Active Learning based Weak Supervision for Textual Survey Response Classification*

Sangameshwar Patil[1,2] and B. Ravindran[1]

[1] Dept. of Computer Science and Engineering,
Indian Institute of Technology Madras, India
{sangam, ravi}@cse.iitm.ac.in
[2] TRDDC, Tata Consultancy Services, India
sangameshwar.patil@tcs.com

Abstract. Analysing textual responses to open-ended survey questions has been one of the challenging applications for NLP. Such unstructured text data is a rich data source of subjective opinions about a specific topic or entity; but it is not amenable to quick and comprehensive analysis. Survey coding is the process of categorizing such text responses using a pre-specified hierarchy of classes (often called a *code-frame*). In this paper, we identify the factors constraining the automation approaches to this problem and observe that a completely supervised learning approach is not feasible in practice. We then present details of our approach which uses multi-label text classification as a first step without requiring labeled training data. This is followed by the second step of active learning based verification of survey response categorization done in first step. This weak supervision using active learning helps us to optimize the human involvement as well as to adapt the process for different domains. Efficacy of our method is established using the high agreement with real-life, manually annotated benchmark data.

Keywords: Survey Text Mining, Active Learning, Noisy Text, Text Mining Application

1 Introduction

Surveys typically consist of two major types of questions: questions with predetermined choices, and questions with free form answers. In the literature [14] the former are referred to as closed-ended questions and the latter as open-ended questions. Closed-ended questions are typically multiple-choice objective questions where the respondents are expected to select the closest applicable answer(s) among pre-defined choices. While the close-ended questions provide a mechanism for structured feedback and enable quick analysis, the scenario is significantly different for the open-ended questions.

In case of open-ended questions, the respondents are not constrained to choose from a set of options pre-conceived by the survey designer. Such questions enable the respondents to express their opinion and feelings freely using

* A preliminary version of this paper was presented as a poster [11] at NLDB 2013.

language as the medium. Predictably, data available from responses to open-ended questions have been found to be a rich source for variety of purposes such as:

- To identify specific as well as general suggestions for improvements
- To identify topics / issues which were not covered by the closed-ended questions
- To provide additional evidence to reason about and support the findings from quantitative analysis of the closed-ended questions.

To derive broad-based insights from the subjective answers to the open-ended questions, it is necessary to convert the unstructured textual responses to quantitative form.

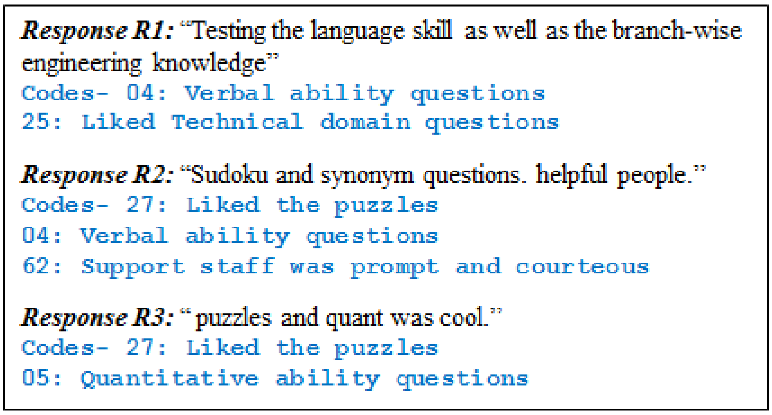
1.1 Survey Coding Process

Survey coding is the process of converting the qualitative input available from the responses to open-ended questions to a quantitative format that helps in summarization, quick analysis and understanding of such responses. The set of customer responses in electronic text format (also known as verbatims in the survey analysis parlance) and a pre-specified set of codes, called as *code-frame*, constitute the input data to the survey-coding process. A code-frame consists of a set of tuples (called code or label) of the form <code-id, code-description>. Each code-id is a unique identifier (usually numeric) assigned to a code and the code-description usually consists of a short description that "explains" the code. Table 1 shows a small, representative code-frame. Figure 1 shows the responses to an open-ended question in a survey seeking students' feedback after an aptitude test at a college. The question asked to the students who had undertaken the test was: "What do you like about the test?".

