iClass – Applying Multiple Multi-Class Machine Learning Classifiers combined with Expert Knowledge to Roper Center Survey Data

Marmar Moussa[1] Marc Maynard[2]

[1]University of Connecticut, CT, USA
marmar.moussa@uconn.edu
[2]Roper Center for Public Opinion Research, CT, USA
mmaynard@ropercenter.org

Abstract. As one of the largest public opinion data archives in the world, Roper Center [1] collects datasets of polled survey questions as they get released from numerous media outlets and organizations with varying degrees of format ambiguity. The volume of data introduces search complexities over survey questions asked since the 1930s and poses challenges when analyzing search trends. Up to this point, Roper Center question-level retrieval applications used human metadata experts to assign topics to content. This has been insufficient to reach required levels of consistency in catalogued data, and provides an inadequate base for creating an advanced search experience for research clients.

The objective of this work is to combine the human expert teams' knowledge of the nature of the poll questions and the concepts and topics these questions express, with the ability of multi-label classifiers to learn this knowledge and apply it to an automated, fast and accurate classification mechanism. This approach cuts down the question analysis and tagging time significantly as well as provides enhanced consistency and scalability for topics' descriptions. At the same time, creating an ensemble of machine learning classifiers combined with expert knowledge is expected to enhance the search experience and provide much needed analytic capabilities to the survey question databases.

In our design, we use classification from several machine learning algorithms like SVM and Decision Trees, combined with expert knowledge in form of handcrafted rules, data analysis and result review. We consolidate this into a 'Multipath Classifier' with a 'Confidence' point system that decides on the relevance of topics assigned to poll questions with nearly perfect accuracy.

Keywords: ensembles; expert knowledge; knowledge base; machine learning; multi-label classifiers; supervised learning; survey datasets

1 Introduction

In this paper, we present an overview of our work at the Roper Center in applying machine learning to the public opinion survey datasets in an attempt to classify the questions to their respective most relevant set of topics/classes. The application iClass is a collection of modules of autonomous classifiers and a knowledge expert 'Admin' module which allows us to combine both human knowledge and machine learning to the classification and review processes. This paper presents the nearly complete first phase of iClass. We describe the business context and motivation behind this development, a design overview and preliminary evaluation results. The last section describes the business payoff and trends for future phases.

1.1 Context and Motivation

The Roper Center collects datasets of survey questions from polls performed by think tanks, media outlets, and academic organizations. The data has been gathered since the 1930s with varying degrees of format ambiguity. The volume of legacy data introduces search complexities and poses challenges when analyzing search trends.

Homegrown backend systems serve up several data retrieval and analysis services to the Roper Center members. The primary two are 1) iPOLL, a question-level retrieval database containing over 650,000 polling questions and answers, and 2) Roper*Express*, a catalog of survey datasets conducted in the US and around the globe. Historically, datasets, iPOLL questions, and secondary material have been managed and cataloged by separate teams, which led to different descriptive practices. Dataset expert teams use free text key-word descriptors to assign topics to content. This means, even though there are clear topical and other kinds of connections among the content, lack of consistent description creates rifts, making these connections elusive (Fig.1). It results in costly string operations for even simple tasks, as well as costly retrospective updates to topics definitions and adding new topics. This approach also does not allow for any further data analytics capabilities.

ID	Question Text	Existing Topics' String
157820	(Now let me ask you about a few specific federal agencies. Using this card is your opinion of them highly favorable or moderately favorable, or not too favorable or rather unfavorable?)...O.S.H.A. (Occupational Safety and Health Administration)	GOVERNMENT RATINGS WORK REGULATION
157835	(Now I'm going to name some things, and for each one would you tell me whether you think there is too much government regulation of it now, or not enough government regulation now, or about the right amount of government regulation now?)...Health and safety of working conditions	REGULATION WORK HEALTH

Fig. 1. Example of inconsistent topics assigned to 'similar' content

Our objective is therefore to develop a scalable system for concept-based classification of questions that implements an intelligent automated approach for identifying conceptual links between content at point of acquisition/creation using machine learning classifiers while at the same time leverage existing expert knowledge.

