converted to a vector format for use with the linear classifier and support vector machine by creating a lexicon of unique words in the corpus and labeling each word with a unique feature number. Experiments were run both treating the sentence as a simple "bag of words", and also by adding an additional feature that indicated the sentence location in the abstract (ie. 0=first sentence, 0.5=midway through, 1=last sentence) Several word weighting schemes were tried however all performed similarly, so later experiments were taken using the simple binary presence of the word, without respect to its document or corpus frequency.

Support Vector Machine models were trained and evaluated using Joachams' `svm_light`

software[3]. In preliminary studies linear, polynomial and radial bias kernels were tried. In nearly all cases the linear kernel outperformed the others and were therefore used for this evaluation.

Linear models were trained and evaluated using code written by Miguel Ruiz[8]. In preliminary experiments Widrow-Huff (WH), Rocchio, and EG learning algorithms were evaluated, however models generated by Rocchio and EG performed more poorly than those from the WH so subsequent evaluation was only performed using this WH. Preliminary experiments with feature limitation showed generally decreased performance below 300 features, but minimal performance improvement for greater than 300 features. For experiments reported here, features were limited to 300.

With exception of the final experiment, all machine learning performance tests were conducted using 10 fold cross-validation, using random splits in the data. For each holdout test set, sensitivity and specificity were calculated based on thirteen discrete threshold values that were subsequently used to calculate estimates of ROC area using the method described by Pollack and Norman[9]. The F-measure[10] is reported as a composite measure of precision and recall using a ß of 1. This is listed as F_1 in the results. Measures of simple accuracy, precision, recall and F_1 were taken from the threshold that yielded the highest simple accuracy. Results reported for the cross-validated data represent the average of the 10 testing sets.

A series of experiments were performed. The first consisted on a simple learning of the unstructured (hand labeled) sentences. Next, models were constructed and tested for the larger sample of structured abstracts (labels assigned from regular expression match). Next, a smaller random subset of the structured sentences was selected to match the unstructured dataset in size and character. This dataset was manually reviewed and sentence break errors

Abstract type	Training Cases*		Model Type	Sentence Location Feature	Linear Classifier (Widrow-Huff)				SVM			
	Total	Positive			ROC	Acc	P/R	F_1	ROC	Acc	P/R	F_1
Unstructreud	1,532	196 (12.8%)	Intro	N	0.863	0.889	0.71/0.34	0.465	0.910	0.921	0.82/0.49	0.616
				Y	0.867	0.890	0.71/0.30	0.423	0.957	0.947	0.88/0.72	0.789
		430 (28.1%)	Method	N	0.851	0.829	0.85/0.48	0.611	0.935	0.894	0.87/0.74	0.800
				Y	0.854	0.832	0.86/0.48	0.612	0.954	0.909	0.84/0.84	0.837
		686 (44.8%)	Result	N	0.672	0.737	0.82/0.57	0.672	0.920	0.863	0.85/0.85	0.851
				Y	0.650	0.730	0.80/0.55	0.650	0.930	0.860	0.83/0.86	0.845
		220 (14.4%)	Concl	N	0.877	0.888	0.77/0.33	0.457	0.911	0.903	0.67/0.60	0.639
				Y	0.883	0.891	0.77/0.33	0.457	0.965	0.936	0.81/0.74	0.773
Structured	1,669	314 (18.8%)	Intro	N	0.832	0.854	0.69/0.41	0.518	0.912	0.888	0.75/0.62	0.679
				Y	0.841	0.857	0.71/0.43	0.534	0.980	0.969	0.94/0.88	0.908
		547 (32.8%)	Method	N	0.758	0.764	0.76/0.40	0.524	0.910	0.867	0.80/0.78	0.790
				Y	0.754	0.755	0.75/0.39	0.511	0.909	0.858	0.81/0.75	0.778
		554 (33.2%)	Result	N	0.822	0.782	0.80/0.47	0.591	0.894	0.845	0.79/0.73	0.762
				Y	0.826	0.785	0.80/0.46	0.586	0.905	0.846	0.77/0.76	0.763
		249 (14.9%)	Concl	N	0.820	0.870	0.75/0.19	0.306	0.851	0.882	0.68/0.42	0.520
				Y	0.823	0.868	0.70/0.20	0.307	0.974	0.954	0.86/0.82	0.840
	90,665	14,248 (15.7%)	Intro	N	0.876	0.890	0.80/0.41	0.545	0.933	0.924	0.80/0.92	0.746
				Y	0.873	0.890	0.79/0.41	0.541	0.975	0.967	0.92/0.97	0.892
		25,826 (28.5%)	Method	N	0.846	0.813	0.77/0.49	0.600	0.939	0.891	0.80/0.82	0.811
				Y	0.832	0.811	0.77/0.48	0.594	0.942	0.895	0.81/0.83	0.820
		34,671 (38.2%)	Result	N	0.831	0.786	0.78/0.61	0.687	0.929	0.871	0.81/0.86	0.835
				Y	0.816	0.783	0.78/0.60	0.678	0.922	0.860	0.81/0.83	0.821
		12,805 (14.1%)	Concl	N	0.880	0.893	0.73/0.36	0.478	0.939	0.918	0.74/0.63	0.682
				Y	0.850	0.889	0.72/0.35	0.469	0.991	0.970	0.88/0.91	0.895

Table 1 – Model performance on cross validated data.