Table 1. A sample code-frame.

Code Id	Code Description
04	Verbal ability questions
05	Quantitative ability questions
25	Liked technical domain questions
27	Liked the puzzles
62	Support staff was prompt and courteous

Survey coding (also called tagging or labeling) is the task of assigning one or more codes from the given code-frame to each customer response. As per the current practice in market research industry, it is carried out by specially trained human annotators (also known as coders). The code description helps the human coder in identifying the responses to which a particular code can be assigned. Sample output of the survey-coding process is shown in Figure 1. Below each survey response, the applicable codes selected from the sample code-frame in Table 1 are given.



Response R1: "Testing the language skill as well as the branch-wise engineering knowledge"
Codes- 04: Verbal ability questions
25: Liked Technical domain questions

Response R2: "Sudoku and synonym questions. helpful people."
Codes- 27: Liked the puzzles
04: Verbal ability questions
62: Support staff was prompt and courteous

Response R3: " puzzles and quant was cool."
Codes- 27: Liked the puzzles
05: Quantitative ability questions

Fig. 1. Sample output of survey coding process: Examples of survey responses and code-IDs assigned to them. (These are responses to the open-ended question "What do you like about the test?" by the students who had just undertaken an aptitude test.)

1.2 Challenges in the Survey Coding Process

Majority of the survey responses being extempore do not follow the orthographic rules and grammatical conventions of language. The typical "noise" observed in the survey responses may be categorized as:

- **Syntactic Noise:** They are typically incomplete sentences (e.g., see Figure 1). Spelling and grammatical errors are commonplace, so are other violations such as incorrect punctuation, incorrect capitalization etc.
- **Semantic Noise:** The meaning of a word or a phrase may not be apparent due to inherent ambiguity in natural language. This could be due to multiple reasons:
    - Informal or colloquial usage of words is common. For instance, we have found many real-life examples in which verbs `cover, remove, control, treat, combat, clean, eliminate, wipe off` and even `help with` have been used in place of the verb `kill` to describe the notion *`kill germs`*.
    - Label noise: Sometimes, the descriptions of two or more codes in the code-frame could be semantically overlapping and will lead to ambiguity, e.g., `protect against germs` and `neutralizes germs` are semantically equivalent and have occurred together in a code-frame [3].

Analyzing such noisy text has proved to be challenging for existing Natural Language Processing (NLP) tools and techniques. Further, the task of survey

[3] Such scenarios are likely because code-frames typically contain more than 100 codes and are updated by human coders for coding a given set of survey responses.

coding becomes cumbersome for a human annotator due to business demands for quick turn-around time and the large amount of textual data that has to be read, understood and categorized. A more subtle and even more challenging problem with manual annotation is that it is prone to errors due to subjectivity in human interpretation. As elaborated in Section 2.2, survey coding task presents unique challenges and does not easily lend itself to automation.

In this paper, we outline existing approaches for classifying textual survey responses and their limitations in Section 2. In particular, we observe that a completely supervised learning approach is not feasible in practice and the unsupervised learning approach has severe limitations. Then, in Section 3, we present a weakly supervised approach for classifying responses to open-ended survey questions. Our approach makes use of active learning techniques to minimize the amount of supervision required. We present experimental validation of the proposed approach on real-life datasets in Section 4 and then conclude with our observations.

2 Related Work

The hardness of automating the coding problem is underscored by the advisory nature of available software that seeks human intervention as well as the apparent lack of fully automated solutions. Most of existing commercial technologies aid the human annotator to find responses that match with regular expression based pattern. These methods cannot handle responses which express same concept/feeling in other words; e.g. synonyms, hypernyms, etc. Such pattern matching methods also fall short in handling spelling errors, grammatical errors and ambiguity.

2.1 Research Literature

Academic and industry research community is well aware of the problems and challenges faced in the survey coding process for more than two decades [18], [12]. The gravity of the problem has increased exponentially with the Internet boom as well as ease and the lower cost of conducting online surveys compared to the traditional paper-pencil surveys. Over the past decade the problem has been attracting increasing attention both in the research community as well as the industry.