1.2 Related Work

In statistics and machine learning, ensemble methods achieve performance by combining opinions of multiple learners [2]. There are different ways of combining base learners into ensembles [3]. We decided to design a combining method that is tailored to our specific goals like scalability and utilizing available expert knowledge. This is required to accommodate changes in topic definitions over time and the emergence of new topics from newly acquired studies. Our combining method is a mix of weighting, majority voting and performance weighting. In weighting methods a classifier has strength proportional to its assigned weight. In a voting scheme, the number of classifiers that decide on a specific label is counted and the label with the highest number of votes is considered. For performance weighting [4], the weight of each classifier is set proportional to its accuracy performance on a given validation set.

2 Design Overview

2.1 Data Analysis and Tools

Several housekeeping steps had to take place before we would be able to develop a reliable system with high accuracy. The first was performing a data cleanup. The initial classification tests revealed numerous discrepancies and inconsistencies between the actual concepts of questions and assigned topics as described in Section 1.1.

Also, the review revealed the need for a number of new topics and a three-level topic hierarchy. This meant defining categories at the parent level in a new topics hierarchy (Fig.2), as well as refining existing topic definitions to achieve consistency. The effort resulted in 119 topics for the current question bank, with over 20 new topics, identified as a result of initial classification test reviews and analysis. The topics were arranged into 6 main categories and 3 levels of hierarchy. We also needed to implement necessary workflow changes to include testing results review, a review of the ‘Before & After’ list of topics associated with each question. An ‘Admin’ role with the necessary expert knowledge reviews the result and ‘approves’ the topics assigned and selects some of the accepted question-topic pairs to be fed back into the training set.

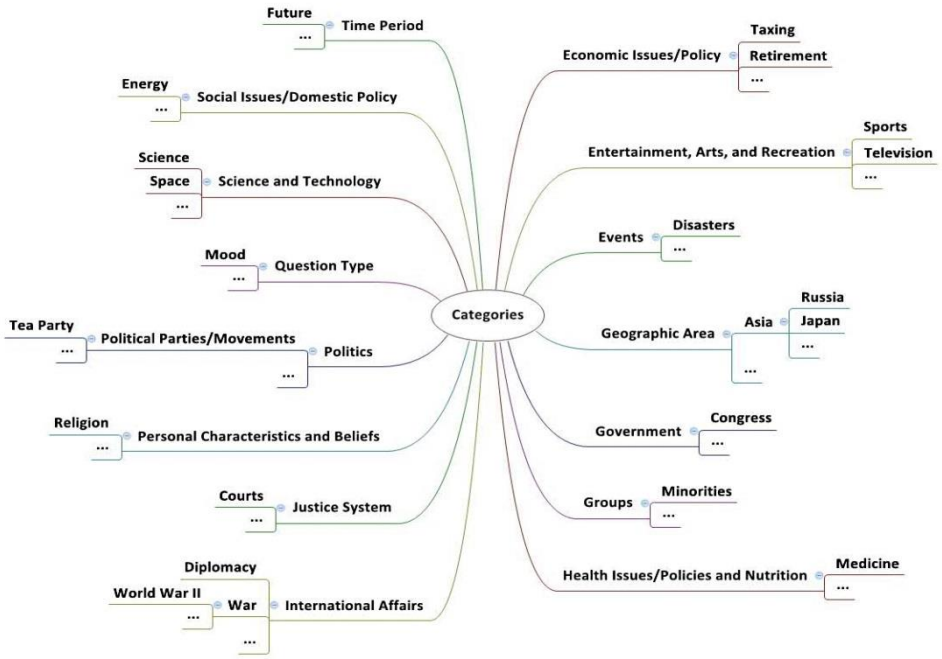

Fig. 2. New Categories Hierarchy

Roper Center metadata is stored in an Oracle 11g database, prompting an examination of machine learning algorithms supported by Oracle classifier functions. We conducted tests using the RTextTools package over datasets exported from the Oracle Database [5], also tests and evaluation using python scikit-learn package over exported data [6]. The main modules however used Oracle Pl/SQL for analysis as well as the training and classification for compatibility with the Roper Center's architecture.

