Evaluation Methods in Natural l Processing (NLP): Part-1

Jaimin Mungalpara · Follow
8 min read · Apr 27, 2023

Evaluation in NLP (Natural Language Processing) refers to the process of assessing the quality and performance of NLP models. It involves measuring how well a model is able to complete a specific NLP task, such as text classification, sentiment analysis, machine translation, or question answering. Evaluation is typically performed using metrics that reflect the accuracy or effectiveness of the model. These metrics may vary depending on the task and the specific goals of the evaluation. For example, accuracy, precision, recall, and F1-score are common metrics for evaluating text classification and information retrieval models, while

BLEU and ROUGE are metrics used in machine translation evaluation, Perplexity and WER metrics is used for Automatic speech recognition and text generation.

Let's take a deep dive in accuracy, precision, recall, F1 score and BLEU score in this part.

Accuracy, precision, recall and F1-score

These metrics are especially useful in classification tasks such as sentiment analysis, named entity recognition, and text classification. All these metrics are dependent on TP, TN, FP and FN. TP (True Positive) be the number of correctly identified positive instances, TN (True Negative) be the number of correctly identified negative instances, FP (False Positive) be the number of incorrectly identified positive instances, and FN (False Negative) be the number of incorrectly identified negative instances.

Accuracy: Accuracy is the proportion of correctly classified instances out of the total number of instances. In NLP tasks, accuracy is often used to measure the overall performance of a model.

> *Accuracy = (TP + TN) / (TP + TN + FP + FN)*

Precision: Precision is the proportion of true positives (correctly identified instances) to the total number of instances identified as positive. In NLP tasks, precision is used to measure how many of the instances identified as positive are actually positive.

> *Precision = TP / (TP + FP)*

Recall: Recall is the proportion of true positives to the total number of instances that are actually positive. In NLP tasks, recall is used to measure how many of the positive instances were correctly identified by the model.

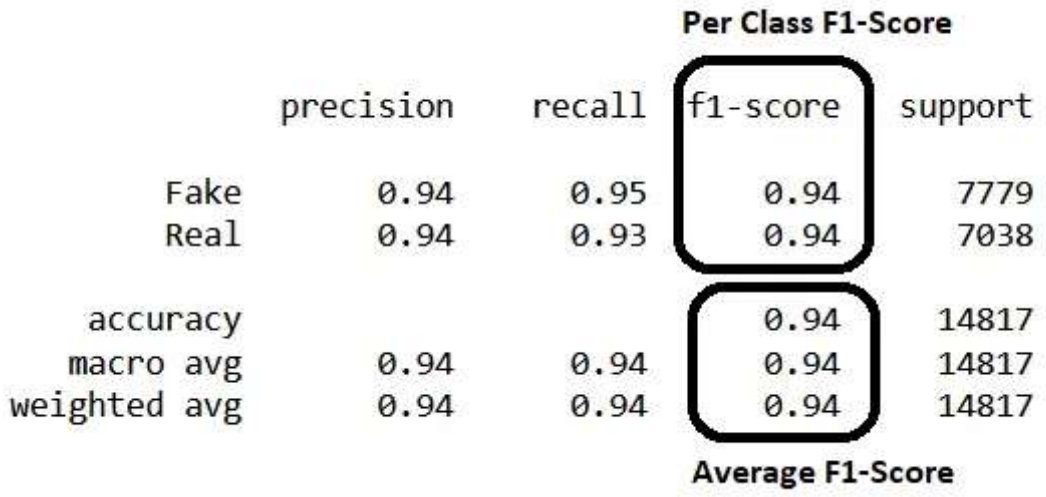
> *Recall = TP / (TP + FN)*

F1-score: F1-score is the harmonic mean of precision and recall, and is used to balance the trade-off between precision and recall. It ranges from 0 to 1, with 1 being the best possible score. In NLP tasks, F1-score is often used to evaluate the overall performance of a model.