Research community has approached the problem of automatic code assignment from multiple perspectives. The most active research group in this area is led by Sebastiani and Esuli et al. [15, 5, 8, 3]. They approach the survey coding problem primarily using supervised learning. They formulate the problem of coding as a classification problem and try to learn the model using pre-coded open answers from the survey responses. They have compared the results of three approaches [5]: dictionary based approach, naïve Bayesian, and SVM. According to their observations, supervised learning methods provide more accurate and stable results than the dictionary based approach. At the same time, naïve Bayesian technique outperforms SVM by small margin. Esuli and Sebastini [2] have also used active learning to get positive and negative samples of each code and use it as training data to develop supervised learning techniques for automated survey coding.

Li and Yamanishi [7] apply classification rules and association rules to categorize survey responses for a car brand image survey. They use stochastic complexity to learn the classification rule and association rules. Classification rules are in the form of IF THEN ELSE and association rules are in the form of IF THEN OR. Given a phrase or a word in the phrase from the open answer, the decision rule assigns the target to the textual response.

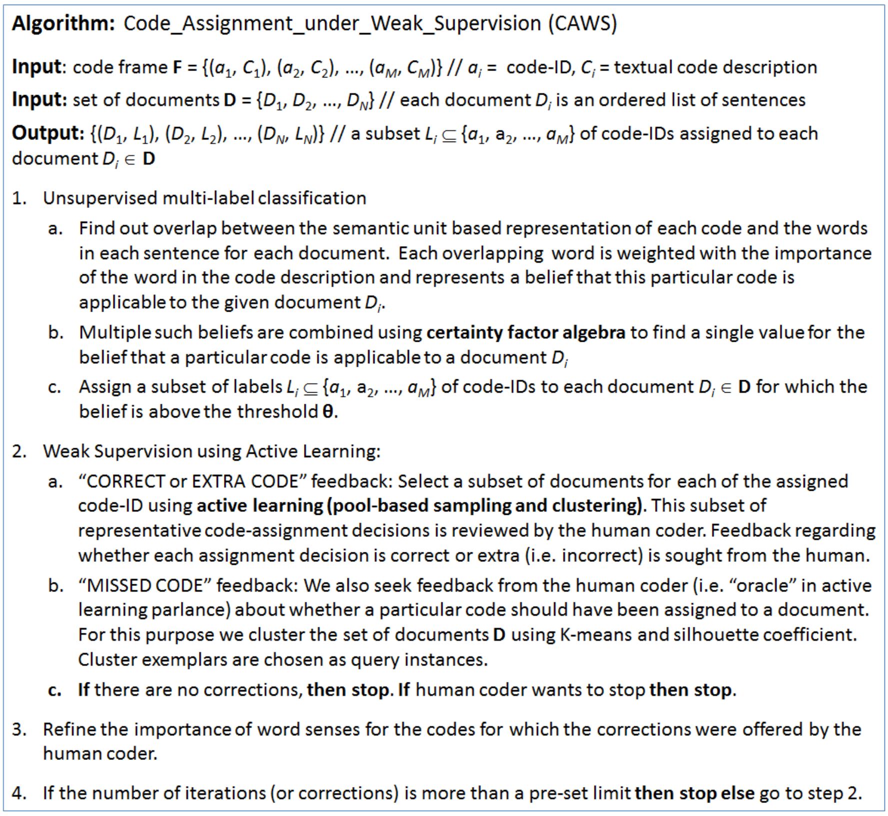
Xu et al. [19] use weighted ridge regression for automatic coding in medical domain and show that it outperforms conventional ridge regression as well as linear SVM. Essentially, their approach assumes that sufficiently large amount of labeled dataset, i.e., training dataset is available. However, in real-life, especially in non-medical domain (for instance, the market research industry), often such training data is either not be available or generating the training data is an expensive, time-consuming proposition.