2.2 Machine Learning Algorithms

We used two machine learning techniques for this phase of iClass, Support Vector Machine (SVM) and Decision Tree. SVM is known to perform well with significant accuracy, even with sparse data, also SVM classification attempts to separate target classes with the widest possible margin, and is very fast. Distinct versions of SVM use different kernel functions to handle different types of data sets. Linear and Gaussian (nonlinear) kernels are supported in SVM. We used linear kernels in this phase of iClass. SVM however does not produce human readable rules. In contrast, the Decision Tree (DT) algorithm produces human readable and extendable rules. Decision trees extract predictive information in the form of human-understandable rules. The rules are nearly if-then-else expressions; they explain the decisions that led to the prediction. DT has good missing value interpretation, is fast and performs with good accuracy [7].

3 Design Details

3.1 Classification Process Flow

The assembling of a comprehensive training set that represents all topics and their features was challenging yet critical for success. The data analysis and initial tests resulted in a selected set of expert-classified questions to use as the seed for the training set. The training set also included handcrafted question samples for under-represented topics. For new topic definitions, we used a set of SQL queries for a fine-grained selection of questions to be assigned the new topics.

After the training set is constructed, SVM and Decision Tree classifiers are trained to produce a set of rules for each topic. Each topic also gets an additional set of Admin/Expert-defined rules in the form of keywords to look for or exclude from the question text. These manually defined rules formed the third set of rules to process.

Three modules (DT, SVM and Rule-Based Classifiers) are created to use these sets of rules and 'vote' with different scores over the topics to be assigned. A fourth path for classification is formed by the direct SQL queries representing the more complex expert defined rules that are not included in the Rule-Based Classifier. For this path too, the implementation assigned confidence scores to the selected question-topic pairs. The four paths' (sources) results construct a vector for each question and topic pair, containing the source and the designated score/confidence.

(Fig.3) below provides a description of this process flow in iClass. Three values are then considered in combining the information from this ensemble of classifiers: 1) the (weighted) number of sources/votes that classified a topic to a specific question, 2) the threshold (possible one for each topic and source) that would consider this classification true positive or false positive, and 3) the confidence/score values.

A combined confidence/score is formed and then the classified question-topic pairs are reviewed by an expert to approve or reject. Approved results can then be fed back to the training set pool for a new round of training. This is needed as the dataset grows with incoming poll questions from newer studies acquired by the Roper Center.

