HIDS-6001: Project Report

Smoking Status Identification from Clinical Notes

by Joan Mattle, Yuktha Penumala, Aizhan Uteubayeva

1. Introduction

Our project is modeled after the National NLP Clinical Challenges (n2c2) on automatically determining the smoking status of patients from discharge records. This challenge was hosted by Harvard Medical School, as part of the i2b2 (Informatics for Integrating Biology to the Bedside) project. There were two main tasks of the challenge: automatic de-identification of clinical data, i.e., de-identification challenge and automatic evaluation of the smoking status of patients based on medical records, i.e., smoking challenge. Specifically, there were a total of 502 de-identified medical discharge records and a total of 11 teams with 23 submissions. There were 12 systems with micro averaged F-measures above 0.84, using machine learning models such as SVMs, kNNs, logistic regression, and more.

Building upon the foundations laid by the n2c2 challenge, our project navigated the complexities of de-identified clinical text by similarly defining key textual features, employing tokenization, and leveraging classification models. Our focus extended to the exploration of both non-reduced and reduced text, mirroring the real-world challenge of handling extensive clinical notes. We trained three machine learning models (multiNomialNB, logistic regression, Random Forest) using the best parameters on the non-reduced text and reduced text columns of an unannotated set of clinical notes (test set). In addition, we have attempted to employ a deep-learning model Keras for further evaluation of the techniques. Across all three NLP models, we obtained consistent performance evaluation metrics, however, Multinomial Naive Bayes was the best performing model.

All in all, in the scope of healthcare, our project endeavors to contribute to the automation of determining patients' smoking status from clinical records, aligning with the broader challenge posed by the National NLP Clinical Challenges (n2c2). Our models performed overall well, which aligns with the importance of utilizing NLP in healthcare since the sheer volume and intricacy of clinical data makes manual extraction a laborious and error-prone task. Ultimately, then, our project contributes to improving efficiency and streamlining clinical workflows in the healthcare landscape, which ultimately leads to better patient outcomes.

2. Task

The task of this project is to automatically determine the smoking status of patients from information found in their discharge records (clinical notes). There are five possible smoking status categories: past smoker, current smoker, smoker, non-smoker, and unknown.

3. Methods

3.1 Processing the data

Our dataset comprised a training set, a testing set, and a testing set labeled with smoking statuses. Our initial step involved the creation of "reduced text" columns for both the training and testing datasets. This process entailed defining a regular expression pattern to identify smoking-related sentences and employing a function to extract such sentences from the original text columns. Additionally, we ensured the absence of null values in the newly formed "reduced text" column.

3.2. Training three NLP models

Subsequently, we trained three models—Multinomial Naive Bayes, Logistic Regression, and Random Forest—utilizing the training data. We established a scikit-learn pipeline for text classification, incorporating a TfidfVectorizer for tokenization and TF-IDF transformation, coupled with the specified classifier. Employing grid search, we determined the optimal hyperparameter values for each model. These optimized parameters were applied to both the "text" and "reduced text" datasets during model training. Consequently, predictions were generated for the test set's non-reduced and reduced text using each model. Utilizing classification reports, we evaluated the accuracy of each model under the best parameters as well as alternative parameter configurations.

3.3 Keras Model

Keras is building a deep learning model, specifically a recurrent neural network (RNN) using LSTM (Long Short-Term Memory) cells, utilized for text classification tasks. The methodology employed for smoking status prediction involves several sequential steps. Initially, the textual data from both training and test sets are loaded, where the training set's text content and corresponding smoking status labels are stored in X_train_red and y_train, while X_test_red and y_test contain the test set's information. Tokenization and padding techniques are applied using Keras Tokenizer and pad_sequences to convert text sequences into fixed-length numerical representations. Subsequently, label encoding via LabelEncoder and conversion to categorical one-hot encoding using Keras' to_categorical are implemented to prepare the target labels for multi-class classification. The model architecture, constructed using Keras' Sequential API, involves an Embedding layer for word embeddings, followed by an LSTM layer with 128 units, an additional Dense layer with 128 units and ReLU activation, and an output layer with softmax activation. Training occurs over 25 epochs, utilizing the Adam optimizer and categorical cross-entropy loss, with a 10% validation split for monitoring

performance. The train_model() function encapsulates this process, generating the model architecture, compiling it, and executing the training while capturing the training history for subsequent analysis or evaluation.