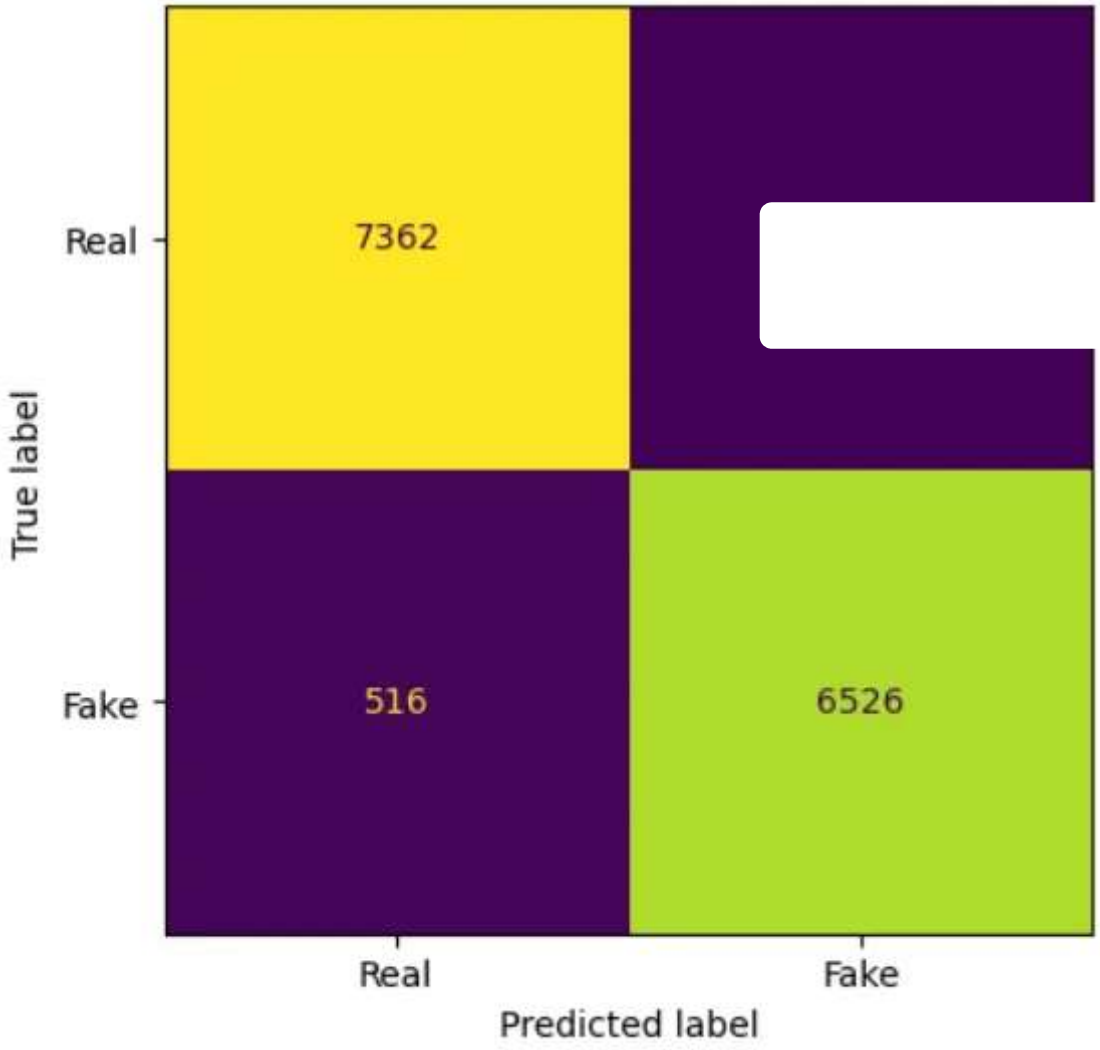
> *F1-score = 2 * (Precision * Recall) / (Precision + Recall)*

Let's implement a text classification model and try to find out the evaluation metrics for the same. For this we are going to use the dataset from here.

While running `sklearn.metrics.classification_report`, we get the following classification report:



We can see the recall, precision and f1-score per class and average. Instead, let us look at the **confusion matrix** for a holistic understanding of all the classes which we have taken as Y label.