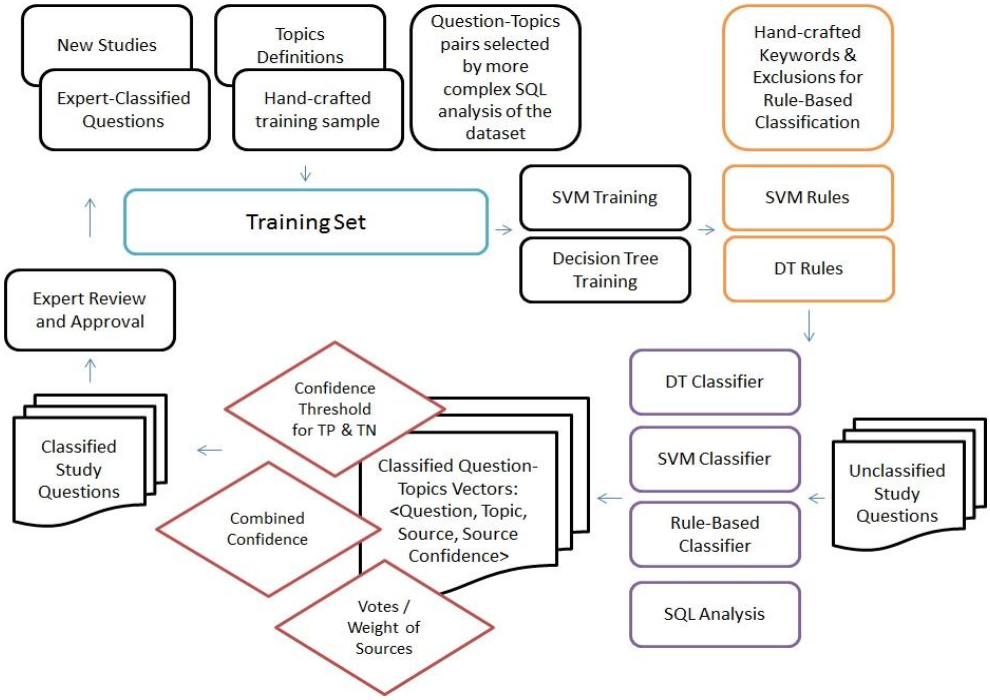

Fig. 3. iClass Components and Process Flow details

3.2 Confidence

Fig. 4. Confidence Points 1(low) to 12(high)

As described in previous sections, iClass current phase has four different sources of classification: SVM, DT, Rule-Based and Expert/Manual direct selection paths. To combine them, we applied a 'Confidence Points System'. Confidence/relevance levels (Low, Medium, and High) from each classification algorithm/path aggregate to an (N*3) point system, where N is the number of classification paths. As we currently implement 4 paths, there are 12 Points of Confidence (Fig.4).

For sources 1 and 2, SVM and DT, the confidence is calculated via the Classifier functions as a value > 0 and <100, we convert this to a value 1→ 3 by using a pseudo-count, scaling, and rounding. For the Expert/Manual path, where the direct analysis process is implemented in the various SQL scripts, the confidence for each topic is configured directly based on the Admin's analysis. For the Rule_Based Classifier, each topic is assigned a rule confidence level associated with the keywords and exclusion rules defined for that topic. A question-topic classified pair has therefore 1 → 12 possible confidence points: if 4 sources vote for a topic with the max confidence points (3) each, then the total confidence for this question-topic classification is 12. If

on the other hand only one source votes for this assignment and with the lowest confidence possible (1), the total will be 1 point (Fig.5). The Admin sets different thresholds for different functionalities, for instance a threshold >3 to appear in search results, a threshold of ≤2 for admin review process to look at the weakest items.

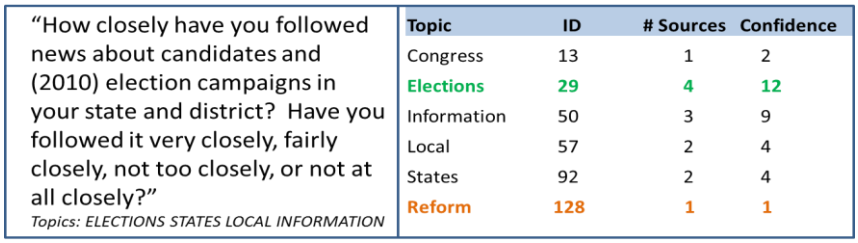

Fig. 5. Example of Question-topics assigned with highest and lowest confidence.

A tradeoff exists between adding more (maybe distantly related) topics which could cause a degree of confusion to the reviewer/user versus being extra cautious in assigning topics and risking that related questions might not appear in results of related but not main topic searches. (Fig.6) is an example of this tradeoff, and is also an example of how iClass identified more relevant topics than were assigned by a human cataloger. 'Family' and 'Religion' topics in this example, although both are topics long available in the system, were not initially assigned by manual classification during data entry. iClass assigned lower confidence to these topics compared to other more relevant topics, like 'Abortion' and 'Courts'. Topic 'Supreme Court' is a new topic. It is also very relevant and is correctly captured and assigned a high confidence level.

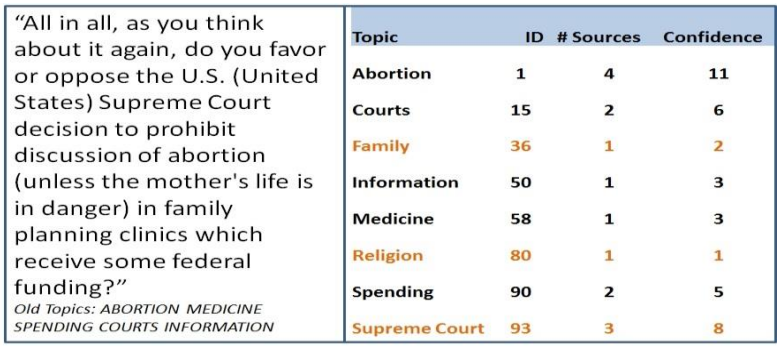

Fig. 6. Example of the new classification results

4 Evaluation

The evaluation of classical multiclass classifiers is by nature challenging, as most of the metrics usually make the most sense when applied to binary classifiers. One way

to explore the performance of a multiclass classifier is to construct the confusion matrix (Fig.7) and extend it to (NxN) matrix, where N is the number of classes (topics).