4. Results

In overall evaluation, the best model used is the Multinomial Naive Bayes (MNB) with an accuracy score of 86.54%. To provide a point of comparison, here's the evaluation metrics of the non-reduced vs. the reduced text. Please note, the tables display the macro averages and weighted averages of the five smoking smoking status categories, as directly reported by the classification reports.

4.1 MNB using non-reduced text and optimal parameters

Optimal parameters: clf__alpha: 0.01, tfidf__ngram_range: (1, 1), tfidf__use_idf: False. Accuracy 62.50%.

	Precision	*Recall*	*F1-score*	*Support*
Macro averages	0.22	0.24	0.22	104
Weighted averages	0.46	0.62	0.52	104

The accuracy score obtained was 62.50%, meaning our model correctly predicted the smoking status for 62.50% of the instances in the non-reduced text. Looking at the Precision, which is the measure of the accuracy of the positive predictions, around 22% of the positive predictions were correct (macro) and around 46% (weighted). Then, looking at the recall, which is the proportion of actual positive instances that were correctly predicted, around 24% of actual positives were correctly predicted (macro) and around 62% (weighted). For F1-score, which is the harmonic mean of precision and recall, on average, around 22% of the positive predictions were correct while also accounting for the proportion of actual positives (macro), and around 52% (weighted). Lastly, the support is 104, which is just the number of actual occurrences of each class in the specified order (past smoker, current smoker, smoker, non-smoker, unknown)

4.2 MNB using reduced text and optimal parameters

Optimal parameters: clf__alpha: 0.01, tfidf__ngram_range: (1, 3), tfidf__use_idf: True. Accuracy on the reduced test set: 86.54%

	Precision	*Recall*	*F1-score*	*Support*
Macro averages	0.62	0.66	0.63	104
Weighted averages	0.86	0.87	0.86	104

For the reduced-text column, we obtained an accuracy score of 86.54%, which makes sense since we defined a regular expression pattern (smoking_patt) that matches words related to smoking, thus reducing noise and efficiently tokenizing. Looking at the Precision, on average, around 62% of the positive predictions were correct (macro) and around 86% (weighted). Then, looking at the recall, on average, around 66% of actual positives were correctly predicted (macro) and around 88% (weighted). For F1-score, on average, around 63% of the positive predictions were correct while also accounting for the proportion of actual positives (macro), and around 86% (weighted). Lastly, the support is the same 104 because it represents the number of patients.

4.3 Multinomial Naive Bayes with various n-grams

	Precision	*Recall*	*F1-score*
Non-reduced + ngram(1,1)	**0.47**	**0.62**	**0.53**
Non-reduced + ngram(1,2)	0.37	0.61	0.46
Non-reduced + ngram(1,3)	0.37	0.61	0.46
Reduced + ngram(1,1)	0.84	0.84	0.83
Reduced + ngram(1,2)	0.85	0.85	0.85
Reduced + ngram(1,3)	**0.85**	**0.86**	**0.85**

We then played around with some additional parameters to see if there was another n-gram that optimizes the performance of a model, while we kept the other two parameters of smoothing parameter and IDF reweighting constant. These numbers represent the weighted averages. Right off the bat, you can tell that any combination of the reduced text columns had much higher evaluation metrics than the non-reduced text columns. For the non-reduced text, as the n-gram size increases from unigram/trigram, precision, recall, and F1-score decrease. This could indicate that the model considers pairs/trios of adjacent words as features, thus capturing more noise and resulting in lower metrics.

However, for the reduced text, from unigram to bigram/trigram, there's a slight increase in precision, recall, and F1-scores. This could indicate that additional words such as not, quit provides more context, capturing more accurate relationships of smoking status. Overall, among the different n-gram settings for the reduced text, precision, recall, and F1-score are consistently high across. This suggests that the reduced text, along with appropriate n-gram settings, performs well, likely due to its focus on relevant information. Overall, the best performance is observed for the reduced text with n-gram(1,2) and n-gram(1,3), where precision, recall, and F1-score are relatively high and balanced.

4.4 Keras Model Evaluation

The model's performance in predicting smoking status exhibits notable disparities across different classes. It demonstrates relatively strong precision and recall values for classes like 'NON-SMOKER' and 'UNKNOWN,' with precision values of 0.93 and 0.97, respectively. These high precision scores indicate the model's ability to accurately classify instances within these categories, particularly in identifying individuals who do not smoke or have an unknown smoking status. However, the model struggles notably with the 'SMOKER' class, showing precision, recall, and F1-score of zero, implying an inability to effectively identify individuals who smoke. This imbalance in performance across classes suggests potential challenges in capturing the nuanced patterns or features specific to smokers within the dataset.

