Information Extraction from Biomedical Text

Aly Valliani
Swarthmore College
500 College Avenue
Swarthmore, PA 19081, USA
avallia1@swarthmore.edu

Abstract

This paper details a supervised machine learning classifier for information extraction from biomedical texts. The system extends upon TEES 2.1, the highest performing system in the 2013 Cancer Genetics task, by utilizing SVM and two random forest classifiers in conjunction with the TEES system's preprocessing techniques. The resulting system obtained accuracies of 88%, 88% and 85% using the SVM and unbatched and batched random forest classifiers, respectively. Further experimentation yielded interesting contrasts between the three classifiers and relevant trade offs that will form the basis for future research.

1 Introduction

Cancer genetics is a complex phenomenon involving multiple molecular pathways and various patterns of mutation (Balmain et al., 2003; Balmain, 2002). The explosion of genome sequencing and molecular profiling techniques have increased our understanding of the molecular mechanisms of cancer and greatly expanded the scientific literature: a PubMed query for cancer returns 3.1 million scientific articles, with 166,000 citations from 2014. Therefore, clinical text-mining systems are integral to automate the maintenance of comprehensive and up-to-date databases on cancer genetics.

The BioNLP Shared Task (BioNLP-ST) series has been instrumental in promoting a community-wide trend in information extraction of biomedical text. Historically, task efforts were almost exclusively focused on the identification of normal physiological processes and molecular-level entities and events (Kim et al., 2011; Kim et al., 2011b). The Cancer Genetics (CG) task organized as part of BioNLP-ST 2013 expands upon existing identification criteria by incorporating higher level biological processes, such as mutation, cell proliferation, apoptosis, angiogenesis and metastasis, to promote the development of systems able to generalize at all levels of pathological processes. The overall goal of the CG task is the automatic extraction of information from biomedical text on biological processes relating to the development and progression of cancer. In particular, it involves the extraction of events to enable the identification of complex causal relationships between entities in the following manner:

Given: labeled entities of biological relevance

Do: identify biological relationships among entities

Further task-specific information can be found in the task description paper (Pyysalo et al., 2013).

Six different teams from diverse biological and linguistic backgrounds participated in the CG task yielding six systems employing a variety of preprocessing, information extraction and machine learning techniques. The highest performance was achieved by the previously established machine learning based Turku Event Extraction (TEES) system, with an F-score of 55%. This paper describes a supervised machine learning classifier for the extraction of information from biomedical text that extends upon the TEES system. In particular, it compares the performance of a support vector machine

(SVM) and two random forest classifiers in the CG task.

The paper is organized as follows. Section 2 describes the data sets, machine learners and evaluation metrics that were used to conduct the analysis. Sections 3 and 4 describe evaluation results and avenues for future work, respectively.

2 Methods

2.1 Data Sets

All utilized data was provided by the CG task organizers who curated data sets from 600 PubMed abstracts. Provided data contained preliminary entity annotation via automatic named entity and entity mention taggers such as BANNER (Leaman and Gonzalez, 2008), NERsuite[1] and LINNAEUS (Gerner et al., 2010).

Pre-processing was performed using the open-source TEES 2.1 pipeline submitted by Björne and Salakoski as part of BioNLP-ST 2013, which utilized the Porter stemmer (Porter, 1980), McClosky-Charniak-Johnson parser (McClosky, 2009), and Stanford Dependency converter (de Marneffe et al., 2006) to generate part of speech tagged words organized in a bag-of-words representation. The system's built-in EdgeExampleBuilder class was used to convert from the native Interaction XML file format to edge machine learning examples and feature vectors. The resulting example file follows the SVM-multiclass format where each example begins with a class id followed by a feature vector (feature id:value pairs) separated by a hash mark designating the comments section. A total of 36,684 edge examples containing 10 classes were utilized. Evaluation was conducted using an 80%/20% training/testing split.

2.2 Machine Learning

Support vector machine (SVM) and random forest classifiers were used to conduct the analysis. This is in contrast to the TEES system, which utilized three layers of SVM classifiers with each layer designed for edge detection, entity labeling and negation detection, respectively. The current system's SVM classifier utilizes a linear kernel that plots a hyperplane that best separates data into appropriate classes. It supports multiclass classification and was optimized to perform on classification tasks in which the number of samples is larger than the number of features.

