# IIMAS NLP TEAM at SemEval-2023 Task 8: AI-TEXTIFICATION

**Manuel Alejandro León Rosas**
National Autonomous
University of Mexico (UNAM)
Applied Mathematics and Systems
Research Institute
alejandroleon12@aragon.unam.mx

**Iván Alejandro Ramos Herrera**
Universidad Autónoma de Aguascalientes
Ingeniería en Computación Inteligente
arhcoder@gmail.com

## Abstract

***SemEval-2023 Task 8*** made available datasets with text samples generated by various sources, spanning both human and machine generation. These data offer a unique opportunity to develop knowledge that leads to the ***creation of tools for texts authorship source detection***. It is in this context that we propouse a solution to two stated objectives:

- **Subtask A:** Detection and classification of a text according to whether it is presumed to be written by a human or by a generative model.

- **Subtask B:** Detection and classification of a text according to its author (human or models *ChatGPT, Davinci, Cohere, Bloomz, Dolly*).

It is then that we explore the implementation of the ***K-Nearest Neighbors*** algorithm to give rise to the classification, in order to train a model that classifies text according to the criteria of *tasks A and B* set out above. Obtaining as results for *task A* the **classification accuracy of 83%** in general domain texts from the *ChatGPT, Cohere, Davinci, Dolly* models vs Human writing, and **58%** for texts from the *Bloomz model* (not used during model training). In addition to accuracy in *Task B* of **52%** for the classification of the five possible classes of authors on the task dataset.

## 1 Introduction

Artificial Intelligence models represent an important advance in the research of *Natural Language computational tools*; these demonstrate abilities such as generating coherent and logical text, or understanding human communication. The creation of *Generative Text Models* **(Gozalo-Brizuela and Garrido-Merchan, 2023)** offers support in tasks such as generating content in text format, and this in turn brings with it the reflection of the importance of having control over the artificially generated content.

In this entire context born of the visible possibility of creating mass content, an interesting need arises to accurately identify when a text is written by a human, or is a product of an *Artificial Generative Model*; such a task could prove too complicated even for the experienced reviewer.

It is then when the same *Natural Language Processing models* shine, allowing, through mathematics and computation, to understand the nature of patterns that could not be very obvious; patterns that exist in the data cloud that is built from our interactions with language. The same type of mathematical-computational models that allow the generation of text with human appearance are those that would allow them to be differentiated from real human texts.

## 2 Background

This article was prompted by the International Workshop on Semantic Evaluation (SemEval) which hosted an open invitation to participate in several different tasks. This is an early version of our submission for Task 8: Multigenerator, Multidomain, and Multilingual Black-Box Machine-Generated Text Detection **(T8:)**.

In this project we focused on Subtask A, Track 1, which is a Binomial Classification problem between machine-generated text and human-generated text; and Subtask B, a Multiclass Classification problem between human-generated text and text generated by one of several GTMs, both of which involve datasets comprising solely English text.

The task organizers provide a train dataset, a test dataset (Dev), Baselines and an evaluation environment with unseen data. At present time, only the train and test datasets have been used.

Dataset sizes can be checked in Table 1.

| | Train | Dev |
|---|---|---|
| Subtask A | 119,757 | 5,000 |
| Subtask B | 71,027 | 3,000 |

Table 1: Dataset Sizes for Subtasks A and B

The dataset entries consist of individual sentences, each represented as a single string, to be classified based on the respective Subtask. The output labels are integers, and their meanings are detailed in Tables 2 and 3.

| label | text source |
|---|---|
| 0 | human |
| 1 | machine |

Table 2: Label Meanings for Subtask A

| label | text source |
|---|---|
| 0 | human |
| 1 | chatGPT |
| 2 | cohere |
| 3 | davinci |
| 4 | bloomz |
| 5 | dolly |

Table 3: Label Meanings for Subtask B

All the datasets for both subtasks are balanced, with an uniform distribution over all classes, however, there are certain critical considerations for the Train and Dev datasets in the Binomial Classification problem, which will be discussed in the following sections.

## 3 Methodology

To bring a solution to the problems raised, we abide by the following methodology coding the solution in ***Python 3.11***:

**PHASE 1 [PREPARATION]:** This phase was limited to accessing the data offered:

**1. Exploration:**
a) Label count to identify possible imbalance.
b) Display of author sources in labels.
c) Selection of relevant attributes (text, label).
d) Cleaning:
- Converting text to lower case.
- Replacing punctuation symbols with words so as not to lose them when tokenizing. The symbols of the text preserved were the following; because as a hypothesis we propose them as important for the conservation of style characteristics in the text: parenthesis, commas, periods, semicolons, quotes, dashes, interrogation symbols, admiration symbols and "&" symbol.
- Removing all except alphabet letter or number.

**2. Separation:** Four different datasets were created with different pre-processing:
**a) Lemma - UNK - GENSIM Embeddings.**
**b) Lemma - UNK - OWN Embeddings.**
**c) UNK - GENSIM Embeddings.**
**d) UNK - OWN Embedings.**
Where:
- **LEMMA:** It was the lemmatization process used through WordNetLematizer from the nltk python library.
- **UNK:** Step in which the frequency of each word in the dataset is counted and those that do not cover a minimum length are replaced by *"xunk"* (with a chosen value of 4).
- **GENSIM EMBEDDINGS:** from Python library *"gensim.models import Word2Vec"*, with the configurations:
- 50 dimension word vectors.
- Window: 5.
- Workers: 4.