The overall metrics, while indicating decent precision and recall scores around 0.84, underscore the need for further model refinement, especially in enhancing its ability to discern instances within the 'SMOKER' category, in order to achieve a more balanced and accurate predictive performance across all smoking status classes.

Class-wise Metrics:

	Class	Precision	Recall	F1-Score
0	CURRENT SMOKER	0.533333	0.727273	0.615385
1	NON-SMOKER	0.928571	0.812500	0.866667
2	PAST SMOKER	0.538462	0.636364	0.583333
3	SMOKER	0.000000	0.000000	0.000000
4	UNKNOWN	0.967742	0.952381	0.960000

Overall Metrics:

	Overall Precision	Overall Recall	Overall F1-Score
0	0.842448	0.846154	0.841659

5. Discussion

In alignment with the Harvard challenge, our project focused on defining key textual features, employing tokenization, classification models, and pipelines for parameter optimization. We trained three machine learning models—Multinomial Naive Bayes, Logistic Regression, and Random Forest—using the best parameters on both non-reduced and reduced text columns of an unannotated set of clinical notes (test set). Across all three models the use of reduced text consistently outperformed non-reduced text. Employing reduced text resulted in significantly improved accuracy across the models, with Multinomial Naive Bayes achieving the highest accuracy of 86.54%. Overall, the NLP model demonstrated commendable performance in predicting smoking status. Employing the Keras model, has shown promising results in some class-wise performance. While our project has shown promising results in automatically determining the smoking status of patients from clinical notes, a longer timeline would have allowed for more in-depth exploration and refinement. Given a year, we could have conducted more extensive hyperparameter tuning for our models. Exploring a wider range of hyperparameters and considering more advanced optimization techniques might have led to further improvements in model performance.

Additionally, with an additional year, we could have explored more complex machine learning models including richer deep learning models and techniques such as attention mechanisms. We also could even have incorporated hybrid models, combining traditional machine learning models and deep learning models. Then, using hybrid models, we would have more refined techniques such as ensemble techniques or adaptive learning, which would leverage the strength of multiple algorithms. All in all, using more sophisticated models would have helped us capture stronger relationships and intricate patterns, ultimately enhancing the accuracy and fine tuning results.

References

3.2. Tuning the hyper-parameters of an estimator [WWW Document], n.d. . scikit-learn. URL https://scikit-learn/stable/modules/grid_search.html (accessed 12.18.23).

Brownlee, J., 2017a. A Gentle Introduction to the Bag-of-Words Model. MachineLearningMastery.com. URL https://machinelearningmastery.com/gentle-introduction-bag-words-model/ (accessed 12.18.23).

Brownlee, J., 2017b. How to Encode Text Data for Machine Learning with scikit-learn. MachineLearningMastery.com. URL https://machinelearningmastery.com/prepare-text-data-machine-learning-scikit-learn/ (accessed 12.18.23).

Python, R., n.d. Practical Text Classification With Python and Keras – Real Python [WWW Document]. URL https://realpython.com/python-keras-text-classification/ (accessed 12.18.23).

sklearn.feature_extraction.text.TfidfVectorizer [WWW Document], n.d. . scikit-learn. URL https://scikit-learn/stable/modules/generated/sklearn.feature_extraction.text.TfidfVectorizer.html (accessed 12.18.23).

Team, K., n.d. Keras documentation: The Model class [WWW Document]. URL https://keras.io/api/models/model/ (accessed 12.18.23).

tf.keras.Model | TensorFlow v2.14.0 [WWW Document], n.d. . TensorFlow. URL https://www.tensorflow.org/api_docs/python/tf/keras/Model (accessed 12.18.23).

Uzuner, Ö., Goldstein, I., Luo, Y., Kohane, I., 2008. Identifying Patient Smoking Status from Medical Discharge Records. J Am Med Inform Assoc 15, 14–24. https://doi.org/10.1197/jamia.M2408

Working With Text Data [WWW Document], n.d. . scikit-learn. URL https://scikit-learn/stable/tutorial/text_analytics/working_with_text_data.html (accessed 12.18.23).