# A Multi-Label Classification Approach for Coding Cancer Information Service Chat Transcripts

**Anthony Rios**
Dept. of Computer Science
University of Kentucky
anthony.rios1@uky.edu

**Robin Vanderpool**
Dept. of Health Behavior
University of Kentucky
robin@kcr.uky.edu

**Pam Shaw**
Kentucky Cancer Registry
University of Kentucky
pshaw@kcr.uky.edu

**Ramakanth Kavuluru***
Div. of Biomedical Informatics
Dept. of Computer Science
University of Kentucky
ramakanth.kavuluru@uky.edu

**Abstract**

National Cancer Institute's (NCI) Cancer Information Service (CIS) offers online instant messaging based information service called LiveHelp to patients, family members, friends, and other cancer information consumers. A cancer information specialist (IS) 'chats' with a consumer and provides information on a variety of topics including clinical trials. After a LiveHelp chat session is finished, the IS codes about 20 different elements of metadata about the session in electronic contact record forms (ECRF), which are to be later used for quality control and reporting. Besides straightforward elements like age and gender, more specific elements to be coded include the purpose of contact, the subjects of interaction, and the different responses provided to the consumer, the latter two often taking on multiple values. As such, ECRF coding is a time consuming task and automating this process could help ISs to focus more on their primary goal of helping consumers with valuable cancer related information. As a first attempt in this task, we explored multi-label and multi-class text classification approaches to code the purpose, subjects of interaction, and the responses provided based on the chat transcripts. With a sample dataset of about 673 transcripts, we achieved example-based F-scores of 0.67 (for subjects) and 0.58 (responses). We also achieved label-based micro F-scores of 0.65 (for subjects), 0.62 (for responses), and 0.61 (for purpose). To our knowledge this is the first attempt in automatic coding of LiveHelp transcripts and our initial results on the smaller corpus indicate promising future directions in this task.

## 1 Introduction

The Cancer Information Service (CIS) program at the National Cancer Institute (NCI) provides assistance to cancer patients and their friends and family through an online instant messaging chat program called LiveHelp. Through these chat sessions, CIS cancer information specialists (ISs) answer specific questions about cancer, clinical trials, and quitting smoking. They also provide information about dealing with cancer such as support groups and coping methods. However, the ISs do not give medical advice, but only point to relevant clinical trials and cancer information that can help patients and their family members make informed decisions. LiveHelp is available 75 hours a week where individual ISs provide specific information tailored to particular user situations. To give an estimate, according to the internal documents of CIS there were 13485 chat sessions in 2010.

After each chat session an IS records about 20 elements of metadata associated with the concluded chat session in the form of an electronic contact record form (ECRF), a task which is expected to take up to 10 minutes assuming some simple elements are filled during the chat. Three of the important elements they are expected to capture include 'response', 'subject of interaction', and 'purpose of contact'. The subject of interaction captures the different topics that were discussed during the chat. The response captures different classes of responses given by the IS during the chat. The purpose of contact captures the original reason the patient or family member initiated the chat. The response and subject of interaction elements can take on multiple values. That is, there can be multiple subjects of interaction and responses. However, the purpose of contact can take only one value. There are several other metadata elements, some of which take on multiple values that need to be coded for each chat session. CIS also has an official manual for the ISs that details how the coding should be done. Hence, coding ECRF forms is a very time consuming process, which although helps with reporting, staff training, and quality control, does not directly contribute to the original purpose of LiveHelp – to empower cancer patients and family (henceforth referred to as information consumers) with the necessary information. Having computational methods that can automate the coding process or expedite it by providing candidate elements can help free up the ISs' time that can be used to help more consumers. The main purpose of this paper to study how text classification approaches can be used to automate the coding process. The initial results in this paper pertain to the response, subject of interaction, and purpose elements.