2.2 Limitations of Existing Approaches

Almost all the supervised learning techniques in research papers and commercial products need training data which is specific to each survey. This training data is not available with the survey and has to be created by the human annotators to begin with. In most of the cases, the cost and effort required to create necessary training data outweighs the benefits of using supervised learning techniques. Thus use of supervised learning techniques is not the best possible solution and it has been found to be a non-viable option in practice.

One may attempt to apply unsupervised techniques such as text clustering to the problem of survey coding. In text clustering, a set of given documents are grouped into one or more clusters, such that documents within a cluster are very similar and documents belonging to different clusters are quite dissimilar. The task of clustering is critically dependent on the notion of text similarity used. It may appear that all documents that have high similarity to a particular code description belong to the same cluster, i.e., each code description defines a cluster of documents. However, in survey coding, more than one code-ID may be assigned to a document; in (non-fuzzy) clustering, a document belongs to one and only one cluster. Even in the case of fuzzy clustering, the clusters formed may not correspond to the pre-specified codes in the code-frame. A much more serious problem is that there is no obvious and fool-proof way to compute the similarity of a document with a given code description. This is because code descriptions as well as the survey responses are typically very short. Also, the similarity of a document with a code is often quite indirect and requires background knowledge.

As a result, current standard practice is to do survey coding using specially trained work-force of human coders and use limited, but viable automation such as regular expression based pattern matching.



```
Algorithm: Code_Assignment_under_Weak_Supervision (CAWS)
Input: code frame F = {($a_1$, $C_1$), ($a_2$, $C_2$), ..., ($a_M$, $C_M$)} // $a_i$ = code-ID, $C_i$ = textual code description
Input: set of documents D = {$D_1$, $D_2$, ..., $D_N$} // each document $D_i$ is an ordered list of sentences
Output: {($D_1$, $L_1$), ($D_2$, $L_2$), ..., ($D_N$, $L_N$)} // a subset $L_i \subseteq$ {$a_1$, a$_2$, ..., $a_M$} of code-IDs assigned to each document $D_i \in$ D

1. Unsupervised multi-label classification
    a. Find out overlap between the semantic unit based representation of each code and the words in each sentence for each document. Each overlapping word is weighted with the importance of the word in the code description and represents a belief that this particular code is applicable to the given document $D_i$.
    b. Multiple such beliefs are combined using certainty factor algebra to find a single value for the belief that a particular code is applicable to a document $D_i$
    c. Assign a subset of labels $L_i \subseteq$ {$a_1$, a$_2$, ..., $a_M$} of code-IDs to each document $D_i \in$ D for which the belief is above the threshold θ.
2. Weak Supervision using Active Learning:
    a. "CORRECT or EXTRA CODE" feedback: Select a subset of documents for each of the assigned code-ID using active learning (pool-based sampling and clustering). This subset of representative code-assignment decisions is reviewed by the human coder. Feedback regarding whether each assignment decision is correct or extra (i.e. incorrect) is sought from the human.
    b. "MISSED CODE" feedback: We also seek feedback from the human coder (i.e. "oracle" in active learning parlance) about whether a particular code should have been assigned to a document. For this purpose we cluster the set of documents D using K-means and silhouette coefficient. Cluster exemplars are chosen as query instances.
    c. If there are no corrections, then stop. If human coder wants to stop then stop.
3. Refine the importance of word senses for the codes for which the corrections were offered by the human coder.
4. If the number of iterations (or corrections) is more than a pre-set limit then stop else go to step 2.
```

Fig. 2. Code Assignment under Weak Supervision (CAWS) algorithm

3 Our Approach

We now present our two stage iterative method in which first we use a new unsupervised multi-label text categorization algorithm. The output of this stage is then passed through a weakly supervised learning stage that uses active learning paradigm. Details of the solution including the individual components, algorithms are given below. A user needs to provide a code-frame F and the set of documents D, i.e., survey responses to which appropriate codes from F are to be assigned.