The confusion matrix allows us to compute the values of True Positive (**TP**), False Positive (**FP**), and False Negative (**FN**), as shown below. Also, Let's calculate precision, recall and F1-score for each class according to formula mentioned above.

Label	TP	FP	FN	Precision (TP / (TP + FP)	Recall (TP / (TP + FN)	F1-Score (2 * (Precision * Recall) / (Precision + Recall))
Fake	7362	516	413	0.93	0.95	0.94
Real	6526	413	516	0.94	0.93	0.93

Having multiple per-class F1 scores, average of it would be better to check overall performance of a model. However, this is good when we are dealing with class imbalance or Named entity recognition task.

Let's take Macro, Weighted and Micro average of both the classes.

The macro-average is computed using the arithmetic mean of all the per-class F1 scores which treats all classes equally regardless of their support values.

Label	TP	FP	FN	P(	
Fake	7362	516	413	0.94	0.94+0.93 / 2
Real	6526	413	516	0.93	= 0.94

	precision	recall	f1-score	support
Fake	0.93	0.95	0.94	7775
Real	0.94	0.93	0.93	7042
accuracy			0.94	14817
macro avg	0.94	0.94	0.94	14817
weighted avg	0.94	0.94	0.94	14817

The **weighted-averaged** is calculated by taking the mean of all per-class F1 scores with consideration of each class's support. Support mean the no. of sample used per class in calculation of classification metrics.

Label	TP	FP	FN	Per-Class F1-Score	Support	Support %	Weighted Average
Fake	7362	516	413	0.94	7775	0.52	(0.94*0.52)+(0.93*0.48)
Real	6526	413	516	0.93	7042	0.48	= 0.94

	precision	recall	f1-score	support
Fake	0.93	0.95	0.94	7775
Real	0.94	0.93	0.93	7042
accuracy			0.94	14817
macro avg	0.94	0.94	0.94	14817
weighted avg	0.94	0.94	0.94	14817

The Micro average computes a global average by counting the sums of the True Positives (**TP**), False Negatives (**FN**), and False Positives (**FP**).

Label	TP	FP	FN	Micro Average
Fake	7362	516	413	TP / (TP + ((FP+FN)/2))
Real	6526	413	516	138[illegible]
Total	13888	929	929	

	precision	recall	f1-score	support
Fake	0.93	0.95	0.94	7775
Real	0.94	0.93	0.93	7042
accuracy			0.94	14817
macro avg	0.94	0.94	0.94	14817
weighted avg	0.94	0.94	0.94	14817

Here, the question is we are not seeing precision and recall value for accuracy (Micro Average). Micro average is not mentioned here only the accuracy is given because micro-average essentially computes the proportion of correctly classified observations out of all observations. In addition, if we were to do micro-average for precision and recall, we would get the same value.

Label	TP	FP	FN	Micro Average (F1- Score)	Micro Average (Precision)	Micro Average (F1- Score)
Fake	7362	516	413	TP / (TP + ((FP+FN)/2))	TP / (TP + FP)	TP / (TP + FN)
Real	6526	413	516	13888 / (13888+((929+929)/2))	13888 / (13888+929)	13888 / (13888+929)
Total	13888	929	929	0.94	0.94	0.94

Based on this we can conclude that in case of class imbalance we can deal with macro and weighted average and in case of balanced dataset overall accuracy would be fine to evaluate model performance.

BLEU (Bilingual Evaluation Understudy)

The BLEU score was proposed by Kishore Papineni, et al. in their 2002 paper "BLEU: a Method for Automatic Evaluation of Machine Translation".

BLEU (Bilingual Evaluation Understudy) score is a metric used to evaluate the quality of machine-generated text or translations in Natural Language Processing (NLP). It is a statistical measure that calculates the degree of similarity between a machine-generated sentence and one or more reference sentences. The BLEU score ranges from 0 to 1, with 1 indicating perfect similarity between the machine-generated text and the reference text.

BLEU score is also used in other NLP tasks such as text summarization and image captioning. However, it is important to note that the BLEU score is not always an accurate measure of the quality of machine-generated text, as it has certain limitations and can produce misleading results in some cases.