We experimented with multi-label text classification approaches to extract the codes for response and subject as each of these can take on multiple values. We experimented with multi-class classification approaches for the purpose element. The text used was the entire transcript of each chat session. To the best of our knowledge, this is the first attempt in automatic coding of LiveHelp transcripts. Our results on

*Corresponding Author

a smaller dataset indicate potential of automated methods in assisting ISs in expediting the process of ECRF coding in the future.

In Section 2, we introduce general methods used in multi-label classification and analysis of chat transcripts. We describe our dataset in Section 3 and the methods we used in Section 4. After discussing evaluation measures in Section 5, we present our results in Section 6 and conduct a qualitative error analysis in Section 7.

## 2 Background

There has been a lot of recent work in multi-label learning, the general problem of choosing multiple labels among a set of possible labels for each object that needs to be classified. A class of approaches called problem transformation approaches convert the multi-label problem into multiple single-label classification instances. A second class of methods adapts the specific algorithms for single-label classification problems to directly predict multiple labels. Both problem transformation and algorithm adaptation techniques are covered in this recent survey by de Carvalho and Freitas (2009). Recent attempts in multi-label classification also consider label correlations (Read et al. 2011; Zhang and Zhang 2010) when building a model for multi-label data. An important challenge in problems with a large number of labels per document is to decide the number of candidates after which candidate labels should be ignored, which has been recently addressed by calibrated ranking (Fürnkranz et al. 2008).

Chat transcript analysis has also been subject of interest to computer scientists in recent years especially with the advent of online social networks. Gianvecchio et al. (2011) developed classification models to distinguish human users vs automated bots in chat rooms. Sentiment analysis is being used to learn about international terrorism by analyzing the 'dark side of the web' (Chen 2011). Much of the current research with chat transcripts involves chat analysis considering each individual message. Dascalu, Rebedea, and Trausan-Matu (2010) propose an end-to-end information system with methods to assess utterances and participants, visualize and summarize topics discussed and enable semantic search in the context of analyzing chats between tutors and students.

Our current effort differs from much of the current research using chat transcripts in that we don't analyze each message for sentiment but use the entire chat transcript for ECRF code extraction. Goes et al. (2012) tried to prioritize customers for assignment to available chat agents on e-commerce websites to make more efficient use of their employees' time. While their dataset is similar to what we have, the fundamental problem we are working on is different. The goal of our research is to use current methods in multi-label and multi-class learning to attempt to build a system for automatic coding of NCI LiveHelp chat transcripts.

## 3 LiveHelp Dataset

Our dataset has a total of 673 LiveHelp chat transcripts (from 2010) obtained from CIS. 50 of these were removed to be used for testing purposes. For this paper, we decided to extract three important elements : subject of interaction, response, and purpose of contact. We trained three different sets of classifiers, one for each of these elements, with the corresponding pre-processing tailored to each of the elements.

Subject had a total of 50 possible labels which could be used to label each transcript. Out of the 50 total possible labels, 31 labels were used at least once in our training set. To have enough training data, we only trained on labels that were used at least 20 times, which left us with ten labels to be used for training the subject classifier. After removing labels with less than 20 examples, we were left with a few transcripts not labeled with any of the ten labels we decided to trian on. We removed these transcripts from our training set which left us with a total of 580 transcripts to use for training for the ten labels.

Response had a total of 38 possible labels of which 31 were used at least once in our training set. Again, we removed labels that did not occur at least 20 times, and then removed transcripts without any labels that we decided to train on. This left us with a total of 575 transcripts to use for training the response classifier and we trained on fourteen labels that had at least 20 examples.

