Automated Annotation of Immuno-Therapy Related Toxicities in Clinical Notes Using Machine Learning, Natural Language Processing, and Robust Data Sampling

INNOVATION CENTER FOR BIOMEDICAL INFORMATICS
Georgetown | Lombardi COMPREHENSIVE CANCER CENTER
GEORGETOWN UNIVERSITY Georgetown University Medical Center

Suchet Sapre[1], Samir Gupta[2], Subha Madhavan[2]

[1]Thomas Jefferson High School for Science and Technology, Alexandria, VA
[2]Innovation Center for Biomedical Informatics, Georgetown University Medical Center, Washington DC

SUMMARY/Abstract

OBJECTIVES/SPECIFIC AIMS

The current standard for the annotation of treatment outcomes in clinical notes is both a labor-intensive and time-consuming process. Specifically, doctors have to manually read through notes that can be thousands of words long, a procedure that can take upwards of 1 year for annotating the immuno-therapy related toxicities of just 724 patients. With the rapid growth in the versatility of modern computing, the automation of this task is a necessity. Thus, the goal of this project is to evaluate various Natural Language Processing (NLP) and Machine Learning (ML) based models in their ability to annotate medical notes. This project only focuses on the annotation of Colitis.

METHODS

This project consisted of three main steps: preprocessing the data, training the NLP model (Word2Vec/Doc2Vec), and constructing a ML classifier that would implement this NLP model. Firstly, we had to concatenate the text files for each patient, while doing this we filtered out sentences that did not contain keywords relating to "Colitis". Then, each patient's note data was tokenized through the Keras tokenization algorithm. Concurrently, Gensim's Word2Vec algorithm was applied to the patient's note data, creating the word embeddings used for this project. Then, many different ML approaches were tested to determine which could best annotate adverse events in medical notes. The approaches tested were Artificial Neural Networks (ANNs), Convolutional Neural Networks (CNNs), usage of class weights, and two data sampling algorithms: k-Nearest-Neighbor Oversampling and NearMiss Undersampling. Finally, each of these models was evaluated by measuring their precision, recall, F1-score, and ROC-AUC.

RESULTS

Here we show that employing word-embeddings to vectorize the key-word ("Colitis") filtered medical notes along with a Convolutional Neural Network with Class Weights yields the best results. This approach showed promising results with an F1-score of 0.641 and ROC-AUC of 0.808 from a 10-fold Cross Validation (CV).

CONCLUSION/FUTURE WORKS

This research shows the potential in automating the annotation of adverse events in clinical note data. Many of the evaluation results from the various trials exhibit to low precisions. We believe that this can be attributed to the tendency for the classifiers to predict false positives. The data sampling techniques tested in this project (over and under sampling) both proved to be unbeneficial in this situation.

With respect to future works, a promising next step would be finding optimized methods of removing unnecessary information from the clinical notes. This would prevent the ML classifiers from misinterpreting the random noise. Furthermore, implementing a "stacked" classifier can potentially help solve the problem of excess false positives and would consequently bolster the performance of our model. In summary, significant work still needs to be done in generalizing this procedure to all toxicities so that it can have a greater impact on the medical community.

Natural Language Processing Techniques

Text Tokenization

Description: Text Tokenization replaces words with their popularity or frequency rank in a given corpus of text.

FIGURE 1. Tabular Representation of Keras' Tokenization Algorithm

Note Data: "The boy liked the ice cream."

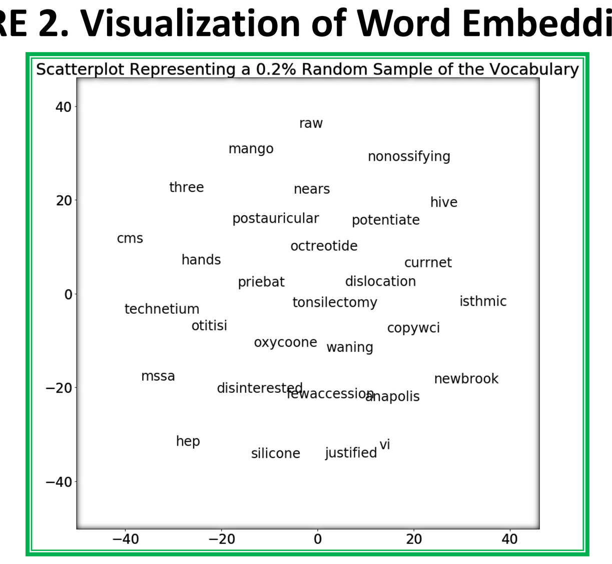
Word	Frequency	Frequency Rank
the	2	1
boy	1	2
liked	1	3
ice	1	4
cream	1	5

Tokenized Vector: < 1, 2, 3, 1, 4, 5 >

Word2Vec

Description: The Word2Vec algorithm creates a vector representation of each word in a vocabulary by analyzing that word's context (i.e. the words surrounding it).

