Mining Disorder Attributes with Rules and Statistical Learning

Xiaohua Liu, Xiaojie Liu, Wei Shen, and Jianyun Nie

DIRO, Universit de Montral, H3C 3J7, Qubec, Canada
liuxiao,xiaojie,shenwei,nie@iro.umontreal.ca

Abstract. This working note describes our approach to CLEF 2014 task 2a. It also reports our experimental results and discusses some future work we want to explore.

Keywords: attribute extraction, rule, classification

1 Introduction

The CLEF 2014 task 2a [1] aims to provide the normalized value for each of 10 attributes of each disease/disorder mention template. Each mention template consists of the mention's Unified Medical Language System concept unique identifiers (CUI) and mention boundaries, and a pointer to the report from which that mention template is extracted.

Each disease/disorder mention has the following 10 different attributes: Negation Indicator, Subject Class, Uncertainty Indicator, Course Class, Severity Class, Conditional Class, Generic Class, DocTime Class, Temporal Expression, and Body Location. The normalized value for any of the first nine of the attributes comes from a list of possible values, such as "yes", "no" for Negation Indicator. Normalized values for the tenth attribute-Body Location-come from the UMLS CUIs.

We decompose the task into 10 sub tasks, and consider each sub task as a classification problem, and accordingly design a classifier for each of the 10 attributes. We find that for some attributes, using simple rules, e.g., always setting it to some value, yields better results than statical learning. Therefore, in addition to developing some classifiers based on machine learning, we also build some rule based classifiers. Most of the rules are automatically extracted from the annotated training data. Furthermore, we manually create a few rules, for example, some regular expressions to identify time expressions.

The main goal of our experiments is to figure out the effective features or rules for each attribute extraction task. In our first attempt, we focus on such features and rules that can be directly extracted from the training data, not using any additional resources, such as WordNet [2], UMLS. While designing features, we only consider local features which can be easily extracted from a text window with the disease/disorder mention in the center.

[1] http://clefehealth2014.dcu.ie/task-2/2014-dataset

For some attributes, such as Generic Class and Negation Indicator, our system works well. But for some attributes, such as DocTime Class, it performs bad. We hope to better understand the reasons once the test data with the ground truth is released.

2 Our Approach

In our experiments, we design a classifier for each attribute. With cross validation on the training data, we find in most cases statistical classifiers achieve the best performance than rule bases systems. However, for "Body Location", since there are many possible values, a statistical classifier does not work. We also find that for some attribute, such as "Course Class", always setting it to a default value gives the best performance.

In what follows, we describe details of each classifier. Note that for each attribute, the type of classifier and the features sets are determined with 10 fold cross validation on the training corpus.

2.1 Negation Indicator Classifier

We train a classifier with LIBLINEAR [2]. While training, we set the type of solver to L1-regularized logistic regression, i.e., -s 6, and the regularization and experimental loss trade-off parameter to 3, i.e., -c 3.

We consider the following features in a text windows of size 17 with the mention in the middle: 1) uni-gram on the left/right of the mention; and 2) bi-gram on the left/right of the mention. For each n-gram, we append ":L" (":R") to its end, indicating it is on the left (right) of the mention.

On the official test data, its accuracy is 0.922.

2.2 Subject Class Classifier

We train a classifier with LIBLINEAR with -s 6 and -c 3. We use only uni-gram in a text windows of size 17 with the mention in the middle. For each uni-gram, we also consider its position. For example, feature "weight:3" ("weight:-3")means "weight" is on the right (left) of the mention, and there are two words between them. On the official test data, its accuracy is 0.611.

2.3 Uncertainty Indicator Classifier

We train a classifier with LIBLINEAR with -s 6 and -c 3. We use two sets of features in a text windows of size 17 with the mention in the middle: uni-gram with position, the same features as used for Negation Indicator Classifier; and bi-gram features with ":L" or ":R" as suffix. On the official test data, its accuracy is 0.923.

[2] http://www.csie.ntu.edu.tw/ cjlin/liblinear/

2.4 Course Class Classifier

We always set its value to "true". On the official test data, its accuracy is 0.961.

2.5 Severity Class Classifier

We build a rule based system for this attribute. First we extract all clue words for each severity class. For example, for "SEVERE", we get clue words like "acute","sharp","critical". Then we consider all words (context words) in a text window of size 5 with the mention in the middle and select the severity class that consists of the greatest number of clue words that appear in the text window. If there is a draw, we randomly choose one from them. And in case there are no rules to apply, we use the default value.

On the official test data, its accuracy is 0.611.

Table 1. Clue Words for Each Severity Class.