*Label-cardinality* is the average number of labels used per transcript. Let $m$ be the total number of transcripts and $\mathbf{Y}_i$ be the labels for transcript $i$. Then we have

$$\text{Label-Cardinality} = \frac{1}{m}\sum_{i=1}^{m}|\mathbf{Y}_i|.$$

Subject had a label cardinality of $1.4$ and response had a label cardinality of $2.6$. Another useful statistic that describes the dataset is *label-density*, which divides the average number of labels per transcript by the total number of labels $q$. We have

$$\text{Label-Density} = \frac{1}{q}\cdot\frac{1}{m}\sum_{i=1}^{m}|\mathbf{Y}_i|.$$

The label-density for subject is $0.15$ and for response is $0.19$. Unlike label-cardinality, it also takes into account the number of unique labels possible. Two datasets with the same label cardinality can have different label densities and might need different learning approaches tailored to the situations. Intuitively, in this case, the dataset with the smaller density is expected to have fewer training examples per label.

Purpose of contact had a total of thirteen possible labels. There were twelve labels used out of the total thirteen in our training set. After removing labels not used at least 20 times and removing transcripts without any labels that were used at least 20 times, we were left with 607 transcripts and seven labels to use for training. Since purpose of contact can only take one possible label, we do not report label cardinality and density here. All these stats are summarized in Table 1.

A label count summary of the dataset is shown in Table 2. It shows the number of times each label was used in the training set. The label numbers in the top row of the table do not mean that the subject, response, and purpose elements all contain the same labels. They are merely indicators of

the label ids with more than 20 examples. That is, the actual label $\#i$ in the table for each $i$ is different for response, subject, and purpose.

## 4 Text Classification Approaches

We used the Java based packages Weka (Hall et al. 2009) and Mulan (Tsoumakas et al. 2011) for our experiments. Subject and response were trained using the same feature types and learning algorithms. We used unigrams and bigrams (just referred to as ngrams henceforth) with the term frequency and inverse document frequency (TF-IDF) transformation applied to raw frequencies. Ngrams containing a token which is a single letter or starts with a non-alphabetic character are removed from feature list. Stop words (determiners, prepositions, and so on) are removed from the unigram features. The ngram features were used only if they occurred in at least five documents. This left us with a total of 11646 features for subject and 12551 features for response. For purpose we used the same features and removed the features in the same way, except we also removed bigrams that contained a stop word. We had a total of 3813 features for purpose. We experimented with three different multi-label problem transformation methods for both subject and response (Tsoumakas et al. 2011) with three different base classifiers. We used *binary relevance, copy transformation,* and *ensemble of classifier chains* for transformation and used naive Bayes (NB), support vector machines (SVMs), and logistic regression (LR) for base classifiers. Thus we used a total of nine different combinations.

The copy transformation transforms multi-label data into single-label data. Let $T = \{T_1, \ldots, T_q\}$ be the set of $q$ possible labels for a given multi-label problem. Let each document $D_j \in D$, $j = 1, \ldots, m$, has a set of labels $\mathbf{Y}_j \subseteq T$ associated with it. The copy transformation transforms each document–label-set pair $(D_j, \mathbf{Y}_j)$ into $|\mathbf{Y}_j|$ document–label pairs $(D_j, T_s)$, for all $T_s \in \mathbf{Y}_j$. After the transformation, each input pair for the classification algorithm will only have one label associated with it and one can use any single-label method for classification. In our method we used one-vs-all SVM and LR, and multinomial NB. The labels are then ranked based on the score given from the classifier when generating predictions. We then take the top $k$ labels as our predicted label set.

Binary relevance learns $q$ binary classifiers, one for each label in $T$. It will transform the dataset into $q$ separate datasets. For each label $T_j$, we obtain the dataset for the corresponding binary classifier by considering each document–label-set pair $(D_i, \mathbf{Y}_i)$ and generating the document-label pair $(D_i, T_j)$ when $T_j \in \mathbf{Y}_i$ and generating the pair $(D_i, \neg T_j)$ when $T_j \notin \mathbf{Y}_i$. Binary relevance is often used as a baseline method for performance evaluation of other advanced methods. When predicting, the labels are ranked based on their score output by the corresponding binary classifiers and the top $k$ labels are considered as the predicted set for a suitable $k$.

