Investigating the Use of Statistical Tests in Machine Learning Models

Jule Valendo Halim -1425567
University of Melbourne

Abstract

Objective

Materials

Methods

Results

Discussion

Conclusion

Introduction

Statistical tests are a popular and long-standing method in multiple fields of science. However, studies in natural language processing (NLP) and machine learning (ML) do not generally include statistcal testing as part of their model evaluation. A survey on 233 published papers in the field of NLP showed that 132 of these papers did not report statistical significance[1]. However, more studies have begun advocating for the use of these tests to show that experimental results are not coincidental[1] and argue that a combination of NLP and statistical tests can provide a framework for the development of robust, high-throughput health NLP systems[2].

In this report, I aim to investigate the use of statistical tests on ML models that predict multi-label and multi-class classification tasks using the statistical test workflow suggested by Rainio, Tauho, and Klén[3], as seen in figure 1. However, instead of comparing results from different test sets, this report compares different model feature inputs.

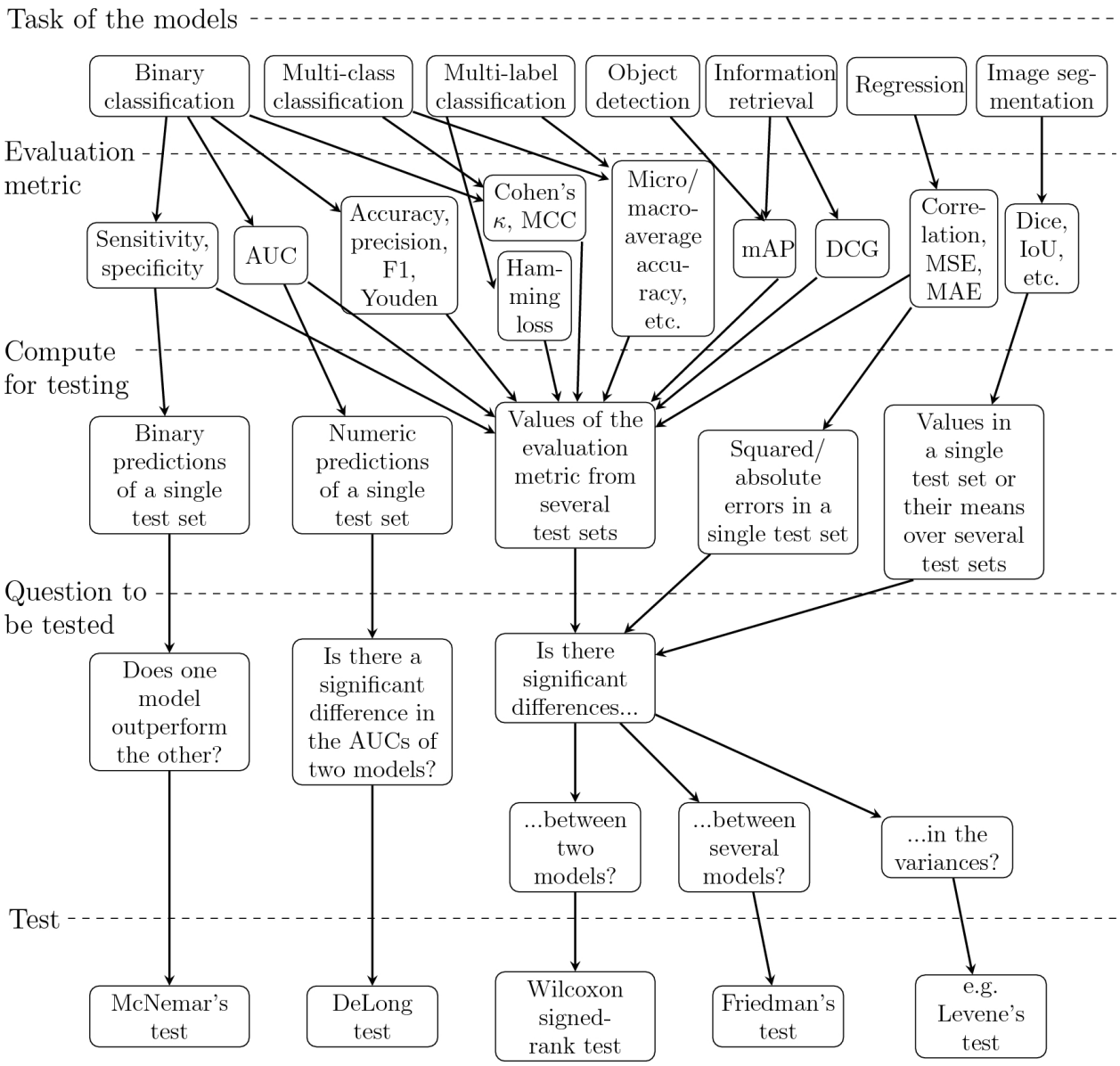

Figure 1. Statistcal Workflow Provided by Rainio, Tauho, and Klén [3]