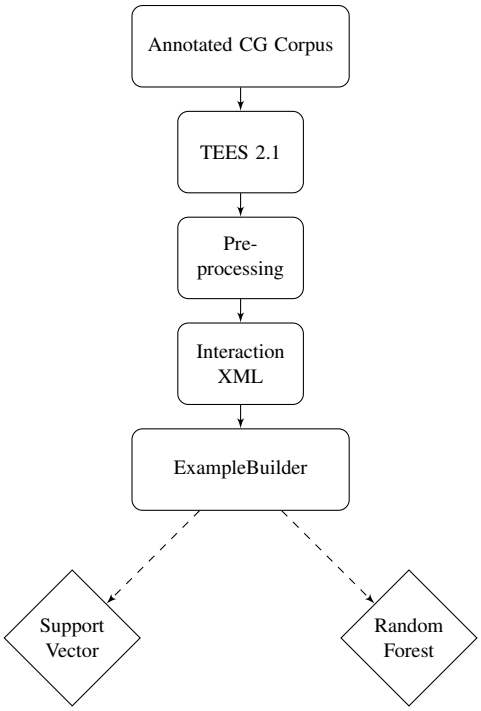

Figure 1: Classifier Workflow

In contrast, the random forest classifier utilizes an ensemble of decision trees whose prediction probabilities are averaged to make an appropriate classification. Since the number of samples exceeded computational power when building random forests of size greater than 100 trees, the random forest classifier was trained using two different methods. In the first case, forests of 100 trees were trained on chunks of training data across all training samples. The resulting forests were combined during classification. In the second case, batch processing was used to train a single forest of 100 trees on one chunk of training data at a time across all training samples. With each unique chunk of training data, random forest parameters were adjusted. Upon completion of training, the resulting forest was used for classification. In each case, the effect of varying forest sizes and tree heights on performance was analyzed. Additional parameters dealing with the relative importance of features and the size of splits per node will be analyzed in future experiments.

[1] http://nersuite.nlplab.org

Table 1. Edge Detection Experimental Results

Edge Classes	# Training Examples	# Testing Examples	Support Vector Machine			Random Forests (Unbatched, Batched, n=100)		
-	-	-	Precision	Recall	F-score	Precision	Recall	F-score
Neg	20508	5403	0.916	0.929	0.922	0.883, 0.864	0.979, 0.979	0.929, 0.918
Theme	6124	1393	0.806	0.834	0.820	0.818, 0.793	0.703, 0.633	0.756, 0.704
Cause	1580	362	0.736	0.655	0.693	0.982, 0.920	0.307, 0.191	0.467, 0.316
AtLoc	198	49	0.484	0.273	0.349	1.000, 0.000	0.082, 0.000	0.151, 0.000
Site	165	24	0.714	0.385	0.500	1.000, 0.000	0.083, 0.000	0.154, 0.000
SiteParent	117	10	1.000	0.091	0.167	1.000, 0.000	0.100, 0.000	0.182, 0.000
Instrument	440	62	0.709	0.622	0.663	0.917, 0.920	0.355, 0.371	0.512, 0.529
ToLoc	110	19	0.938	0.652	0.769	1.000, 0.500	0.053, 0.105	0.100, 0.174
FromLoc	48	2	0.429	0.375	0.400	0.000, 0.000	0.000, 0.000	0.000, 0.000
Participant	57	13	0.909	0.606	0.727	1.000, 0.000	0.692, 0.000	0.818, 0.000
Accuracy	-	-	**0.878**			**0.875, 0.854**		

Neg: modifier that detects negations among relationships.
Theme: entity undergoing the primary effect of an event.
Cause: entity responsible for the event's occurrence.
AtLoc: anatomical location where the event takes place.
Site: specific part of the theme that is affected.
SiteParent: connects the entity and the protein it belongs to.
Instrument: specifies a mechanism of event detection.
ToLoc: anatomical destination of the particular event.
FromLoc: anatomical orgin of the particular event.
Participant: entity whose precise role is unstated.

All classifiers were implemented using the scikit-learn library[2]. Training data was passed into each machine learner in the form of an NxM sparse matrix organized as a dictionary of keys, with N representing the number of examples and M representing the number of features[3]. Figure 1 illustrates the classifier workflow.