One of the main disadvantages of binary relevance and copy transformation methods is that they assume label independence. In practical situations, there can be dependence between labels where labels co-occur very frequently or where a label occurs only when a different label is also tagged. Classifier chains, based on the binary relevance transformation, try to account for these dependencies that the base transformations cannot. Like binary relevance, classifier chains transform the dataset into $q$ datasets for binary classification per each label. But they differ from binary relevance in the training phase. Binary relevance trains each of the $q$ classifiers independently, but classifier chains loop through each dataset in some order, training each classifier one at a time. After a classifier is trained, it makes a prediction for each document in the training phase of all the subsequent Boolean classifiers. Thus, it will add a new Boolean feature to the subsequent binary classifier datasets to be trained next. The feature value will be a 0 or 1 depending on whether the label was predicted by the already trained corresponding classifier. So, the final $q$-th classifier to be trained will have $q-1$ additional Boolean features whose values are based on predictions of the first $q-1$ binary classifiers. The prediction on unseen datasets is performed exactly the same way as in the binary relevance approach.

Classifier chain performance heavily depends on the order of chaining of individual classifiers. To account for this, we used the ensemble of classifier chains approach. Here, we use $n$ different classifier-chain classifiers, each of which is trained using a random ordering of the labels in $T$ and uses a random subset of the original dataset. Predictions are made by keeping track of the number of times each label $T_i$ is predicted (among the $n$ chain classifiers) and a threshold is used to obtain a subset of labels for the input document. Thus the $n$ different chains vote for each of the labels.

## 5 Evaluation Measures

We evaluate our methods using both label-based and example-based F-scores (Tsoumakas, Katakis, and Vlahavas 2010). Before we present the results, we briefly discuss these measures.

For each label $T_j$ in the set of labels $T$ being considered, we have label-based precision $P(T_j)$, recall $R(T_j)$, and F-score $F(T_j)$ defined as

$$P(T_j) = \frac{TP_j}{TP_j + FP_j},\ R(T_j) = \frac{TP_j}{TP_j + FN_j},$$
$$\text{and } F(T_j) = \frac{2P(T_j)R(T_j)}{P(T_j) + R(T_j)},$$

where $TP_j$, $FP_j$, and $FN_j$ are true positives, false positives, and false negatives, respectively, of label $T_j$. Given this, the label-based macro average F-score is

$$\text{Macro-F} = \frac{1}{|T|}\sum_{j=1}^{|T|} F(T_j).$$

The label-based micro precision, recall, and F-score are defined as

$$P^{mic} = \frac{\sum_{j=1}^{|T|} TP_j}{\sum_{j=1}^{|T|}(TP_j + FP_j)}, R^{mic} = \frac{\sum_{j=1}^{|T|} TP_j}{\sum_{j=1}^{|T|}(TP_j + FN_j)},$$
$$\text{and Micro-F} = \frac{2P^{mic} \cdot R^{mic}}{P^{mic} + R^{mic}},$$

| | Possible Labels | Labels used $\geq 1$ times | Labels used $\geq 20$ times | Label-Cardinality | Label-Density |
|---|---|---|---|---|---|
| Subject | 50 | 31 | 10 | 1.4 | 0.15 |
| Response | 38 | 31 | 14 | 2.6 | 0.19 |
| Purpose | 13 | 12 | 7 | NA | NA |

Table 1: LiveHelp dataset statistics

| | #1 | #2 | #3 | #4 | #5 | #6 | #7 | #8 | #9 | #10 | #11 | #12 | #13 | #14 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Response | 363 | 32 | 37 | 174 | 23 | 96 | 417 | 36 | 123 | 57 | 60 | 28 | 121 | 20 |
| Subject | 33 | 216 | 56 | 45 | 36 | 103 | 30 | 40 | 40 | 243 | | | | |
| Purpose | 151 | 71 | 265 | 40 | 29 | 30 | 21 | | | | | | | |

Table 2: Number of chats for each label with at least 20 examples for Response, Subject, and Purpose.

respectively. While the macro measures consider all labels as equally important, micro measures tend to give more importance to labels that are more frequent. This is relevant for our dataset because we have a very unbalanced set of label counts (see Table 2) and in such cases micro measures are considered more important.