* Data tested using 10 fold cross validation. For any given model 90% of all cases of numbers reported were used in model generation and 10% were used for holdout test set.

were fixed similar to the other hand labeled dataset. Models were evaluated on both the reviewed and un-reviewed, smaller, structured datasets. However, model performance was nearly identical so only the reviewed dataset is reported. Finally, models trained on the larger corpus of sentences from structured abstracts were tested on sentences from unstructured abstracts. The hope was that we might be able to leverage the larger set of tagged, but un-reviewed data from the structured dataset to reliably identify sentences in larger body of unstructured abstracts, avoiding costly manual tagging to build the initial model.

RESULTS

Table 1 outlines the results of the experiment on the models tested on cross-validated data. Several important trends are worth noting. First, in all experiments, the support vector machine performs better (often dramatically better) than the linear classifier on equivalent training data. Next, in all nearly all experiments, the increased number of training data improves SVM performance. The addition of the sentence location feature improved classification for the 'introduction' and 'conclusion' types but is of little help for the 'methods' and 'results' types. Interestingly, the addition of sentence location did not seem to help the linear classifier models.

Table 2 outlines the performance of models trained with the larger, structured abstracts tested on the unstructured abstracts. In all cases except the 'introduction' type performance is similar to that of models trained on unstructured abstracts.

Model Type	Sentence Location	SVM				
		ROC	Acc	P	R	F_1
Intro	N	0.838	0.874	0.75	0.02	0.030
	Y	0.871	0.896	0.63	0.45	0.524
Method	N	0.899	0.877	0.89	0.64	0.744
	Y	0.921	0.897	0.88	0.73	0.799
Result	N	0.905	0.820	0.78	0.83	0.804
	Y	0.935	0.872	0.84	0.88	0.861
Concl	N	0.918	0.909	0.70	0.63	0.665
	Y	0.946	0.941	0.83	0.75	0.785

Table 2 – Performance of models trained on Structured data, tested on unstructured data.

DISCUSSION

This study demonstrates several important findings. As has been found by several others, the SVM models performed dramatically better than traditional linear classifiers. This difference is probably more dramatic in our study due to the sparseness of the vectors and increased abstraction of the class labels. While the domain of this study was deliberately quite small and the generalizability to other domains may be tenuous, it is important to note that machine learning techniques can be successfully used to learn contextual information from content. It provides further evidence for the robust performance of support vector machine in text domains.

The addition of a sentence location feature made significant improvement in performance of several SVM models. In nearly all cases training on more data generated better models, however this effect was often less dramatic than adding the sentence location feature. This supports the idea that traditional text categorization might be improved by adding additional relevant features using human insight into the problem.

Similarly, our model provides a way to add a relevant feature of sentence context that may be useful for other processing techniques. In our case, we use the sentence classifier to help choose processing models for natural language processing and summarization. It may also be useful as a screening tool in other text search processing. For example, users of MedLINE might be interested in searching for trials where the 'Methods' section contains the word 'randomized' to exclude introductory or conclusion statements that refer to other trials.

Future work using a combination of statistical and natural language approaches might substantially improve performance. For example, supplementation of the vector of words

with word pairs or simple parse trees might significantly enhance performance.

We were excited to find that models trained on structured abstracts are robust enough to provide reasonable performance on data from unstructured abstracts. In half of the models, performance of the model trained on structured data was actually better than that of the model trained on unstructured data. The poor recall (and therefore F-measure) on the 'introduction' model was due to the use of thresholds. This technique sacrifices recall for accuracy and is magnified in relatively weak models.

The use of models trained on similar but different datasets provides hope for less expensive model development. A process might be used beginning with model training on a dataset that has some similarity but in which class labels have either already been assigned or can more easily be assigned. The crude model trained on the similar dataset to screen could be used to assign preliminary class, thus facilitating manual label assignment. Iteratively, this process could build progressively larger and more robust datasets where class labels must be assigned manually.

CONCLUSION

The use of machine learning techniques for sentence type identification seems feasible within the domain of both structured and unstructured abstracts about Randomized Control Trails. The use of this contextual information may help other text summarization techniques.

REFERENCES

[1] Lewis DD, Schapire RE, Callan JP, Papka R. *Training Algorithms for Linear Text Classifiers.* Proceedings, ACM SIGIR, 1996. pp 298-306.

[2] Yang Y. *An Evaluation of Statistical Approaches to Text Categorization*. Journal of Information Retrieval, 1999. pp 69-90.

[3] Joachims T. *Text Categorization with Support Vector Machines: Learning with Many Relevant Features.* Proceedings 10th European Conference on Machine Learning, 1998. pp 137-142.

[4] Semrep reference

[5] Srinivasan P, Rindflesch *T Exploring Text Mining from MEDLINE.* AMIA Fall Symposium 2002. In press.

[6] Haynes RB, Wilczynski N, McKibbon KA, Walker CJ, Sinclair JC. *Developing Optimal Search Strategies for Detecting Clinically Sound Studies in MEDLINE.* JAMIA, 1994;1:447-458

[7] Reynar JC, Ratnaparkhi A. *A Maximum Entropy Approach to Identifying Sentence Boundries*. Proceedings, 5th Conference on Applied Natural Language processing, 1997. pp 16-19.

[8] Ruiz ME, Srinivasan P. Hierchical Neural Networks for Text Categorization. Proceedings ACM SIGIR, 1999. pp 281-282.

[9] McNichol D. *A Primer of Signal Detection Theory*. London: Allen and Unwin, 1972.

[10] van Ruhsbergen C.J. *Information Retrieval*. Butterworths, London, second edition, 1979. pp 173-176