MEDINFO 2021: One World, One Health – Global Partnership for Digital Innovation
P. Otero et al. (Eds.)

doi:10.3233/SHTI220269

Comparing Multiple Models for Section Header Classification with Feature Evaluation

Ronak Pipaliya[a], Paul M. Heider[b], Stéphane M. Meystre[b]

[a] College of Medicine, [b]Biomedical Informatics Center, Medical University of South Carolina, Charleston, SC, USC

Abstract

We present on the performance evaluation of machine learning (ML) and Natural Language Processing (NLP) based Section Header classification. The section headers classification task was performed as a two-pass system. The first pass detects a section header while the second pass classifies it. Recall, precision, and F1-measure metrics were reported to explore the best approach for ML based section header classification for use in downstream NLP tasks.

Keywords:
Natural Language Processing; Machine Learning

Introduction

Electronic Health Records (EHR) are full of clinical information that is vital for healthcare providers and researchers to conduct studies or make clinical decisions. NLP techniques and ML can be used to extract clinical information from EHR narrative text notes. NLP based tools will often extract information like named entities and relations between concepts for these tasks. For clinical NLP tasks in particular, extracting the structure of the clinical note in terms of sections is important. The most popular ML section header classifiers are Conditional Random Fields and Support Vector Machines [1]. We evaluated various classifiers and features to explore the best approach to classifying section headers.

Methods

A reference corpus of eighty-six clinical notes was annotated for section headers normalized to an ontology of thirty-nine section header types. Each note was processed using an Apache UIMA NLP pipeline with uimaFIT to perform sentence boundary detection (SBD) and model training [2]. We split notes into sentences using the cTAKES SBD module followed by a regular expression module to further split spans of text that contained multiple sentences. The section header classification task occurred in two passes. Each pass used a machine learning model trained using the Weka API [3]. The first pass performed a binary classification on a sentence to determine if it is a section header or a simple sentence. The second pass classified the section headers (from the previous pass) into one of the thirty-nine section header categories. The section header ontology is available at https://github.com/MUSC-TBIC.

Section Header/Sentence Classification

The classifiers that were evaluated for the binary section header identification task were Support Vector Machines (SVM, with SMO), decision trees (J48), Random Forest, and Naïve Bayes. The features included to train the ML models were based on the sentence text normalized through down casing and removal of punctuation, and morphological features (punctuation type, text casing, and presence of all uppercase lettering in the sentence). The sentence text and punctuation were filtered using a String2WordVector filter available in Weka. The training data and its features were generated using the sentences retrieved after cTAKES SBD and regular expression splitting. A model with each classifier was trained and tested on these features using 10-fold cross validation. We report precision, recall, and F_1-measure with averages (both micro and macro).

Section Header Categories Classification

The classifiers evaluated for the multi-class section header classification task were the same as the binary classification task. The features included were the same as the binary task with some additions: relative position of section header in the note, number of section headers in note, the previous two section headers, and the last non-subsection section header. Again, the sentence string was filtered using a String2WordVector filter. The training data was generated from the corpus using sentences that would have been classified as a section header by the binary classifier assuming perfect performance (i.e., rather than training on the real end-to-end system). Again, we trained and tested using 10-fold cross validation.

Feature Exploration

The multi-class classification training data was used to generate models with five different feature sets. The first set was the **base** features included in every evaluation (i.e., features used for the binary classification). The **second** set included the base features and the relative position of the sentence in the note. The **third** set was made of the base features and the number of sections in the note. The **fourth** group included the base features, the last section header, the section header before last (penultimate section), and the last non-subsection section header. The **fifth** group has the same feature set as the multi-class classification (i.e., all features). These sets were trained and tested using 10-fold cross validation.

Results

Corpus

The annotated corpus contained a total of 2855 section headers. After cTAKES SBD, 2147 section headers (75%) were captured as sentences without any over- or under-splitting. The SBD left some section headers with accessory characters before or after it. After the regular expression module, an additional 122 section headers were correctly split into stand-alone sentence spans, leading to a total of 2269 (79%). Each of the 2269

headers belong to one of the 39 section header classes. Of the 39 classes, 18 of them contain less than 20 instances each.

Section Header/Sentence Classification

The binary classifiers were answering the question, "Is this a section header?" for each sentence in the corpus. The recall, precision, and F_1- measure reported in Table 1 are an average of the 10 cross validation folds. Random Forest had the highest precision and F_1-measure while the SVM (SMO) had the highest recall. Following those two classifiers, the J48 decision tree algorithm had a slightly lower performance on all three metrics. The Naïve Bayes classifier was the worst of the four classifiers.