Feature Extraction: For the first stage of unsupervised multi-label classification, we use a new feature representation called semantic units (SemUnit) for each code. SemUnit tries to capture the concept expressed in the code description. It represents a word in terms of its semantics using its WordNet [4] synset ids and also attaches a weight to measure the relative importance of this word in that code's description. We use the unsupervised word sense disambiguation utilities [9] to estimate the likely word senses. For a given word, this enables us to find out synonyms, antonyms as well as other related words (hypernyms, hyponyms, holonyms, meronyms among many others). This concept-based representation is vital for the next phase of the algorithm. As an illustrative example of SemUnit, consider two sample codes in the sample code-frame shown in Table 1:

- code 04: Verbal ability questions
- code 27: Liked the puzzles

At the end of feature extraction phase, these codes are represented as:

- code 04: Verbal#j#4#i2 ability#n#1#i2 questions#n#1,3#i3
- code 27: Liked#v#2#i3 the#stopword#i0 puzzles#n#1#i1

The above representation denotes that out of all possible meanings of the word *"verbal"*, we consider Wordnet sense number 4 with its part-of-speech tag as adjective. Further, we use one of four pre-determined weights (fractional numbers) to capture the *relative importance* of each word in a particular code's description. Semantics associated with these weights is denoted using following labels:

- **i0** = 0.0 : a word with the importance **i0** is not important at all.
- **i1** = 0.64 : a word with the importance **i1** is the most important word for that particular code and will cause the code to be assigned to a survey response containing this word in the first round of code assignment. (Note that in subsequent rounds of assignment, this weight may get modified.)
- **i2** = 0.4 : a word with the importance **i2** is not sufficient alone to cause the code assignment, but it must be combined with another word from code description which has importance of **i2** or higher.
- **i3** = 0.3 : a word with the importance **i3** is not sufficient alone to cause the code assignment, but it must be combined with at least two other words from code description which have importance of **i3** or higher.

Code-Assignment Stage: In the code-assignment stage, we propose **Code Assignment under Weak Supervision** (CAWS) algorithm (Figure 2) for multi-label text classification. We make use of the semantic unit based representation of each code to find out overlap between that code's textual description and the words in each sentence for each document. We group this lexical overlap along five major word categories, viz., nouns, verbs, adjectives, adverbs and cardinals. Each overlapping word is weighted with the importance of the word in the code description and quantifies our partial belief about whether the corresponding code can be assigned to the given document D_i.

To decide whether a code is applicable to a document, we need to combine the evidence presented by multiple such overlapping words. For this purpose, we use the certainty factor algebra (CFA) [4] to get a single, consolidated value for this belief.

CFA [1] is a simple mechanism to encode a measure of belief (or disbelief) for a hypothesis, rule or event given relevant evidence as well as combining multiple such partial beliefs into a single belief regarding the conclusion.

If the final belief regarding the conclusion is above certain threshold (denoted by θ), we assign the code to the document. Based on the given values for the importance factors (**i0, i1, i2, i3**) as described in previous subsection, the value of this threshold θ is chosen to be 0.6. One can easily note that the specific values of **i1, i2, i3** and θ do not matter much. The threshold value θ is actually a function of **i1, i2, i3**. Any choice of values which preserve the CFA semantics associated with **i1, i2, i3** would work for us.

Active Learning based Weak Supervision: We exploit Quality checking (QC) step in survey coding process to improve the baseline classification done by the unsupervised multi-label classifier described in Section 3. Quality checking (QC) step is a necessary and well established part of the industry standard process to minimize the problem of inter-coder disagreement. QC step essentially consists of verification of the code-assignments by another human coder.

