Design of an Active Learning System with Human Correction for Content Analysis

Nancy McCracken
School of Information Studies
Syracuse University, USA
njmccrac@syr.edu

Jasy Liew Suet Yan
School of Information Studies
Syracuse University, USA
jliewsue@syr.edu

Kevin Crowston
National Science Foundation
Syracuse University, USA
crowston@syr.edu

Abstract

Our research investigation focuses on the role of humans in supplying corrected examples in active learning cycles, an important aspect of deploying active learning in practice. In this paper, we discuss sampling strategies and sampling sizes in setting up an active learning system for human experiments in the task of content analysis, which involves labeling concepts in large volumes of text. The cost of conducting comprehensive human subject studies to experimentally determine the effects of sampling sizes and sampling sizes is high. To reduce those costs, we first applied an active learning simulation approach to test the effect of different sampling strategies and sampling sizes on machine learning (ML) performance in order to select a smaller set of parameters to be evaluated in human subject studies.

1 Introduction

Social scientists often use content analysis to understand the practices of groups by analyzing texts such as transcripts of interpersonal communication. Content analysis is the process of identifying and labeling conceptually significant features in text, referred to as "coding" (Miles and Huberman, 1994). For example, researchers studying leadership might look for evidence of behaviors such as "suggesting or recommending" or "inclusive reference" expressed in email messages. However, analyzing text is very labor-intensive, as the text must be read and understood by a human. Consequently, important research questions in the qualitative social sciences may not be addressed because there is too much data for humans to analyze in a reasonable time.

A few researchers have tried automatic techniques on content analysis problems. For example, Crowston *et al.* (2012) manually developed a classifier to identify codes related to group maintenance behavior in free/libre open source software (FLOSS) teams. Others have applied machine-learning (ML) techniques. For example, Ishita *et al.* (2010) used ML to automatically classify sections of text within documents on ten human values taken from the Schwartz's Value Inventory. Broadwell *et al.* (2012) developed models to classify sociolinguistic behaviors to infer social roles (e.g., leadership). On the best performing codes, these approaches achieve accuracies from 60–80%, showing the potential of automatic qualitative content analysis. However, these studies all limited their reports to a subset of codes used by the social scientists, due in part to the need for a large volume of training data.

The state-of-the-art ML approaches for content analysis require researchers to obtain a large amount of annotated data upfront, which is often costly or impractical. An active learning approach which uses human correction during the steps of active learning could potentially help produce a large amount of annotated data while minimizing the cost of human annotation effort. Unlike other text annotation tasks, the code annotation for content analysis requires significant cognitive effort, which may limit, or even nullify, the benefits of active learning.

We are building an active machine learning system to semi-automate the process of content analysis, and are planning to study the human role in such machine learning systems.

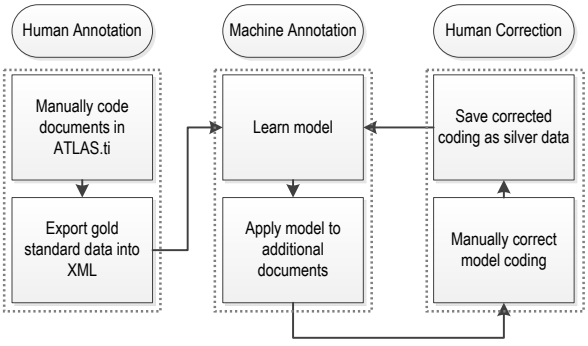

Figure 1. Active learning for semi-automatic content analysis.

As illustrated in Figure 1, the system design incorporates building a classifier from an initial set of hand-coded examples and iteratively improv-

Proceedings of the Workshop on Interactive Language Learning, Visualization, and Interfaces, pages 59–62,
Baltimore, Maryland, USA, June 27, 2014.

ing the model by having human annotators correct new examples identified by the system

Little is yet known about the optimal number of machine annotations to be presented to human annotators for correction, and how the sample sizes of machine annotations affect ML performance. Also, existing active learning sampling strategies to pick out the most "beneficial" examples for human correction to be used in the next round of ML training have not been tested in the context of social science data, where concept codes may be multi-dimensional or hierarchical, and the problem may be multi-label (one phrase or sentence in the annotated text has multiple labels). Also, concept codes tend to be very sparse in the text, resulting in a classification problem that has both imbalance—the non-annotated pieces of text (negative examples) tend to be far more frequent that annotated text—and rarity, where there may not be enough examples of some codes to achieve a good classifier.

The cost of conducting comprehensive human subject studies to experimentally determine the effects of sampling sizes and sampling sizes is high. Therefore, we first applied an active learning simulation approach to test the effect of different sampling strategies and sampling sizes on machine learning (ML) performance. This allows the human subject studies to involve a smaller set of parameters to be evaluated.