Severity Class	**Clue Words**
SEVERE	acute severe,sharp,rapid,significant,critical,flash,abrupt acute onset,severely moderate to severe,significantly greater, pleuritic, moderately severe acute to subacute subacute, mild,extremely,advanced, high grade, profound extreme moderately to severely,acutely,markedly,high-grade,crushing severity marked,moderate-severe,more marked,extensive sharp or knife-like,moderate-to-severe sub acute moderate to large sized, breakthrough, substantial coarse, massively critically high, sudden onset,volatile massive,considerable
MODERATE	moderately,mildly,mild,moderate,modest,mild to moderate,dense mild-moderate,markedly,mild-to-moderate,moderate to large subacute,significant,mildy,large-to-moderate
UNMARKED	worsening,severity,elevated,increase
SLIGHT	trace,mild,slight,minimal,minimally,minimally displaced mildly,slightly,modest,minor,partially,much lesser extent minimal amounts,minimal amount,trivial,partial,min,little

2.6 Conditional Class Classifier

We always set its value to "false". On the official test data, its accuracy is 0.936.

2.7 Generic Class Classifier

We train a classifier with LIBLINEAR with -s 6 and -c 3. We use only uni-gram with position as features in a text windows of size 17 with the mention in the middle. On the official test data, its accuracy is 1.000.

2.8 Body Location Classifier

Since there are potentially many body location labels, a statistical classifier with a fixed number of labels will not work. Therefore, we build a rule based classifier for this attribute. Each rule is is a clue word body location pair, which are automatically extracted from annotations. Here are some examples: (inferoapical,C1299408), (liver,C0223884). The procedure of extracting rules is similar to what we have done for Severity Class. With the mined rules, we select the body location that has the greatest number of clue words that occur in the mention. In case of a tie, we randomly choose one from them. On the official test data, its accuracy is 0.635.

2.9 DocTime Class Classifier

We always set its value to "OVERLAP". On 10-fold cross validation, it outperforms Support Vector Machines (SVMs) [1] based classifier (with an accuracy of 0.411) and other rule based classifiers. However, on the official test data, its accuracy is very low, i.e., 0.024.

2.10 Temporal Expression Classifier

We build a rule base classifier for this attribute. We run two steps to construct the rules: 1) first we extract clues words for each "Temporal Expression", i.e., DATE, DURATION and TIME; and 2) we manually compile regular expressions based on the clue words to make them more general. Step 2 is necessary because that clue words related to temporal expression are often long, and contain concrete numbers, making them hard to be matched. Here are some examples of such rules: DATE ←\d{4}-\d{2}-\d{2}, DURATION ← ((several) | (\d+) | (one)|(two)|(three)|(four)|(five)|(six)|(seven)|(eight)|(nine)|(ten))\s+((minute) |(second)|(hour)|(day)|(week)|(wk)|(month)|(year)|(yr)|(mn))s?\s+((duration) |(interval)|(history)), where "DATE","DURATION" are labels. In total, we have 30 regular expression based rules.

To do the prediction, we consider a text window of size 11 with the mention in the middle, to which we apply all the compiled regular expressions. Finally we choose the label that has the greatest number of matched regular expressions. In case of a tie, we randomly choose one; in case not any regular expression is matched, we choose the default value, i.e., "none".

On the official test data, its accuracy is 0.824.

3 Conclusions and Future Work

We build 10 classifiers to handle Task 2a. These classifiers can be organized into three groups: the first group of classifiers always predict the same value. This strategy works very well for Course Class and Conditional Class, but does not work for DocTime Class; the second group of classifiers are based on rules

mined from clue words of disease/disorder mentions. The rules are used for majority based voting to output the prediction. Classifiers for Severity Class, Body Location and Temporal Expression belong to this group; the third group of classifiers are based on SVMs and mainly use n-gram features extracted from a text window with the mention in the middle. There are 4 instances in this group, i.e., classifiers for Negation Indicator, Subject Class, Uncertainty Indicator and Generic Class. Except for the classifier for Subject Class attribute, the classifiers in this group perform well with an accuracy of more than 0.900 on the official test data.

In our current experiments, we don't use any external resources, and don't use any features more complicated than n-grams. In future we would like to consider more resources and more advanced features, such as features from the values of related attributes, particularly for the attributes on which our current approach does not work well.

References

1. Burges, C.J.C.: A tutorial on support vector machines for pattern recognition. Data Min. Knowl. Discov. 2(2), 121–167 (Jun 1998), http://dx.doi.org/10.1023/A:1009715923555
2. Miller, G.A.: Wordnet: A lexical database for english. COMMUNICATIONS OF THE ACM 38, 39–41 (1995)