The Sentence_Labeling sheet of the PsyTAR dataset used in this study is provided by Zoolnoori et al [4] and Zoolnori et al [5]. This sheet contains three sections of interest. The first is sentences from online patient reviews of certain drugs. The second is annotations of certain labels, described below in table 1. The third is drug labels, which indicates which drug each sentence is reviewing.

Two classification tasks were performed using a baseline logistic regression model and a transformer model. The multi label class predicts six binary annotations, while the multi-class predicts four different drugs. For each model, different features will be used as inputs. The resulting predictions will be used for statistical tests in order to determine their statistical signifance. Specific details are described in the methods section.

Methods

Preprocessing

Two main preprocessing steps were done. Firstly, as the drug labels in the data were concatenated with review ID, the review ID was stripped and only the drug label was added to a new column. Secondly, the review text was preprocessed through tokenization, as well as stopword and punctuation removal. The dataset consists of 6009 records. The data was split into train, validation, and test sets using a 45:45:10 ratio.

Multi-Label Classification

Multi-label classification involves predicting six labels, as shown in table 1. Each label is a binary task (1 for present and 0 for absent), which has been manually annotated. This task aims to predict annotations for a given review text.

Predicted Class	**Description**
Adverse Drug Reactions (ADR)	-
Withdrawal Symptom (WD)	-
Effective (EF)	-
Ineffective (INF)	-
Sign/Symptom/Illness (SSI)	Text contains explicit SSI as a result of the drug
Drug Indication (DI)	Text contains SSI that is currently being addressed by the drug

Table 1. The six labels predicted by multi-label classifiers

Selected Features Two features were used for multi-label classification. The first is the preprocessed sentences without any additional features. From here on, references to this task-feature pair will be referred as Task_1_Feature_1. The second contains the drug name added to the start of the review text. This will be referred to as Task_1_Feature_2.

Logistic Regression The logistic regression model performed Term Frequency Inverse Document Frequency (TF-IDF) vectorization on the input text. A multi-output classifier was built upon a logistic regression model, which was then tuned using hyperparameter tuning.

Transformer The transformer model uses a pre-trained BERT model, which was trained on a downstream task in order to create a multi-label transformer classifier. Table 2 shows the configurations used in the transformer.

Parameter	Value
hidden_size	768
num_hidden_layers	24
num_attention_heads	12
intermediate_size	3072
num_labels	6
optimizer	AdamW

Table 2. BERT Configuration for Multi-Label Classification

Statistical Tests Multi-label classification will be tested using macro and micro averaged precision, recall, and F1-scores. Accuracy will also be reported. In addition, follwing figure 1, the Hamming Loss (HL) of the predictions are also calculated. HL measures the fraction of labels that are incorrectly predicted, on average, across all samples and is used to evaluate multi-label classification tasks [6].

Statistical test equations are found in Appendix A.

Multi-Class Classification

Multiclass classification aims to take in text and predict the drug that is being reviewed. There are four classes of drugs to predict. Lexapro, cymbalta, effexorxr, and zoloft.

Selected Features Three features were selected for multi-class classification. The first is the preprocessed sentence without any additional features, referred to as Task_2_Feature_1.

The second is the preprocessed text along with its annotations. For the logistic regression model, the binary annotations are simply added onto the end of the sentences as 1s and 0s.

On the other hand, the transformer's input has the review text concatenated with the predicted class names. If the class is labelled 1, a [POS] token was placed in front of it. If the class is labelled 0, a [NEG] token was placed instead. These tokens identify positive(1) and negative(0) labels respectively. This feature will be referred to as Task_2_Feature_2. References to this task-feature pair will be described with the model (e.g., Task_2_Feature_2(Transformer)) when the specifc model feature is of importance. Otherwise, Task_2_Feature_2 will refer to this task-feature pair generally.