2.3 Evaluation Metric

The metrics used for evaluation were accuracy, precision, recall and F-score. They can be calculated as follows:

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FP}}$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

$$\text{F-score} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$$

[2]http://scikit-learn.org/stable/
[3]http://docs.scipy.org

where TP and TN represent classes that were accurately predicted and FP and FN represents classes predicted inaccurately.

3 Results

As indicated earlier in section 2.2, the proposed system utilizes an SVM and two versions of the random forest classifier for experimentation. Each classifier was used to perform event detection on preprocessed examples obtained from the TEES system (Table 1). In accordance with expectations, all three classifiers perform well (F-score > 0.70) on overrepresented classes, such as "Neg" and "Theme" and perform poorly on sparsely represented anatomical classes, such as "AtLoc", "Site", "SiteParent", "FromLoc" and "ToLoc." High precision in combination with low recall values for underrepresented classes suggests that classifiers are reluctant to label novel examples as such absent certainty. Interestingly, unlike the SVM and unbatched random forest classifier, the batched classifier often avoided labeling examples as sparsely represented classes altogether, thereby indicating the heightened importance of probabilistic confidence and the tendency of ensemble learn-

ers to maximize recall over precision. In contrast, the unbatched random forest classifier likely generated more specialized clusters of trees that aided in the classification of less represented classes. Overall, all classifiers performed with high accuracy due to their ability to classify densely represented classes with high precision and recall.

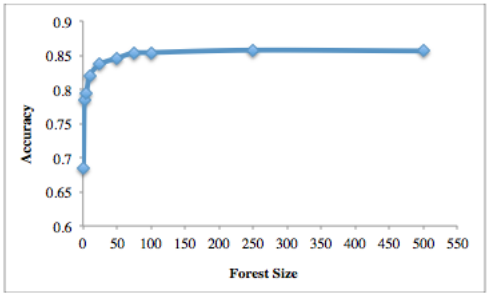

Figure 2: Random forest classifier accuracy across ten different forest sizes. Results depict a steep learning curve indicative of the random forest classifier's ability to perform near maximal accuracy with few estimators.

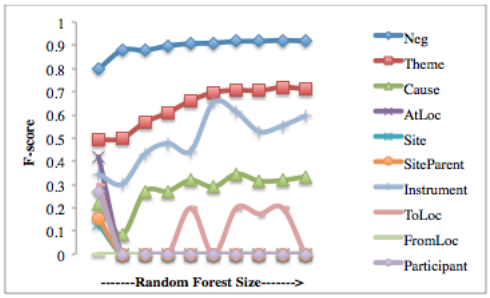

Figure 3: Random forest classifier performance on event classes across ten different forest sizes. Results indicate negligible improvement on well-represented classes with increasing forest size but substantial improvement on underrepresented classes, such as "AtLoc", "SiteParent", "ToLoc", "FromLoc", with decreasing forest size.

Additional experiments tested the effects of various forest sizes and tree heights on accuracy and the ensemble learner's ability to classify different event classes. Figure 2 depicts a learning curve generated by random forest classifiers containing populations of trees ranging from 1 to 500. Results indicate the random forest classifier's ability to obtain close to maximal accuracies with as few as 50 estimators thereby obviating the need to generate large forests of trees that exhibit severe run-time costs. Figure 3 depicts the random forest classifier's performance as a function of forest size across the 10 event classes. The general trend indicates that larger forest sizes negligibly increase overall performance on well-represented classes. On the contrary, as depicted by the lower trend lines, smaller forest sizes substantially increase performance on sparsely represented classes. A possible explanation for such a phenomenon may be that smaller forest sizes are less susceptible to probabilistic confidence levels that account for levels of class representation since the classification decision made by a single tree has more influence in the overall decision of the forest and is not flooded by the probabilistic decisions of many trees. Results on the effect of tree height, however, indicate the random forest classifier's ability to obtain near maximal accuracy with tree heights of 100 or greater (Figure 4). In addition, unlike forest size, lower tree height does not increase classification performance on sparse data (Figure 5). Regardless, the results provide compelling evidence of a trade off between forest size and efficacy of classification that warrants further study.