FIGURE 2. Visualization of Word Embeddings



Embedding Matrix

Description: The Embedding Matrix embeds the tokenized vectors with the word embeddings

FIGURE 3. Tabular Representation Embedding Matrix

Tokenized Vector: < 1, 2, 3, 1, 4, 5 >

Word	Frequency Rank	Word Embedding Vector (dim = 3)
the	1	<4.4, 1.3, 9.8>
boy	2	<5.8, 8.7, 0.6>
liked	3	<7.4, 5.3, 2.2>
ice	4	<9.9, 9.1, 3.9>
cream	5	<5.6, 7.1, 8.0>

Embedding Matrix: <4.4, 1.3, 9.8> <5.8, 8.7, 0.6> ⋮ <9.9, 9.1, 3.9> <5.6, 7.1, 8.0>

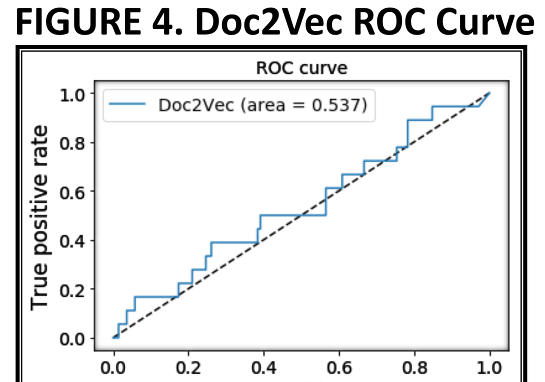
Doc2Vec

Description: Doc2Vec allows for the vectorization of entire documents. This method performed much worse than Word2Vec so it will not be analyzed in detail on this poster.

TABLE 1. Doc2Vec Results

CNN - No CV, With Class Weights		
Precision	Recall	F1 Score
0.285714	0.111111	0.160000

FIGURE 4. Doc2Vec ROC Curve



Machine Learning Algorithms and Topologies

Word2Vec Approach

Classifier 1: Artificial Neural Networks (ANNs)

Description: ANNs function by inputting data through a series of weighted layers (matrices).