Recall that $\mathbf{Y}_i$, $i = 1, \ldots, m$, is the set of correct labels in the dataset for the $i$-th transcript, where $m$ is the total number of transcripts. Let $\mathbf{Z}_i$ be the set of predicted labels for the $i$-th transcript. The example-based precision, recall, and F-score are defined as

$$P_{ex} = \frac{1}{m}\sum_{i=1}^{m}\frac{|\mathbf{Y}_i \cap \mathbf{Z}_i|}{|\mathbf{Z}_i|},\ R_{ex} = \frac{1}{m}\sum_{i=1}^{m}\frac{|\mathbf{Y}_i \cap \mathbf{Z}_i|}{|\mathbf{Y}_i|},$$

$$\text{and } F_{ex} = \frac{1}{m}\sum_{i=1}^{m}\frac{2|\mathbf{Y}_i \cap \mathbf{Z}_i|}{|\mathbf{Z}_i| + |\mathbf{Y}_i|}, \text{ respectively.}$$

## 6 Results

As indicated in Section 3, 50 transcripts from the original dataset were removed for testing purposes. To strengthen our evaluation, we had two former CIS ISs separately code these 50 transcripts for subject, response, and purpose. Each IS separately coded a maximum of 5 subjects, 4 responses, and one purpose, following the CIS guidelines. Later, the two ISes met and resolved disagreements and came up with a finalized list of expert predictions for the 50 held out transcripts. These are the ground truth predictions used for testing purposes.

We compute measures for both testing and training datasets in this section. First, we present the example-based measures for subject of interaction in Table 3 and response in Table 4. These results were obtained using the ensemble of classifier chains approach (with five classifier chains in the ensemble) and SVMs as the base classification method in each chain, which we found to be the best of the nine combinations of transformation methods and base classifiers we used (see Section 4). The training scores were calculated by using 5-fold cross-validation. We note that the performance measures reported here only take into account the labels we have trained on, that is, those that had at least 20 example transcripts in the dataset. This is because we won't be able to predict labels that we did not train on, and hence errors due to labels besides those we trained on should not contribute to the performance scores we report here.

The column $k$ in Tables 3 and 4 corresponds to the number of labels we predicted and used for computing the evaluation measures for multi-label elements. For example, $k = 1$ means we predict exactly one label for each transcript. By examining the tables, not surprisingly, we note that the best scores occur when we use a $k$ that is closest to the label-cardinality of the training data because we don't predict too few or too many labels. As can be noticed, choosing a larger $k$ increases the recall but the decrease in precision is almost twice the increase in recall in Table 3. For subject, setting $k = 1$ gives the best F-score and using $k = 2$ or 3 gives the best F-score for response. Interestingly, the performance does not decrease on the testing test. On the contrary, it increases for subject and stays the same for response.

| | Subject of interaction | | | | | |
|---|---|---|---|---|---|---|
| | Training | | | Testing | | |
| $k$ | Prec | Recall | F-score | Prec | Recall | F-score |
| 1 | 0.74 | 0.58 | 0.63 | 0.76 | 0.62 | 0.67 |
| 2 | 0.46 | 0.69 | 0.53 | 0.48 | 0.77 | 0.62 |
| 3 | 0.33 | 0.73 | 0.43 | 0.35 | 0.83 | 0.50 |
| 4 | 0.26 | 0.77 | 0.38 | 0.27 | 0.86 | 0.41 |
| 5 | 0.21 | 0.78 | 0.33 | 0.22 | 0.88 | 0.36 |

Table 3: Example-based scores for Subject.

| | IS Response to Consumers | | | | | |
|---|---|---|---|---|---|---|
| | Training | | | Testing | | |
| $k$ | Prec | Recall | F-score | Prec | Recall | F-score |
| 1 | 0.75 | 0.32 | 0.42 | 0.72 | 0.39 | 0.43 |
| 2 | 0.69 | 0.57 | 0.58 | 0.65 | 0.62 | 0.57 |
| 3 | 0.56 | 0.66 | 0.57 | 0.55 | 0.75 | 0.58 |
| 4 | 0.47 | 0.71 | 0.53 | 0.45 | 0.79 | 0.53 |

Table 4: Example-based scores for Response.