2 Related Work

For active learning in our system, we are using what is sometimes called pool-based active learning, where a large number of unlabeled examples are available to be the pool of the next samples. This type of active learning has been well explored for text categorization tasks (Lewis and Gale, 1994; Tong and Koller 2000; Schohn and Cohn 2000). This approach often uses the method of uncertainty sampling to pick new samples from the pool, both with probability models to give the "uncertainty" (Lewis and Gale, 1994) and with SVM models, where the margin numbers give the "uncertainty" (Tong and Koller 2000; Schohn and Cohn 2000). While much of the research focus has been on the sampling method, some has also focused on the size of the sample, e.g. in Schohn and Cohn (2000), sample sizes of 4, 8, 16, and 32 were used, where the result was that smaller sizes gave a steeper learning curve with a greater classification cost, and the authors settled on a sample size of 8. For additional active learning references, see the Settles (2009) survey of active learning literature.

This type of active learning has also been used in the context of human correction. One such system is described in Mandel *et al.* (2006), using active learning for music retrieval, where users were presented with up to 6 examples of songs to label. Another is the DUALLIST system described in Settles (2011) and Settles and Zhu (2012) where human experiments were carried out for text classification and other tasks. While most active learning experiments focus on reducing the number of examples to achieve an accurate model, there has been some effort to model the reduction of the cost of human time in annotation, where the human time is non-uniform per example. Both the systems in Culotta and McCallum (2005) and in Clancy *et al.* (2012) for the task of named entity extraction, modeled human cost in the context of sequential information extraction tasks. However, one difference between these systems and ours is that all of the tasks studied in these systems did not require annotators to have extensive training to annotate complex concept codes.

3 Problem

We worked with a pilot project in which researchers are studying leadership in open source software groups by analyzing open source developer emails. After a year of part-time annotation by two annotators, the researchers developed a codebook that provides a definition and examples for 35 codes. The coders achieved an inter-annotator agreement (kappa) of about 80%, and annotated about 400 email threads, consisting of about 3700 sentences. We used these coded messages as the "gold standard" data for our study. However, only 15 codes had more than 25 instances in the gold standard set. The most common code ("Explanation/Rationale/ Background") occurred only 319 times.

In our pilot correction experiments, annotators tried correcting samples of sizes ranging from about 50 to about 400. Anecdotal evidence indicates that annotators liked to annotate sample sizes of about 100 in order to achieve good focus on a particular code definition at one time, but without getting stressed with too many examples. Part of the required focus is that annotators need to refresh their memory on any particular code at the start of annotation, so switching frequently between different codes is cognitively taxing. This desired sample size contrasts with prior ac-

tive learning systems that employ much smaller sample sizes, in the range of 1 to 20.

We are currently in the process of setting up the human experiments to test our main research question of achieving an accurate model for content analysis using a minimum of human effort.

In this paper, we discuss two questions for active learning in order to have annotators correct an acceptable number of machine annotations that are most likely to increase the performance of the ML model in each iteration. These are: how do different sample sizes and different sampling strategies of machine annotations presented to human annotators for correction in each round affect ML performance?

4 Active Learning Simulation Setup

In a similar strategy to that of Clancy *et al.* (2012), we carried out a preliminary investigation by conducting an active learning simulation on our gold standard data. The simulation starts with a small initial sample, and uses active learning where we "correct" the sample labels by taking labels from the gold standard corpus. For our simulation experiments, we separated the gold standard data randomly into a training set of 90% of the examples, 3298 sentences, and a test set of 10%, 366 sentences.

In the experimental setup, we used a version of libSVM that was modified to produce numbers of distance to the margin of the SVM classification. We implemented the multi-label classification by classifying each label separately where some sentences have the selected label and all others were counted as "negative" labels. We used svm weights to handle the problem of imbalance in the negative examples. After experimentation with different combinations of features, we used a set of features that was best overall for the codes: unigram tokens lowercased and filtered by stop words, bigrams, orthographic features from capitalization, the token count, and the role of the sender of the email.

For an initial sample, we randomly chose 3 positive and 3 negative examples from the development set to be the initial training set used for all experimental runs. We carried out experiments with a number of sample sizes, *b*, ranging over 5, 10, 20, 40, 50, 60, 80 and 100 instances.

For experiments on methods used to select correction examples, we have chosen to experiment with sampling methods similar to those found in Lewis and Gale (1994) and Lewis (1995) using a *random sampling method*, where a new sample is chosen randomly from the remaining examples in the development set, a *relevance sampling method*, where a new sample is chosen as the *b* number of most likely labeled candidates in the development set with the largest distance from the margin of the SVM classification, and an *uncertainty sampling method*, where a new sample is chosen as the *b* number of candidates in the region of uncertainty on either side of the margin of the SVM classification.

5 Preliminary Results

In this simulation experiment, the pool size is quite small (3664 examples) compared to the large amount of unlabeled data that is normally available for active learning, and would be available for our system under actual use. We tested the active learning simulation on 8 codes. There was no clear winning sampling strategy out of the 3 we used in the simulation experiment but random sampling (5 out of 8 codes) appeared to be the one that most often produced the highest $F_{\beta 2}$ score in the shortest number of iterations. Figure 2 shows the $F_{\beta 2}$ score for each sampling strategy based on code "Opinion/Preference" using sample sizes 5 and 100 respectively.