FIGURE 5. Illustration of an ANN

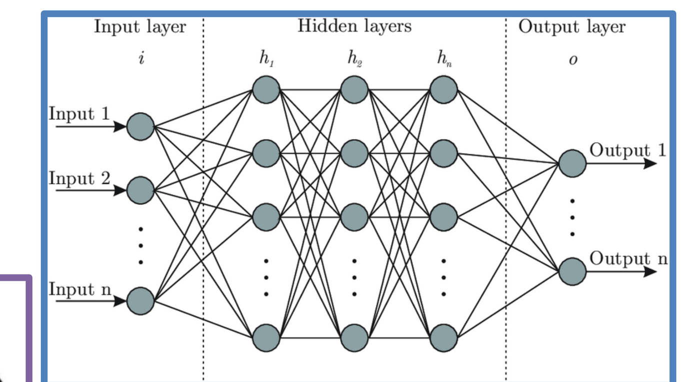


FIGURE 6. Mathematical Portrayal of One Neuron

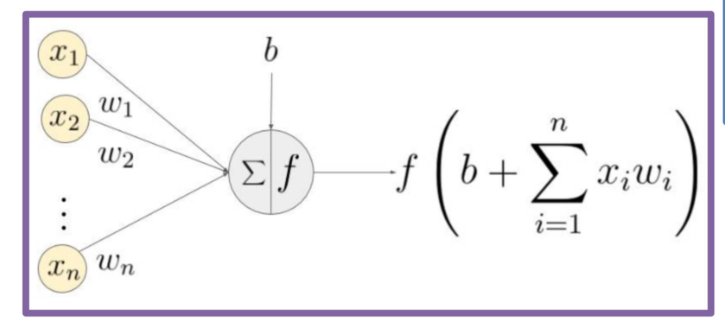


TABLE 2. Confusion Matrix

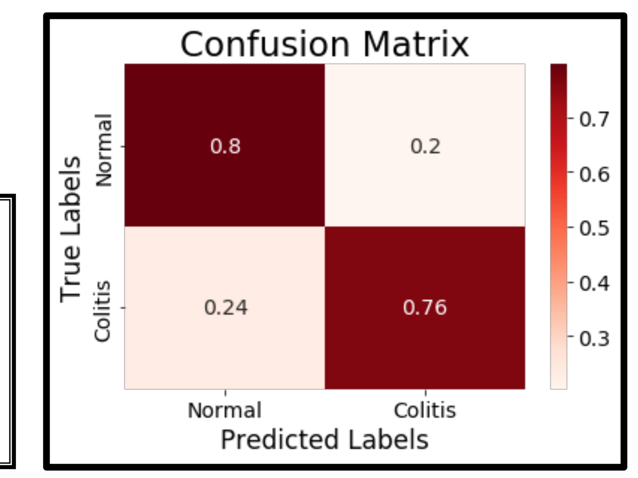


TABLE 3. Classification Metrics

ANN - 10 Fold CV with Class Weights		
Precision	Recall	F1 Score
0.297853	0.764881	0.413110

Classifier 2: Convolutional Neural Networks (CNNs)

Description: CNNs are a subset of ANNs which take into account the order and local features of the input data

FIGURE 7. CNN Illustration

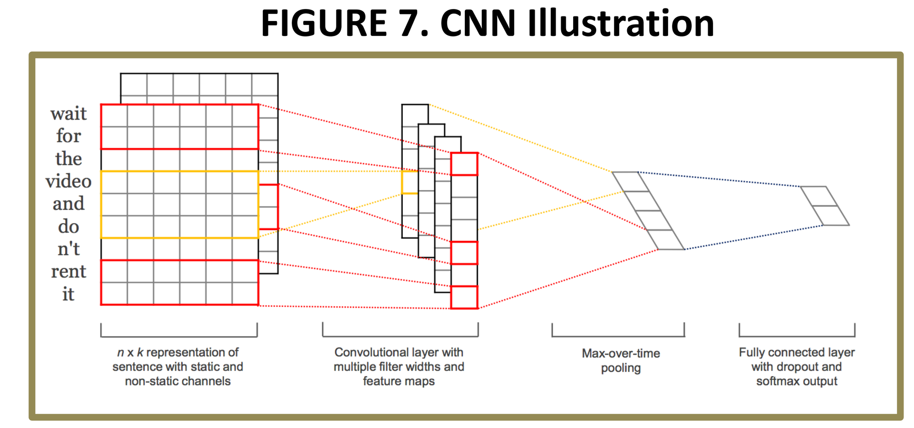


FIGURE 8. CNN Topology

Embedding Matrix

Convolutional Layer:
- 400 Filters
- Activation Function: ReLU

$ReLU(x) = \max(0, x)$

Dropout Layer (0.5): Ignores 50% of nodes in this layer, serves as a regularization mechanism