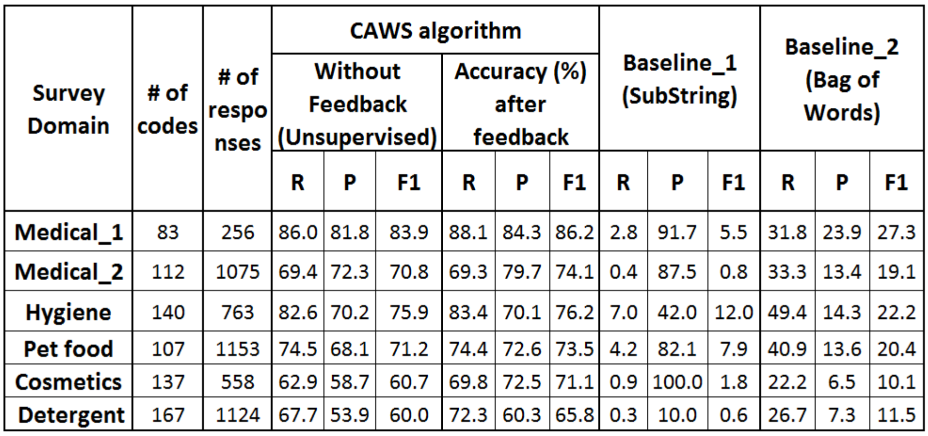
We use active learning [16] techniques to optimize feedback solicitation. We query a human coder, i.e., *"oracle"* in active learning parlance, regarding whether a subset of code-assignments to survey responses are correct. In particular, we use cluster based active learning [10]. For every code a_i in the code-frame, let S_i be the set of responses to which code a_i has been assigned. We cluster S_i using K-means algorithm and query a representative code-assignment instance for each cluster. We use silhouette coefficient [17, 6, 13] to decide number of clusters at run-time. Silhouette Coefficient (ShC) provides a quantified measure of the trade-off between intra-cluster cohesiveness and inter-cluster separation. Silhouette coefficient for i^{th} data point is given by $ShC_i = \frac{b_i - a_i}{max(a_i, b_i)}$, where a_i is the average distance between i^{th} data point and other points in the same cluster; and b_i is average distance between i^{th} data point and all other points in the next nearest cluster. Silhouette Coefficient for a given clustering of data-points is average of individual ShC_i values. We try out different clusterings and pick the one with maximum silhouette coefficient. Medoids of individual clusters (and potentially a few more data-points within each cluster which have maximum distance from the given medoid) are selected as exemplars for which feedback is sought using active learning.

For the query instance, the *oracle*, i.e., human can give feedback regarding whether the code-assignment was correct or extra, i.e., incorrect. If the feedback is correct, our belief regarding the word-senses and their importance is reinforced. If the code-assignment is incorrect, we seek corrective feedback from the *oracle* to know the correct word senses/meaning as well as the relative importance of

[4] A brief summary of CFA is also available at `http://www.cs.fsu.edu/~lacher/courses/CAP5605/documents/scfa.html`

words within the code-description. The *oracle* can also give feedback to identify “missed” code-assignments, i.e., code(s) which should have been assigned, but the multi-label classifier missed it. We update the knowledge base with this feedback so that it can be used to improve the baseline code-assignment in the multi-label classification step as well as future survey coding of surveys of similar category.

If there is corrective feedback provided by the *oracle*, the multi-label classification step is repeated with the additional input available from the feedback step. Thus the code assignments are further refined based on the input available from the QC step. The final set of codes assigned to each document, i.e., survey response are output after the validation as per the quality checking step.