BLEU Score is between 0 and 1. Thus, score of >0.6 is co
achieve. Even humans would rarely achieve a perfect m
closer to 1 is not realistic so it is called a model is over fitting when BLEU score is 1.

Before we get into how BLEU Score we need to understand N-grams and Precision.

N-gram

N-gram is basically a concept used in NLP for regular text processing . It is actually a set of consecutive word is a order.

For instance, in the sentence “This book is informative”, we could have n-grams such as:

- 1-gram (unigram): “This”, “book”, “is”, “informative”
- 2-gram (bigram): “This book”, “book is”, “is informative”
- 3-gram (trigram): “This book is”, “book is informative”
- 4-gram: “This book is informative”

Precision

This metric we have already discussed above. But here it works in this way.

For example, we have:

- **Target Sentence**: This book is informative
- **Predicted Sentence**: That book is informative

Precision = Number of correct predicted words / Number of total predicted words

Precision = 3 / 4

But here we need to handle it in different way. Precision is good when we would have proper class to predict. We need to handle it with the method called Clipped Precision.

Clipped Precision

Let's take an example to understand how it works.

- **Target Sentence 1**: This book is very informative
- **Predicted Sentence**: This book book is is informativ

Now, we will compare each predicted word with target sentences, if words matched we will consider it as correct. We limit the count for each correct word to the maximum number of times that word occurs in the Target Sentence. Which helps to avoid word repetition.

Unigram	Predicted Count	Original Count	Clipped Count
This	1	1	1
book	2	1	1
is	2	1	1
very	0	1	0
informative	1	1	1
Total	6	4	4

Here, the word "book" occurs only once in Target Sentence. So, we have clipped the word "book" even though it occurs twice in the prediction.

Clipped Precision = Clipped number of correct predicted words / Number of total predicted words

Clipped Precision = 4/ 6

Let's calculate BLEU score.

The mathematical representation of BLEU score is.

$$\text{BLEU} = \underbrace{\min\left(1, \exp\left(1 - \frac{\text{reference-length}}{\text{output-length}}\right)\right)}_{\text{brevity penalty}} \underbrace{\left(\prod_{i=1}^{4} precision_i\right)^{1/4}}_{\text{n-gram overlap}}$$

$$precision_i = \frac{\sum_{\text{snt} \in \text{Cand-Corpus}} \sum_{i \in \text{snt}} \min(m^i_{cand}, m^i_{ref})}{w^i_t = \sum_{\text{snt'} \in \text{Cand-Corpus}} \sum_{i' \in \text{snt'}} m^{i'}_{cand}}$$

Where,

- m^i_{cand} is the count of i-gram in candidate matching the reference translation
- m^i_{ref} is the count of i-gram in the reference translation
- w^i_t is the total number of i-grams in candidate translation

Let's take one example and calculate n-gram precision score.

Sentence : The cat is on the mat

Predicted : The cat is on the table.

As per formula mentioned above let's calculate precision

1- gram precision = 5/6

2-gram precision = 4/5

3-gram precision = 3/4

4-gram precision= 2/3

Now, we combine these scores using Geometric Average Precision (N) formula. We can use different values of N using different weights.

$$
\begin{aligned}
Geometric\ Average\ Precision\ (N) &= exp(\sum_{n=1}^{N} w_n\ log\ p_n) \\
&= \prod_{n=1}^{N} \\
&= (p_1, \quad (p_2) \quad (p_3) \quad (p_4)
\end{aligned}
$$

Let's dive in another step which is Brevity Penalty

Brevity Penalty

As we had seen the words like "the" and "cat" the precision is 1/1 which misleads because it encourage model to give higher score to few words. To avoid this, the Brevity Penalty penalizes sentences that are too short.

$$
Brevity\ Penalty = \begin{cases} 1, & \text{if } c > r \\ e^{(1-r/c)}, & \text{if } c <= r \end{cases}
$$

Based on this formula is we predict very few words this Brevity Penalty is small and it can not be larger than 1 even if predicted sentence is larger than target.