Hidden Layer:
- 30 Nodes
- Activation Function: ReLU

Output Layer:
- 1 Node
- Activation Function: sigmoid

$$sigmoid(x) = \frac{1}{1 + e^{-x}}$$

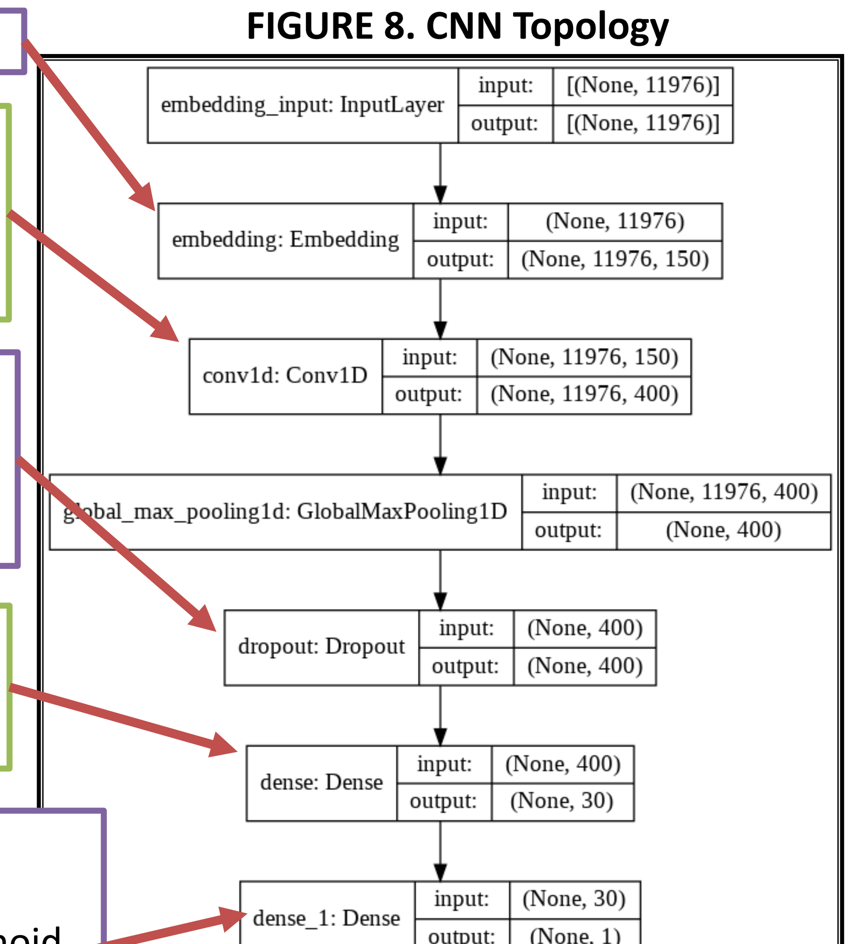


TABLE 4. Confusion Matrix

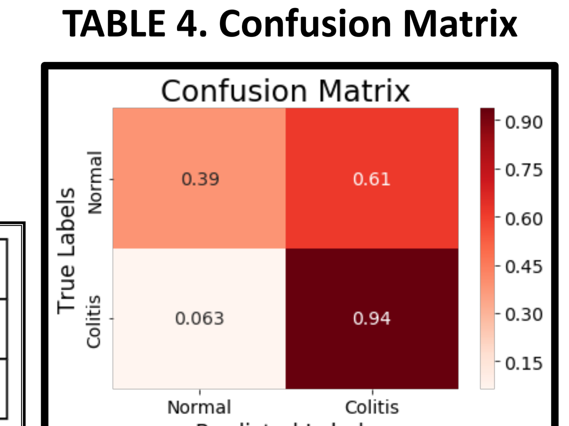


TABLE 5. Classification Metrics

CNN - 10 Fold CV with Class Weights			
Precision	Recall	F1 Score	ROC-AUC
0.6508044733044733	0.657143	0.640945	0.807857

Clinical Note Dataset and Preprocessing

Keyword ("Colitis") Filtering

Description: Removing sentences if they do not contain the word "Colitis" or its synonyms.

Data Sample (Symplified)

TABLE 6. Data was obtained from *MedStar Health's Electronic Health Record System*

Patient Number	Note Data	Colitis?*
0000	"… the pt was subsequently enrolled in clinical trial 701 (denileukin) and completed 4 cycles …"	Yes or No

*This project only focuses on the annotation of Colitis.