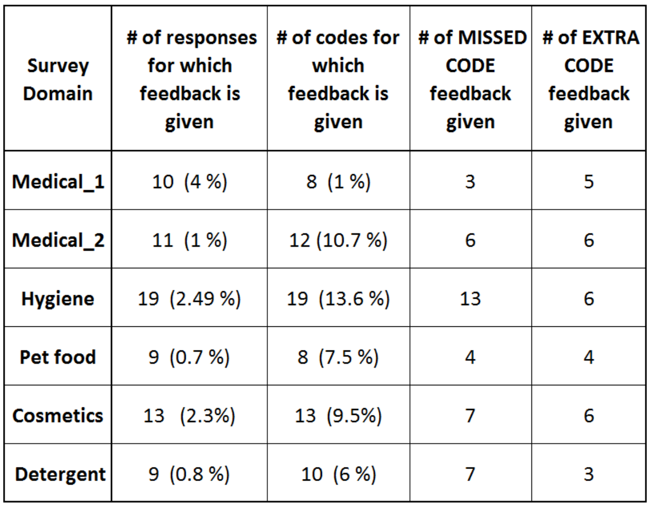
Survey Domain	# of codes	# of responses	CAWS algorithm						Baseline_1 (SubString)			Baseline_2 (Bag of Words)		
			Without Feedback (Unsupervised)			Accuracy (%) after feedback								
			R	P	F1	R	P	F1	R	P	F1	R	P	F1
Medical_1	83	256	86.0	81.8	83.9	88.1	84.3	86.2	2.8	91.7	5.5	31.8	23.9	27.3
Medical_2	112	1075	69.4	72.3	70.8	69.3	79.7	74.1	0.4	87.5	0.8	33.3	13.4	19.1
Hygiene	140	763	82.6	70.2	75.9	83.4	70.1	76.2	7.0	42.0	12.0	49.4	14.3	22.2
Pet food	107	1153	74.5	68.1	71.2	74.4	72.6	73.5	4.2	82.1	7.9	40.9	13.6	20.4
Cosmetics	137	558	62.9	58.7	60.7	69.8	72.5	71.1	0.9	100.0	1.8	22.2	6.5	10.1
Detergent	167	1124	67.7	53.9	60.0	72.3	60.3	65.8	0.3	10.0	0.6	26.7	7.3	11.5

Fig. 3. Sample results for surveys in diverse domains. The accuracy (in %) is reported using the standard measures of Precision (P), Recall (R) and F1.

4 Experimental Results

We have evaluated our approach using a benchmark of multiple survey datasets from diverse domains such as health/medical, household consumer goods (e.g. detergents, fabric softners, etc.), food and snack items, customer satisfaction surveys, campus recruitment test surveys etc. Each dataset was annotated by a human expert. A sample set of responses from each dataset was independently verified by another domain expert. We have chosen datasets for which the sample annotation verification by experts had average agreement of 95%.

We did not come across any public-domain tools for automated survey coding against which we could compare our approach. To show the effectiveness of our method and to highlight the difficulty of survey coding task, we compare with two baseline approaches. In the first baseline approach (Baseline_1), we assign a code to a response if the code description appears as a substring of that response text. In the second baseline approach (Baseline_2), we relax the stringent requirement of exact substring match and use the bag of words (BoW) approach. We compute the word overlap between a code description and a response, after removing the stop words from both. Note that the code-frames for these surveys are organized in a hierarchy of two levels. In Baseline_2, for each parent-level category in a code-frame, we score each code with the fraction of its words overlapping with the response. Within each parent-level category, we assign the code with maximum, non-zero overlap with the response.



Survey Domain	# of responses for which feedback is given	# of codes for which feedback is given	# of MISSED CODE feedback given	# of EXTRA CODE feedback given
Medical_1	10 (4 %)	8 (1 %)	3	5
Medical_2	11 (1 %)	12 (10.7 %)	6	6
Hygiene	19 (2.49 %)	19 (13.6 %)	13	6
Pet food	9 (0.7 %)	8 (7.5 %)	4	4
Cosmetics	13 (2.3%)	13 (9.5%)	7	6
Detergent	9 (0.8 %)	10 (6 %)	7	3

Fig. 4. Details of feedback given for the exemplars selected using active learning for the output shown in Figure 3

Figure 3 summarizes some of our results of unsupervised classification of survey responses (without using any feedback) as well as the improvement in the accuracy after feedback. In Figure 3, we see that Baseline_1 has excellent average precision; however, it performs very poorly in the recall. Baseline_2 does not demand exact match of code description with response. It looks for non-contiguous overlap between code description and response text. Expectedly, Baseline_2 improves the recall. However, it suffers in the precision due to inherent limitation of the bag of words approach which ignores the associated semantics. We contrast this with the high accuracy achieved by our approach even without any feedback and underscore its effectiveness.

Figure 4 shows that the amount of feedback required to achieve improvement in accuracy is quite less compared to the total number of responses and codes. This indicates that active learning is effective for minimizing the feedback required to improve the accuracy.