Table 5 shows the macro and micro averaged precision, recall, and F-Score for all three elements extracted. Here we only used the best $k$ for subject and response, that is $k = 1$ for subject, and $k = 3$ for response (as it was closer to the label-cardinality 2.6). Since purpose is a multi-class problem we only had to predict one label. Again, we notice the testing scores are almost always higher than the training scores.

| | Training | | | | | | Testing | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Micro | | | Macro | | | Micro | | | Macro | | |
| | Prec | Recall | F-score | Prec | Recall | F-score | Prec | Recall | F-score | Prec | Recall | F-score |
| Subject | 0.74 | 0.51 | 0.61 | 0.61 | 0.39 | 0.45 | 0.76 | 0.57 | 0.65 | 0.61 | 0.40 | 0.43 |
| Response | 0.56 | 0.61 | 0.58 | 0.47 | 0.38 | 0.37 | 0.55 | 0.71 | 0.62 | 0.53 | 0.39 | 0.42 |
| Purpose | 0.60 | 0.60 | 0.60 | 0.61 | 0.49 | 0.51 | 0.60 | 0.63 | 0.61 | 0.78 | 0.65 | 0.67 |

Table 5: Micro/Macro scores for Subject, Response, and Purpose.

For completeness, we show the micro, macro, and example-based training F-scores in Table 6 for copy and binary relevance (indicated in the table as "Bin Rel") transformations using SVM as the base classifier with $k = 1$ for subject and $k = 3$ for response. We note that almost always the classifier chain ensemble method outperforms other methods although the scores are closer in some cases. Among the two, we see the copy transformation performs better than the binary relevance approach in general.

| | Subject | | Response | |
|---|---|---|---|---|
| | Copy | Bin Rel | Copy | Bin Rel |
| Micro | 0.58 | 0.53 | 0.51 | 0.32 |
| Macro | 0.46 | 0.36 | 0.31 | 0.16 |
| Example-Based | 0.58 | 0.53 | 0.48 | 0.31 |

Table 6: The micro, macro, and example-based training F-scores for subject and response

## 7 Qualitative Error Analysis

We qualitatively analyzed the recall and precision errors for subject, response, and purpose to identify classes of errors in the testing set and the corresponding reasons arising out of our methods or the nature of coding process. We conducted the analysis along with the domain expert ISs who helped create the testing dataset codes.

The first class of errors were generated due to the inherent dependency between codes and their purposes for certain code pairs. The ISs thought that for these code pairs, either or both codes could be coded. An example for subject is the code pair 'General site information' (related to the cancer site or anatomical location) and 'Treatment/side effect management'. Because the label-cardinality of our dataset for subject was only 1.4, we achieved the best F-scores when predicting one label. The ISs indicated that a site information subject appears initially in the chat and usually ensues a discussion on treatment management for the site. Since we were only predicting one label for subject, we were unable to account for this in our results which resulted in recall errors for the treatment management code. Although predicting two labels would increase the recall, it is not a good idea if the goal is to maximize F-score because up to 60% of the second label predictions could be incorrect based on the label-cardinality of 1.4. This can be noted from Table 3 where going from $k = 1$ to 2 increases recall by 11 percentage points but decreases precision by 38 points. However, for recall oriented applications, this trade-off might be acceptable and in this case the ISs could quickly discard an inaccurate code.