Acuuracy = (TP+TN) / Total		actual result/classification	
		yes	no
predictive result/ classification	yes	TP (true positive	FP (false positive)
	no	FN (false negative)	TN (true negative)

Fig. 7. Binary Confusion Matrix

Aside from having multi-classes in our system, we have a further complexity; a question can be assigned multiple topics with varying confidence/relevance levels. A threshold then determines whether or not questions with lower confidence points are counted towards FP or TP. The cutoff between TP and FP is therefore a little blurred.

Results based on SVM, DT & Rule-Based Classifier modules only:	
Hits: Avg. # of Questions with all topics TP (657,850 total Questions)	576,902
Average Accuracy ((TP+TN)/total)	**0.917**
False Negative/Miss Rate	0.026
"False Positive" Rate	0.030
# Newly identified Question-Topic pairs (Not present in training set)	695,792
# New correct topic assignments rate (added value)	0.455

Fig. 8. Evaluation Results

Our evaluation of only 3 paths, the SVM, DT and Rules_Based Classifier results showed accuracy of 91.7% and over 99% when enabling all 4 paths. This is expected as the 4th path involves more direct human knowledge over the classification decision.

5 Conclusion & Future Work

The development of the first phase of iClass, combining machine learning and expert knowledge, introduced performance and administrative benefits to the business process at the Roper Center. To name a few, the automated classification contributed to better consistency in topics definition and faster, streamlined topics assignment. The expert/Admin review process is dramatically shortened as it is focused on low confidence items. The process is now change-tolerant; when adding/updating topics, we can reflect the updates over the entire questions' bank retrospectively. In terms of performance enhancement, the elimination of costly string operations improves functionalities like search and navigation by topics. Although still a work in progress, iClass is scalable in terms of thresholds, confidence level configurations as well as

adding entire extra classification paths to the system. Analytic capabilities are now part of the system, like efficient metadata statistics, especially about topics trends.

From the application perspective, several components of iClass need further work in next phase, a more user-friendly Admin module is planned, the system currently supports only one set of handwritten rules per topic definition, as well as only one admin user, which needs to be extended for business needs. The classification of the datasets only lays the groundwork for better data analytics, which is currently not fully leveraged. There is also the business need to extend the functionality of iClass to knowledgebase facets other than topics, such as the survey sample classes. In addition, there is still a great deal to be explored about learning techniques that best fit the business. As the classification process is prepared to accept more classification paths, part of the future work includes using other machine learning algorithms to create more classification paths, as well as study other ensemble classification methods for combining weights and votes, and compare the results of the different methods.

6 References

1. http://www.ropercenter.uconn.edu/
2. Polikar R(2006) Ensemble based systems in decision making. IEEE Circuits Syst Mag 6(3)
3. Rokach, L. (2010). "Ensemble-based classifiers". Artificial Intelligence Review Artificial Intelligence Review. February 2010, Volume 33, Issue 1-2, pp 1-39
4. Opitz D, Shavlik J (1996) Generating accurate and diverse members of a neural network ensemble. In: Touretzky DS, Mozer MC, Hasselmo ME (eds) Advances in neural information processing systems, vol 8. The MIT Press, Cambridge, pp 535–541
5. Timothy P. Jurka, Loren Collingwood, Amber E. Boydstun,Emiliano Grossman,Wouter van Atteveldt (2014) Automatic Text Classification via Supervised Learning
6. "Scikit-learn.": Machine Learning in Python — 0.17.dev0 Documentation.
7. Smola, Alex, and S.V.N. Vishwanathan. Introduction to Machine Learning. Cambridge: Cambridge UP, 2008.