Data Sampling Algorithms

K-Nearest-Neighbor Oversampling

New Point Generation Formula

$$x_{new} = x_i + \lambda \cdot (x_{i1} - x_i)$$

FIGURE 9. Visualization of kNN Oversampling

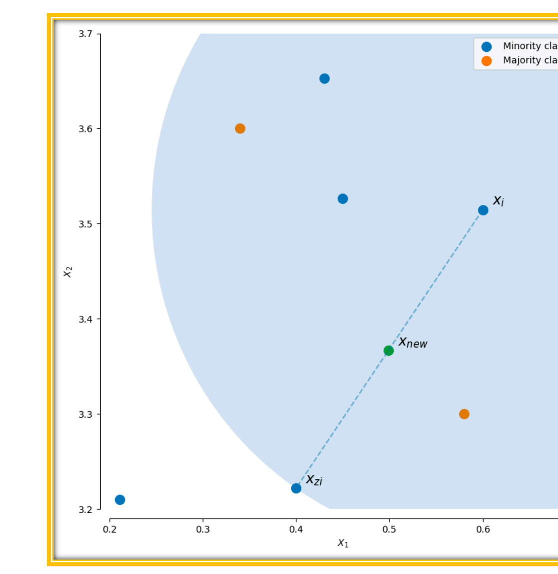


TABLE 7. Confusion Matrix

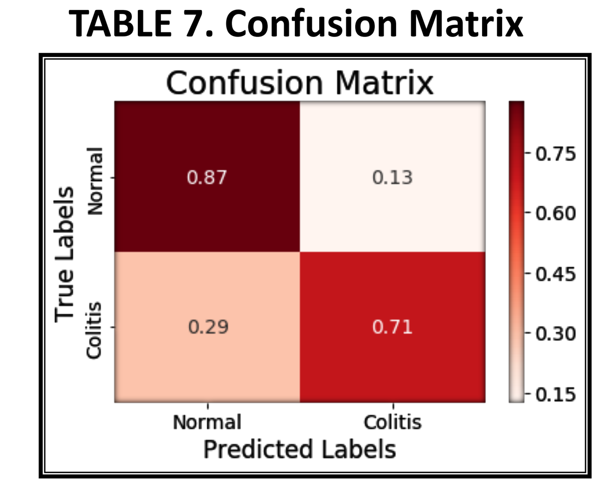


TABLE 8. Classification Metrics

2.5 Times Minority Class Oversampling		
Precision	Recall	F1 Score
0.383472	0.707143	0.489025

NearMiss Undersampling

TABLE 9. Confusion Matrix

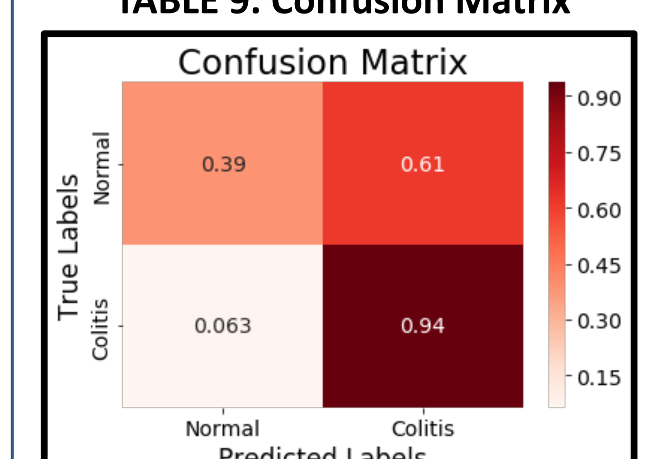


FIGURE 10. Visualization of NearMiss Undersampling

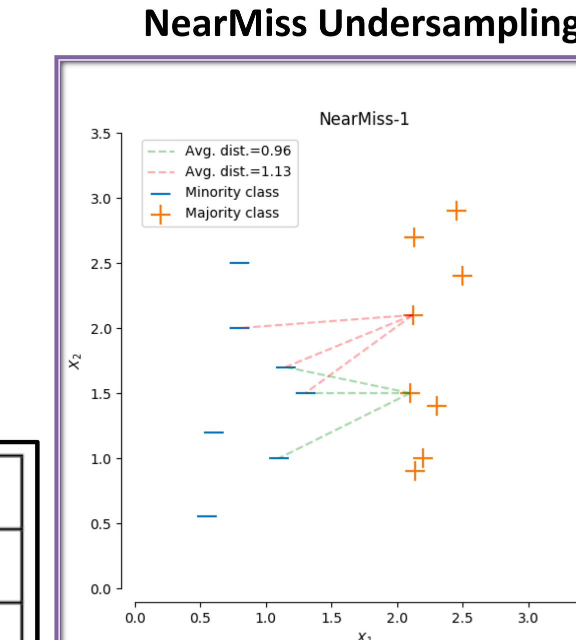


TABLE 10. Classification Metrics

Balanced Undersampling		
Precision	Recall	F1 Score
0.146763	0.936667	0.251035

Major Citations

[1] Nogueira, F., Lemaitre, G., Victor, D., & Aridas, C. (n.d.). Imbalanced-Learn Documentation. Retrieved August 6, 2019, from https://imbalanced-learn.readthedocs.io/en/stable/index.html#.

[2] Brownlee, J. (2019, August 7). How to Develop an Multichannel CNN Model for Text Classification. Retrieved August 10, 2019, from https://machinelearningmastery.com/develop-n-gram-multichannel-convolutional-neural-network-sentiment-analysis/.

[3] Sharma, A. (2017, October 30). Understanding Activation Functions in Deep Learning. Retrieved September 15, 2019, from https://www.learnopencv.com/understanding-activation-functions-in-deep-learning/.

[4] Bre, Facundo & Gimenez, Juan & Fachinotti, Víctor. (2017). Prediction of wind pressure coefficients on building surfaces using Artificial Neural Networks. Energy and Buildings. Retrieved September 17, 2019, from 158. 10.1016/j.enbuild.2017.11.045.