The third feature is to only use the annotation inputs. This will be referred to as Task_2_Feature_3. Similar to feature 2, the logistic regression takes in binary labels while the transformer takes in predicted class names with the described tokens above. Similar references to specific or general model will be made for this task-feature pair.

Logistic Regression TF-IDF was also used on the input text. However, in contrast to using a multi-label classifier built on top of a logistic regression model, multi-class classification uses logistic regression directly.

Transformer The transformer model is identical to the multi-label transformer, except it was trained on a multi-class downstream task. The number of hidden layers was also decreased due to long training times (about 1 hour for 1 epoch using 24 layers on a NVIDIA GeForce RTX 3060 Ti). The transformer specifications are shown in table 3.

Parameter	Value
hidden_size	768
num_hidden_layers	8
num_attention_heads	12
intermediate_size	3072
num_labels	1
optimizer	AdamW

Table 3. BERT Configuration for Multi-Class Classification

Statistical Tests Multi-class classification will be tested using macro and micro averaged precision, recall, and F1-scores as described previously. Accuracy will also be reported.

Additionally, following figure 1, Cohen's Kappa (Cohen's K) and Matthews Correlation Coefficient (MCC) will be used. Cohen's K returns a value between 1 and -1, where 1 means a perfect agreement, 0 means no agreement above chance, and -1 indicating less agreement than random chance. Cutoff points for Cohen's K is given in table 4, which is based on McHugh's research [7].

Absolute Cohen's Kappa Range	Interpretation
$\|\kappa\| \leq 0$	No agreement
$0.01 \leq \|\kappa\| \leq 0.20$	None to slight agreement
$0.21 \leq \|\kappa\| \leq 0.40$	Fair agreement
$0.41 \leq \|\kappa\| \leq 0.60$	Moderate agreement
$0.61 \leq \|\kappa\| \leq 0.80$	Substantial agreement
$0.81 \leq \|\kappa\| \leq 1.00$	Almost perfect agreement

Table 4. Cohen's Kappa Cutoff Points

MCC is used to measure the quality of binary classifications [8]. Similar to Cohen's K, MCC returns a value betwen 1 and -1, where 1 means perfect predictions, 0 means a prediction that is no better than random, and -1 means total disagreement between predictions and observations. While the cutoff points of MCC are generally created using a receiver operating characteristic (ROC) curve, this report will follow a basic cutoff point described in table 5.

Absolute MCC Values	Interpretation
$0 \leq \|\text{MCC}\| \leq 0.1$	Random Performance
$0.1 < \|\text{MCC}\| < 0.3$	Weak Correlation
$0.3 \leq \|\text{MCC}\| < 0.5$	Moderate Correlation
$\|\text{MCC}\| \geq 0.7$	Strong Correlation

Table 5. MCC Cutoff Points

Friedman's Test of Significance

In order to investigate whether the predictions are significantly different from each other, Friedman's test will be performed on both tasks. Friedman's test is a hypothesis testing method, where the null hypothesis is that

there is no significant difference between two samples of predictions[9]. Meanwhile, the alternative hypothesis is that a significant difference does exist. This test returns a probability(p)-value, which indicates how likely is it to get result predictions if the compared samples have no significant differences. A cutoff point (α value) of 0.05 will be used, meaning that if the p-value is <0.05, the null hypothesis will be rejected. This test was found to not be suitable for the multi-class classification task, due to the output as text. Additional discussion on the impact of this will be discussed in Discussion and Conclusion.

- **Null Hypothesis (H_0)**: $H_0 : \text{MSE}_1 = \text{MSE}_2$
- **Alternative Hypothesis (H_1)**: $H_1 : \text{MSE}_1 \neq \text{MSE}_2$
- $\alpha = 0.05$

Results

Discussion and Conclusion

References

1. Dror R, Baumer G, Shlomov S, Reichart R. The Hitchhiker's Guide to Testing Statistical Significance in Natural Language Processing. In: Gurevych I, Miyao Y, editors. Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). Melbourne, Australia: Association for Computational Linguistics; 2018. p. 1383-92. Available from: https://aclanthology.org/P18-1128.

2. Wang X, Hripcsak G, Markatou M, Friedman C. Active Computerized Pharmacovigilance Using Natural Language Processing, Statistics, and Electronic Health Records: A Feasibility Study. Journal of the American Medical Informatics Association. 2009 05;16(3):328-37. Available from: https://doi.org/10.1197/jamia.M3028.