The label interdependency issues also caused precision errors. For response, the dataset label cardinality is 2.6 and hence we chose top $k = 3$ labels per transcript, which generated the best testing F-score (see Table 4). However, there were significant dependencies between the response labels 'discuss/share/visit with health professional' and 'American cancer society (ACS)', which is the largest national voluntary health organization that offers prevention, coping, and clinical trial advice to patients and family members. In our training set, 18 of the 22 transcripts coded with 'discuss with health professional' were also coded with the label ACS. However, in our testing data, ACS was never coded by the ISs but was predicted by our methods for 25 of the 50 transcripts. Out of these 25 precision errors, there were 13 cases where the label 'discuss with health professional' was correctly coded. The strong dependency in the training set made it into the classifier when using the classifier chain ensemble approach. We found that these dependencies were also the reason behind several other response precision errors. It appears that our methods found label dependencies that do not generalize well, an issue that can be mitigated by using a larger dataset.

We had a similar situation for the purpose of contact predictions. There were two labels, 'understand medical information' and 'concerned about family/friend with cancer', both of which could be used for purpose in several transcripts. Since the coding guidelines suggest that only one purpose should be coded, even though the ISs thought that both labels applied, they were forced to code only one of the labels and so did our methods. We failed to predict the right purpose for 18 of the 50 transcripts. Of these, 10 were caused because of the inherent similarity between the two purpose labels discussed above.

A different class of recall errors was caused because of the nature of particular labels and how information about them is expressed in chat transcripts. For example, the label 'CIS fact sheet' for response is expected to be coded when the IS refers the consumer to a CIS fact sheet, generally available as a web page on the CIS website. Hence transcripts that were coded with that label generally contained URLs that had "factsheet" as a substring. Although we used '/' as a delimiter for tokenization, the unigram "factsheet" without the space between 'fact' and 'sheet' was infrequent when compared with transcripts that had "fact sheet" (with the space) as part of the conversation. All of the 8 cases where our methods missed 'CIS fact sheet' had the URLs but not the words 'fact' and 'sheet' in them. Another reason for recall errors for 'CIS fact sheet' (and also for 'discuss with

health professional') was because of the limit on the number of possible codes, which was four for response. When the ISs find four suitable codes and decide to code them, they could miss other relevant codes. For example, when the IS is clearly providing fact sheet URLs to the consumer, according to guidelines, they are supposed to code the corresponding label. But if they already coded four other responses, they may forgo coding the 'CIS fact sheet' label.

## 8 Concluding Remarks

CIS's LiveHelp is a helpful online chat service where specialists answer questions and provide valuable information on cancer related topics to assist cancer patients, their friends and family. The ISs are expected to code ECRF forms for each chat which requires a significant amount of time which could otherwise be used to assist other consumers or to conduct other activities. In this paper, we conducted preliminary experiments to expedite the coding process by producing predictions using multi-label classification approaches based on ngram features extracted from chat transcripts. Given our initial encouraging results on our small training corpus of 673 transcripts, we believe there is excellent potential to make significant progress in this task. We plan to improve our work along the following directions.