5 Conclusion

Survey coding application has non-trivial challenges and does not lend itself easily to automation. In this paper, we suggested that standard machine learning approaches for text classification or clustering are not viable in practice for survey coding task. We presented a two step, iterative algorithm - **Code Assignment under Weak Supervision** (CAWS). In the first step, multi-label categorization is achieved in an unsupervised manner aided by a knowledge base. Then, we exploit the quality checking(QC) phase, which is an important part of survey coding process, to improve the accuracy further. We use active learning technique to optimize the weak supervision available in the form of human feedback in QC. We observed that our approach achieves good accuracy on human annotated benchmark data and works well for surveys from diverse domains.

References

1. Buchanan, B., Shortliffe, E.: Rule Based Expert Systems: The MYCIN Experiments of the Stanford Heuristic Programming Project. Addison-Wesley, Reading, MA (1984), iSBN 978-0-201-10172-0
2. Esuli, A., Sebastiani, F.: Active learning strategies for multi-label text classification. In: Proceedings of ECIR. LNCS, vol. 5478, pp. 102–113. Springer (2009)
3. Esuli, A., Sebastiani, F.: Machines that learn how to code open-ended survey data. International Journal of Market Research 52(6) (2010), dOI: 10.2501/S147078531020165X
4. Fellbaum, C.: WordNet: An Electronic Lexical Database. MIT Press (1998 (ed))
5. Giorgetti, D., Prodanof, I., Sebastiani, F.: Automatic coding of open-ended surveys using text categorization techniques. In: Proceedings of Fourth International Conference of the Association for Survey Computing. pp. 173–184 (2003)
6. Kaufman, L., Rousseeuw, P.J.: Finding groups in data: An introduction to cluster analysis. Wiley series in Probability and Statistics. John Wiley and Sons, New York (1990)
7. Li, H., Yamanishi, K.: Mining from open answers in questionnaire data. In: Proceedings of Seventh ACM SIGKDD (2001)
8. Macer, T., Pearson, M., Sebastiani, F.: Cracking the code: What customers say in their own words. In: Proceedings of MRS Golden Jubilee Conference (2007)
9. Navigli, R.: Word sense disambiguation: A survey. ACM Computing Surveys (CSUR) 41(2), 10 (2009)
10. Nguyen, H., Smeulders, A.: Active learning using pre-clustering. In: Proceedings of the International Conference on Machine Learning (ICML). pp. 79–86. ACM (2004)
11. Patil, S., Palshikar, G.K.: Surveycoder: A system for classification of survey responses. In: Proceedings of the 18th International Conference on Application of Natural Language to Information Systems (NLDB 2013), LNCS 7934. Springer-Verlag (2013)

12. Pratt, D., Mays, J.: Automatic coding of transcript data for a survey of recent college graduates. In: Proceedings of the Section on Survey Methods of the American Statistical Association Annual Meeting. pp. 796–801 (1989)
13. Rousseeuw, P.J.: Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. Journal of Computational and Applied Mathematics 20, 53–65 (1987)
14. Schuman, H., Presser, S.: The open and closed question. American Sociological Review 44(5), 692–712 (1979)
15. Sebastiani, F.: Machine learning in automated text categorization. ACM Computing Surveys 34(1), 1–47 (2002)
16. Settles, B.: Active Learning. Morgan Claypool, Synthesis Lectures on AI and ML (2012)
17. Tan, P.N., Steinbach, M., Kumar, V.: Introduction to Data Mining. Addison-Wesley, Upper Saddle River, NJ (2005)
18. Viechnicki, P.: A performance evaluation of automatic survey classifiers. In: Proceedings of Fourth International Colloquium on Grammatical Inference (ICGI). LNCS, vol. 1433, pp. 244–256. Springer Verlag (1998)
19. Xu, J.W., Yu, S., Bi, J., Lita, L.V., Niculescu, R.S., Rao, R.B.: Automatic medical coding of patient records via weighted ridge regression. In: Proceedings of Sixth International Conference on Machine Learning and Applications (ICMLA) (2007)