3. Rainio O, Tauho J, Klén R. Evaluation metrics and statistical tests for machine learning. Sci Rep. 2024;14(6086).

4. Zolnoori M, Fung K, Patrick T, Fontelo P, Kharrazi H, Faiola A, et al. The PsyTAR Dataset: From Patients Generated Narratives to a Corpus of Adverse Drug Events and Effectiveness of Psychiatric Medications. Data in Brief. 2019 03;24:103838.

5. Zolnoori M, Fung KW, Patrick TB, Fontelo P, Kharrazi H, Faiola A, et al. A systematic approach for developing a corpus of patient reported adverse drug events: A case study for SSRI and SNRI medications. Journal of Biomedical Informatics. 2019;90:103091. Available from: https://www.sciencedirect.com/science/article/pii/S1532046419300012.

6. Díez J, Luaces O, del Coz JJ, Bahamonde A. Optimizing different loss functions in multilabel classifications. Regular Paper. 2015;3:107-18. Available from: https://link.springer.com/article/10.1007/s13748-014-0060-7.

7. McHugh ML. Interrater reliability: the kappa statistic. Biochem Med (Zagreb). 2012;22. Available from: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3900052/#:~:text=Cohen%20suggested%20the%20Kappa%20result,1.00%20as%20almost%20perfect%20agreement.

8. Chicco D, Tötsch N, Jurman G. The Matthews correlation coefficient (MCC) is more reliable than balanced accuracy, bookmaker informedness, and markedness in two-class confusion matrix evaluation. BioData Mining. 2021;13. Available from: https://doi.org/10.1371/journal.pone.0177678.

9. Hoffman JIE. Chapter 26 - Analysis of Variance II. More Complex Forms. In: Hoffman JIE, editor. Biostatistics for Medical and Biomedical Practitioners. Academic Press; 2015. p. 421-47. Available from: https://www.sciencedirect.com/science/article/pii/B9780128023877000263.

Appendix

Appendix A - Equations for Statistical Tests

Cohen's K:

$$\kappa = \frac{p_o - p_e}{1 - p_e} \tag{1}$$

where:

$$p_o = \text{relative observed agreement among raters}$$
$$p_e = \text{hypothetical probability of chance agreement}$$

(p_o) is calculated as:

$$p_o = \frac{a + d}{a + b + c + d} \tag{2}$$

(p_e) is calculated as:

$$p_e = \left(\frac{(a+b)}{a+b+c+d} \cdot \frac{(a+c)}{a+b+c+d}\right) + \left(\frac{(c+d)}{a+b+c+d} \cdot \frac{(b+d)}{a+b+c+d}\right) \tag{3}$$

where:

- a = Number of times both annotators agreed the sample was positive
- d = Number of times both annotators agreed the sample was negative
- b = Number of times Annotator A said positive but Annotator B said negative
- c = Number of times Annotator A said negative but Annotator B said positive

Matthews Correlation Coefficient:

$$\text{MCC} = \frac{TP \cdot TN - FP \cdot FN}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}} \tag{4}$$

Hamming Loss:

$$\text{HL} = \frac{1}{N \times L} \sum_{l=1}^{L} \sum_{i=1}^{N} (Y_{i,l} \neq X_{i,l})$$

where:

N : Number of samples

L : Number of labels

$Y_{i,l}$: True label for ith sample and lth label

$haty_{ij}$: Predicted label for ith sample and lth label

$(Y_{i,l} \neq haty_{ij})$: 1 if $y_{ij} \neq \hat{y}_{ij}$ and 0 otherwise

Accuracy, Precision, Recall, and F1-Scores:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{5}$$

$$Precision = \frac{TP}{TP + FP} \tag{6}$$

$$Recall = \frac{TP}{TP + FN} \tag{7}$$

$$F1 = \frac{2 * Precision * Recall}{Precision + Recall} = \frac{2 * TP}{2 * TP + FP + FN} \tag{8}$$

Friedman's Test:

$$T = \frac{12 \sum S_{ij}^2}{jk(j+1)} - 3k(j+1) \tag{9}$$

where:

T = test statistic

$\sum S_{ij}^2$ = sum of the squared sums of ranks for each prediction set

j = number of prediction sets

k = number of instances