As for sampling sizes, we did not observe a large difference in the evolution of the $F_{\beta 2}$ score between the various sample sizes, and the learning curves in Figure 2, shown for the sample sizes of 5 and 100, are typical. This means that we should be able to use larger sample sizes for human subject studies to achieve the same improvements in performance as with the smaller sample sizes, and can carry out the experiments to relate the cost of human annotation with increases in performance.

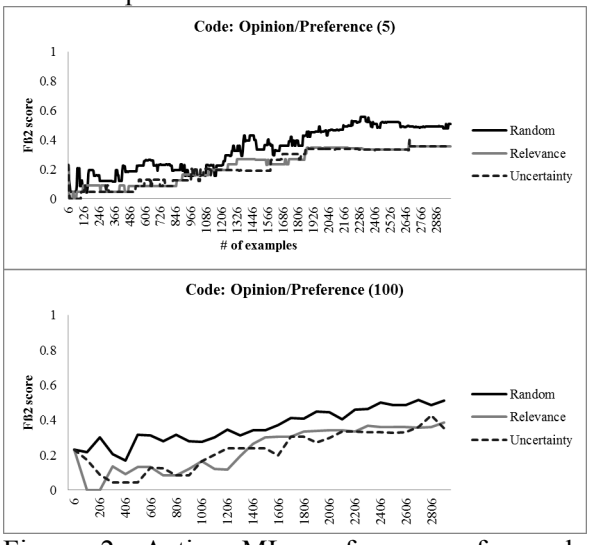

Figure 2. Active ML performance for code Opinion/Preference.

6 Conclusion and Future Work

Our findings are inconclusive as we have yet to run the active learning simulations on all the codes. However, preliminary results are directing us towards using larger sample sizes and then experimenting with random and uncertainty sampling in the human subject studies.

From our experiments with the different codes, we found the performance on less frequent codes to be problematic as it is difficult for the active learning system to identify potential positive examples to improve the models. While the system performance may improve to handle such sparse cases, it may be better to modify the codebook instead. We plan to give the user feedback on the performance of the codes at each iteration of the active learning and support modifications to the codebook, for example, the user may wish to drop some codes or collapse them according to some hierarchy. After all, if a code is not found in the text, it is hard to argue for its theoretical importance.

We are currently completing the design of the parameters of the active learning process for the human correction experiments on our pilot project with the codes about leadership in open source software groups. We will also be testing and undergoing further development of the user interface for the annotators.

Our next step will be to test the system on other projects with other researchers. We hope to gain more insight into what types of coding schemes and codes are easier to learn than others, and to be able to guide social scientists into developing coding schemes that are not only based on the social science theory but also useful in practice to develop an accurate classifier for very large amounts of digital text.

Acknowledgements:

This material is based upon work supported by the National Science Foundation under Grant No. IIS-1111107. Kevin Crowston is supported by the National Science Foundation. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation. The authors gratefully acknowledge helpful suggestions by the reviewers.

Reference

Broadwell, G. A., Stromer-Galley, J., Strzalkowski, T., Shaikh, S., Taylor, S., Liu, T., Boz, U., Elia, A., Jiao, L., & Webb, N. (2013). Modeling sociocultural phenomena in discourse. *Natural Language Engineering*, *19*(02), 213–257.

Clancy, S., Bayer, S. and Kozierok, R. (2012) "Active Learning with a Human In The Loop," Mitre Corporation.

Crowston, K., Allen, E. E., & Heckman, R. (2012). Using natural language processing technology for qualitative data analysis. *International Journal of Social Research Methodology*, *15*(6), 523–543.

Culotta, A. and McCallum, A. (2005) "Reducing Labeling Effort for Structured Prediction Tasks."

Ishita, E., Oard, D. W., Fleischmann, K. R., Cheng, A.-S., & Templeton, T. C. (2010). Investigating multi-label classification for human values. *Proceedings of the American Society for Information Science and Technology*, *47*(1), 1–4.

Miles, M. B., & Huberman, A. M. (1994). *Qualitative data analysis: An expanded sourcebook*. Sage Publications.

Lewis, D. D., & Gale, W. A. (1994). A sequential algorithm for training text classifiers. In *Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval* (pp. 3-12).

Lewis, D. D. (1995). A sequential algorithm for training text classifiers: Corrigendum and additional data. In *ACM SIGIR Forum* (Vol. 29, No. 2, pp. 13-19).

Mandel, M. I., Poliner, G. E., & Ellis, D. P. (2006). Support vector machine active learning for music retrieval. *Multimedia systems*, *12*(1), 3-13.

Schohn, G., & Cohn, D. (2000). Less is more: Active learning with support vector machines. In *International Conference on Machine Learning* (pp. 839-846).

Settles, B. (2010). Active learning literature survey. *University of Wisconsin, Madison*, *52*, 55-66.

Settles, B. (2011). Closing the loop: Fast, interactive semi-supervised annotation with queries on features and instances. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing* (pp. 1467-1478).

Settles, B., & Zhu, X. (2012). Behavioral factors in interactive training of text classifiers. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies* (pp. 563-567).

Tong, S., & Koller, D. (2002). Support vector machine active learning with applications to text classification. *The Journal of Machine Learning Research*, *2*, 45-66.