Finally, BLEU Score is calculated by multiplying the Brevity Penalty with the Geometric Average of the Precision Scores.

$$
Bleu\ (N) = Brevity\ Penalty \cdot Geometric\ Average\ Precision\ Scores\ (N)
$$

However, there are different implementation on interent for BLEU score but the mathemetics behind this is with below formula.

$$
\begin{aligned}
log\ Bleu &= min(1 - \frac{r}{c}, 0) + \sum_{n=1}^{4} \frac{log\ p_n}{4} \\
&= min(1 - \frac{r}{c}, 0) + \frac{log\ p_1 + log\ p_2 + log\ p_3 + log\ p_4}{4}
\end{aligned}
$$

Finally, BLEU score is calculated mostly with N=4 where geometric average of unigram and bigram are taken into consideration.

Cons :

Despite its popularity, Bleu Score has been criticized for

- BLEU Score does not consider the meaning of words assessments.
- BLEU Score only recognizes exact word matches and does not consider variants of the same word.
- BLEU Score does not distinguish between the importance of words and penalizes incorrect words equally, regardless of their relevance to the sentence.
- BLEU Score does not take into account the order of words, which can result in different sentences with the same words receiving the same (unigram) Bleu Score.

We will take a look on ROUGH score, WER and perplexity in another article.

Suggestions are always welcome.

Reference :

Evaluating models | AutoML Translation Documentation | Google Cloud

Note: AutoML Translation capabilities are offered by both the AutoML API and the Cloud Translation - Advanced API. We...

cloud.google.com

https://machinelearningmastery.com/calculate-bleu-score-for-text-python/

Bleu | Precision | Recall | F1 Score | Accuracy

J

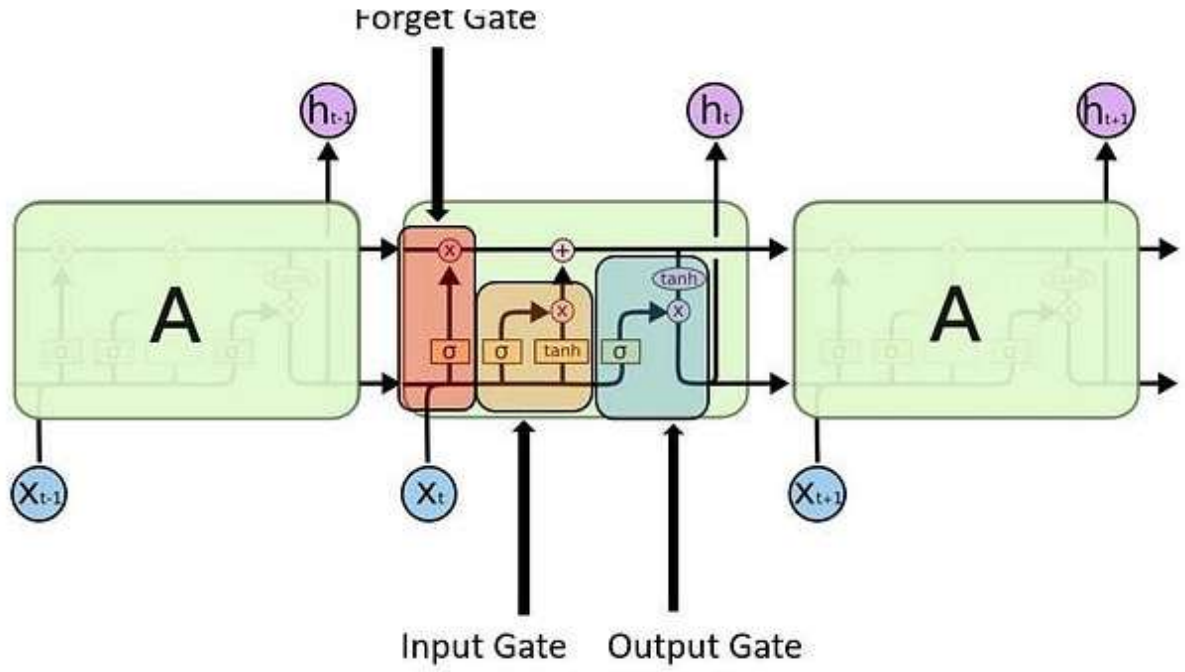
Follow

Written by Jaimin Mungalpara

107 Followers

AI | ML Enthusiastic

More from Jaimin Mungalpara



J Jaimin Mungalpara in Nerd For Tech

What is LSTM , peephole LSTM and GRU?