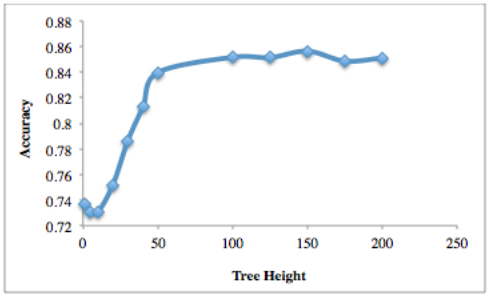

Figure 4: Random forest classifier performance on event classes across various tree heights. Results indicate a shallow learning curve with maximal accuracies obtained using trees of heights greater than or equal to 100.

4 Conclusion

This paper presented a supervised machine learning classifier for clinical text mining that borrows from the TEES system while incorporating new modes of classifications. The SVM and the two random forest classifiers utilized within the system conduct event detection with accuracies of 88%, 88% and 85% (batched), respectively, but are poor predictors of sparsely represented classes. Moreover, it was

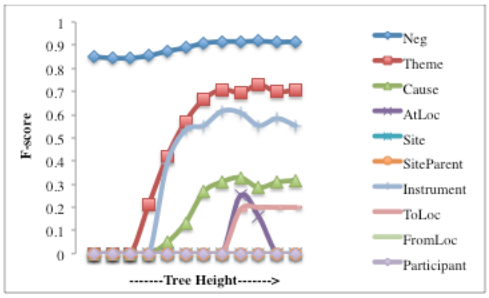

Figure 5: Random forest classifier performance on event classes across various tree heights. Results indicate substantial improvement on all classes with increasing tree height.

determined that each classifier differed in performance on sparsely versus densely represented data with the SVM maximizing precision and the ensemble learner maximizing recall. Further analysis of the random forest classifier yielded valuable insight into a trade off between forest size and performance in labeling underrepresented classes.

As such, future works will focus on efforts to navigate this trade off to enable the reliable detection of all classes regardless of representation. Potential solutions may involve further optimization of classifier parameters. Future efforts will also emphasize the conversion of existing metrics to official scores provided by the BioNLP-ST 2013 task committee to gauge the performance of this system in relation to those in existence.

Acknowledgments

I would like to thank Professors Ameet Soni and Richard Wicentowski for giving me the opportunity to pursue an independent project and for devoting their time to provide assistance, suggestions, and insights during the course of this project.

References

Allan Balmain, Joe Gray, and Bruce Ponder. 2003. The genetics and genomics of cancer. *Nature Genetics*, 33:238–244.

Allan Balmain. 2002. Cancer as a complex genetic trait: Tumor susceptibility in humans and mouse models. *Cell*, 108(2):145–152.

Marie-Catherine de Marneffe, Bill MacCartney, and Christopher D. Manning. 2006. Generating typed dependency parses from phrase structure parses. In *In Proceedings of LREC*, pages 449–454.

Martin Gerner, Goran Nenadic, and Bergman Casey M. 2010. Linnaeus: a species name identification system for biomedical literature. *BMC bioinformatics*, 11(85).

Jin-Dong Kim, Tomoko Ohta, Sampo Pyysalo, Yoshinobu Kano, and Jun'ichi Tsujii. 2011. Extracting biomolecular events from literature - the bionlp'09 shared task. *Computational Intelligence*.

Jin-Dong Kim, Sampo Pyysalo, Tomoko Ohta, Robert Bossy, and Jun'ichi Tsujii. 2011b. Overview of bionlp shared task 2011. In *Proceedings of BioNLP Shared Task 2011*, pages 1–6, Portland, Oregon, June. Association for Computational Linguistics.

Robert Leaman and Graciela Gonzalez. 2008. Banner: an executable survey of advances in biomedical named entity recognition. In *Proceedings of the Pacific Symposium on Biocomputing*, pages 652–663.

David McClosky. 2009. Any domain parsing: Automatic domain adaptation for natural language parsing.

Martin F. Porter. 1980. An algorithm for suffix stripping. *Program*, 14(3):130–137.

Sampo Pyysalo, Tomoko Ohta, and Sophia Ananiadou. 2013. Overview of the cancer genetics (cg) task of bionlp shared task 2013. In *Proceedings of the BioNLP Shared Task 2013 Workshop*, pages 58–66, Sofia, Bulgaria, August. Association for Computational Linguistics.