- Taking into account the nature of the errors in Section 7, we plan to customize our methods. First, we will use regular expressions to capture URLs in transcripts that correspond to specific labels and use them as features instead of splitting them up using delimiters. We will also modify our classifiers to dynamically change the number of labels to be predicted on a case-by-case basis (Fürnkranz et al. 2008) instead of just using a fixed $k$ for all transcripts.
- While our classifier chains chained outputs from labels corresponding to the same element (subject/response), capturing label dependencies across different metadata elements could also give additional performance gains. For example, the label for purpose 'want help saying quit' could indicate a subject label 'smoking/tobacco use' eliciting a IS response coded using the 'smoking/tobacco use cessation' response label. While this example indicates a direct relationship, there could be other subtler latent relationships between purpose, subject, and response labels that could be captured using classifier chains that chain individual label classifiers across different elements. We will investigate this as part of the future work.
- It is also an interesting task to limit training of certain elements to specific portions of the transcripts. For instance, the response codes capture the important responses provided by the ISs to consumers. As such, training only on the IS messages of the chat could lead to more accurate predictions instead of considering the entire transcript, a task we will explore in the immediate future.
- As a proxy for inter-coder agreement, we used the popular set similarity measure, the Jaccard index (Real and Vargas 1996), and averaged it over all testing documents for subject, response, and purpose based on the output code sets by both of our ISs. We obtained similarities of nearly 90% for subject and response, and 80% for purpose. However, we realize that this does not consider the chance agreement. In future work, we plan to use the more formal framework of Cohen's $\kappa$ statistic adapted to the multi-label situation (Rosenberg and Binkowski 2004) for computing inter-coder agreement.
- As indicated in Section 7, using a larger dataset can mitigate some of the issues related to over-fitting label dependencies that manifest in the training data. We plan to obtain more transcripts and the ECRF codes from CIS to conduct experiments on a larger scale.

## Acknowledgements

This publication was supported by the National Center for Research Resources and the National Center for Advancing Translational Sciences, National Institutes of Health, through Grant UL1TR000117. The content is solely the responsibility of the authors and does not necessarily represent the official views of the NIH.

## References

Chen, H. 2011. *Dark Web: Exploring and Data Mining the Dark Side of the Web*. Springer.

Dascalu, M.; Rebedea, T.; and Trausan-Matu, S. 2010. A deep insight in chat analysis: Collaboration, evolution and evaluation, summarization and search. In *Artificial Intelligence: Methodology, Systems, and Applications*, volume 6304 of *LNCS*. 191–200.

de Carvalho, A. C. P. L. F., and Freitas, A. A. 2009. A tutorial on multi-label classification techniques. In *Foundations of Computational Intelligence (5)*, volume 205 of *Studies in Computational Intelligence*. 177–195.

Fürnkranz, J.; Hüllermeier, E.; Loza Mencía, E.; and Brinker, K. 2008. Multilabel classification via calibrated label ranking. *Mach. Learn.* 73(2):133–153.

Gianvecchio, S.; Xie, M.; Wu, Z.; and Wang, H. 2011. Humans and bots in internet chat: measurement, analysis, and automated classification. *IEEE/ACM Trans. Netw.* 19(5):1557–1571.

Goes, P.; Ilk, N.; Yue, W. T.; and Zhao, J. L. 2012. Live-chat agent assignments to heterogeneous e-customers under imperfect classification. *ACM Trans. Manage. Inf. Syst.* 2(4):24:1–24:15.

Hall, M.; Frank, E.; Holmes, G.; Pfahringer, B.; Reutemann, P.; and Witten, I. H. 2009. The weka data mining software: an update. *SIGKDD Explor. Newsl.* 11(1):10–18.

Read, J.; Pfahringer, B.; Holmes, G.; and Frank, E. 2011. Classifier chains for multi-label classification. *Machine Learning* 85(3):335–359.

Real, R., and Vargas, J. 1996. The probabilistic basis of jaccard's index of similarity. *Sys. Biology* 45(3):380–385.

Rosenberg, A., and Binkowski, E. 2004. Augmenting the kappa statistic to determine interannotator reliability for multiply labeled data points. In *Proceedings of HLT-NAACL 2004*, 77–80.

Tsoumakas, G.; Spyromitros-Xioufis, E.; Vilcek, J.; and Vlahavas, I. 2011. Mulan: A java library for multi-label learning. *J. of Machine Learning Res.* 12:2411–2414.

Tsoumakas, G.; Katakis, I.; and Vlahavas, I. P. 2010. Mining multi-label data. In *Data Mining and Knowledge Discovery Handbook*. 667–685.

Zhang, M.-L., and Zhang, K. 2010. Multi-label learning by exploiting label dependency. In *Proceedings of the 16th ACM SIGKDD*, KDD '10, 999–1008.