Table 1 – Binary Performance by Classifier

Classifier	Recall	Precision	F_1-Measure
SVM (SMO)	0.9449	0.8900	0.9166
Naïve Bayes	0.7532	0.6922	0.7214
Decision Tree	0.8797	0.8440	0.8615
Random Forest	0.9127	0.9320	0.9223

Section Header Categories Classification

The multi-class classifiers trained on the 2269 sentences that contained a section header were evaluated using 10-fold cross-validation. Table 2 contains the macro- and micro-averaged precision, recall, and F_1-measure. The macro-averages are noticeably lower than the micro-averages. This could be due to the handful of types of very infrequent section categories in the training data (cf. Figure 1 for a comparison of performance on frequent and infrequent types). The micro-averaged precision, recall, and F_1-measure for the SVM are highest at 0.9304. Random Forest performed the worst with micro-averages at 0.9022.

Table 2 – Multi-Class Performance by Classifier

Classifier	Type	Recall	Prec.	F_1-Measure
SVM	Macro	0.6239	0.6689	0.6456
	Micro	0.9304	0.9304	0.9304
Naive Bayes	Macro	0.6084	0.6393	0.6235
	Micro	0.9101	0.9101	0.9101
Decision tree	Macro	0.5832	0.6118	0.5972
	Micro	0.9048	0.9048	0.9048
Random Forest	Macro	0.5787	0.6368	0.6063
	Micro	0.9022	0.9022	0.9022

Feature Evaluation

The micro and macro-averaged F_1-measure for each classifier and feature set used is reported in Figure 2. Including relative position in the note alongside the base features leads to an increase of micro F_1-measure with the decision tree and SVM classifiers. The base features together with either the number of sections results or the previous section headers lead to an increase in micro F_1- measure of the Naïve Bayes. The base features alongside the previous section headers show lower micro F_1- measure for all other classifiers from base alone. The highest micro F_1- measure of all the classifiers is the SVM based model which is using the base features and the relative position features with a value of 0.9374.

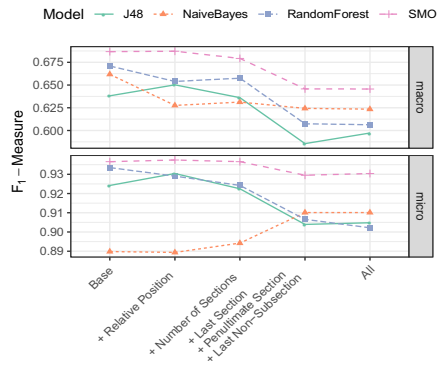

Figure 2- Feature Sets Performance by Classifier (Base Includes: All Caps, Title Case, Punctuation)

Conclusions

Our evaluation of classifiers and features in this two pass system provides insight into the best approach for researchers implementing section header annotation and classification in an NLP tool. Further expansion of the features, evaluation of the entire system as a whole, and using multiple corpora from different sites should provide researchers more insight into the performance of a two pass system and its generalizability.

Acknowledgements

This work was supported in part by funding from PCORI (ME-2018C3-14549).

References

[1] A. Pomares-Quimbaya, M. Kreuzthaler, and S. Schulz, Current approaches to identify sections within clinical narratives from electronic health records: a systematic review, BMC Med Res Methodol. **19** (2019).

[2] P. Ogren, S. Bethard, Building Test Suits for UIMA Components, Proceedings of the Workshop on Software Engineering, Testing, and Quality Assurance for Natural Language Processing (SETQA-NLP 2009), (2019).

[3] E. Frank, M. Hall, and I. Witten, The WEKA Workbench, 4th ed. Morgan Kaufmann, 2016

Address for correspondence

Ronak Pipaliya at pipaliyr@musc.edu

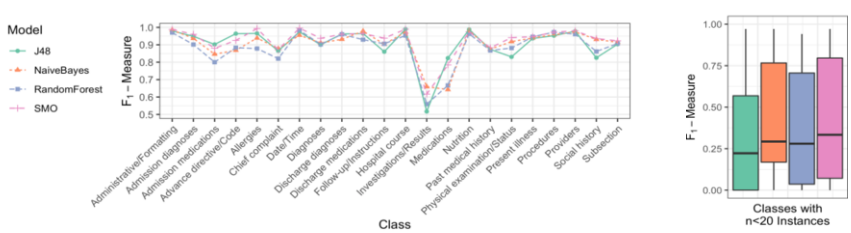

Figure 1- Model Performance for Individual Frequent Headers Types (Left; scale 0.5 to 1) and Average Performance for Infrequent Header Types (Right; n < 20 Instances; scale 0.0 to 1