Long Short Term Memory (LSTM) was introduced by Hochreiter & Schmidhuber (1997) and it was refined by many researchers. LSTM is special...

5 min read · Feb 5, 2021

43

Kia Eisinga in Analytics Vidhya

How to create a Python library

Ever wanted to create a Python library, albeit for your team at work or for some open source project online? In this blog you will learn...

7 min read · Jan 27, 2020

2.2K 27

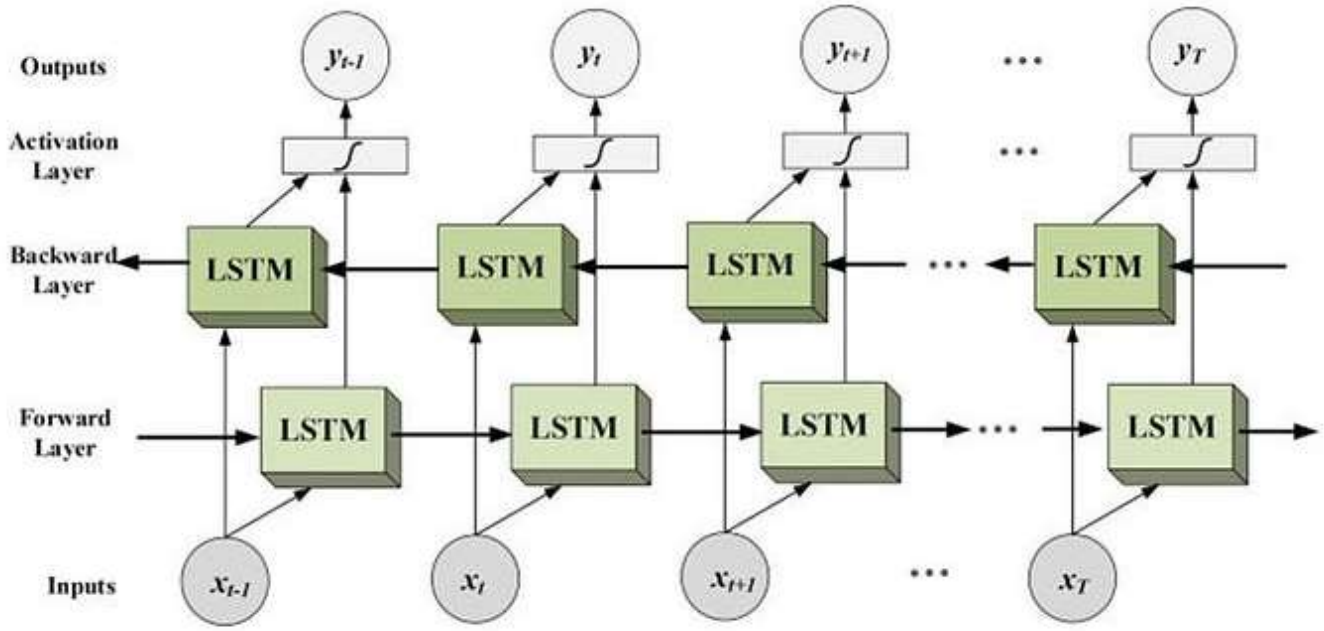
Harikrishnan N B in Analytics Vidhya

Confusion Matrix, Accuracy, Precision, Recall, F1 Score

Binary Classification Metric

6 min read · Dec 10, 2019

890 6



Jaimin Mungalpara in Analytics Vidhya

What does it mean by Bidirectional LSTM?

This has turn the old approach by giving an input from both the direction and by this it can remember the long sequences.

7 min read · Feb 9, 2021

7

See all from Jaimin Mungalpara

Recommended from Medium