**OWN EMBEDDINGS:** Applying the CBOW algorithm to the dataset with a FeedForward Neural Network trained and modeled with:
- h = W1 * X + b1.
- a = ReLU(h).
- z = W2 * a + b2.
- y = Softmax(z).
- Training Epochs: 2000.
- Learning Rate: 0.0001.
- Optimizer: Gradient descent.
- Cost Criterion: Cross Entropy.
- 50 dimension word vectors.

**PHASE 2 [PREPARATION]:** In this phase it was proposed to implement three models:
**1. K-Nearest Neighbors Classifier:** Implemented from the Python library scikit-learn, *from sklearn.neighbors import KNeighborsClassifier* with the following parameters:
- Metric: "cosine".
- With Average Pooling.

- With 1-Grams of Words.

**2. Recurrent Neural Network:** With LSTM architecture:

- model = tf.keras.Sequential() *from tensorflow.keras* module library for *Python*.
- LSTM layer of 100 units.
- Sigmoid Layer with binary unit.
- Loss Criterion: Binary Crossentropy.
- Optimizer: ADAM.
- KFoldsCV with k = 5.

**3. SELF ATTENTIONAL Encoder Model:** With an architecture of the form:

- Input Embedding Layer.
- Positional Encoding Layer.
- Temperature: 10,000.
- Multi-Head Attention Layer.
- Heads: 4.
- Add  Normalization Layer.
- FeedForward Classification Layer.
- Output Function: Softmax.

**IMPORTANT NOTE:** It should be noted that only Model 1 was completely implemented and evaluated. Model 2 was partially evaluated, and Model 3 was not fully implemented. They do not reflect actual results in this work; however, they are included because they were completely implemented even if they could not be trained with complete data sets.

**PHASE 3 [EVALUATION]:** With the following metrics and results. **IMPORTANT NOTE:** For reproducibility, a ***random seed*** with a value of ***11*** was used.

**FOR TASK A:**

**1. KNN-COS-DISTANCE MODEL:** Using as test data samples of "train" dataset (it is because SemEval-2023 Task 8 offers an individual dataset just for testing). It uses the train dataset spliting in 70% train, 30% test.

**For A:** Lemma - UNK - GENSIM Embeddings:
- **Accuracy:** 83%
- **Weighted Average:** 83%
- **F1-Score MACRO:** 82%

**For B:** Lemma - UNK - OWN Embeddings:
- **Accuracy:** 77%
- **Weighted Average:** 77%
- **F1-Score MACRO:** 77%

**For C:** UNK - GENSIM Embeddings:
- **Accuracy:** 80%
- **Weighted Average:** 79%
- **F1-Score MACRO:** 79%

**For D:** UNK - OWN Embedings:
- **Accuracy:** 73%
- **Weighted Average:** 73%
- **F1-Score MACRO:** 73%

**KNN-COS-DISTANCE MODEL:** Using excercise proposed testing dataset:

**For A:** Lemma - UNK - GENSIM Embeddings:
- **Accuracy:** 58%
- **Weighted Average:** 55%
- **F1-Score MACRO:** 55%

**FOR TASK B:**

**1. KNN-COS-DISTANCE MODEL:** Using as test data samples of "train" dataset (it is because SemEval-2023 Task 8 offers an individual dataset just for testing). It uses the train dataset spliting in 70% train, 30% test. **CLASSES = 5.**

**For A:** Lemma - UNK - GENSIM Embeddings:
- **Accuracy:** 52%
- **Weighted Average:** 51%
- **F1-Score MACRO:** 51%

**For B:** Lemma - UNK - OWN Embeddings:
- **Accuracy:** 50%
- **Weighted Average:** 51%
- **F1-Score MACRO:** 50%

**For C:** UNK - GENSIM Embeddings:
- **Accuracy:** 49%
- **Weighted Average:** 49%
- **F1-Score MACRO:** 50%

**For D:** UNK - OWN Embedings:
- **Accuracy:** 46%
- **Weighted Average:** 44%
- **F1-Score MACRO:** 46%

**KNN-COS-DISTANCE MODEL:** Using excercise proposed testing dataset:

**For A:** Lemma - UNK - GENSIM Embeddings:
- **Accuracy:** 51%
- **Weighted Average:** 50%
- **F1-Score MACRO:** 50%

## 4 Conclusions

Most of the current work has been focused in preprocessing of the data into different versions of word vectors, trying out different classifiers, and trying to analyze why the results from an appropriately validated model in the train data, did not generalize to the train dataset for Subtask A. This question has been answered in the FAQ for the task and it is satisfactory to see how the conclusion we reached, how the test data was somehow different

from the train data, held. Data from the test dataset came from an entirely different model, not present within the models in the train data and this represents an additional challenge: generalize results from some language models to all of them.

So far, the main conclusion we obtained from this work is how NLP tasks can benefit a lot from taking into account the posibility of unseen factors within the data and how important it is to include the need for generalizing in the applicability of your solution within your thought proccess, when initially designing it. Within the context of SemEval, these results are rather superfluous, since this insight was the main motivation for the way the organizers picked the evaluation dataset for Subtask A in the first place, but it provides valuable experience for any future projects, and the work done so far represents a solid base for building upon, when attempting to improve the project submission.

## Acknowledgements

We would like to thank Helena and Gemma, for this valuable experience, in addition to sharing their knowledge which allowed us to participate in the first place, and the entire class of IIMAS-LCD, 4th Gen, for their insights in the early stages of this project.

## References

SemEval-2023, Task 8: Multigenerator, multidomain, and multilingual black-box machine-generated text detection. https://github.com/mbzuai-nlp/SemEval2024-task8. Accessed: 2023-11-27.

Roberto Gozalo-Brizuela and Eduardo C. Garrido-Merchan. 2023. Chatgpt is not all you need. a state of the art review of large generative ai models.