Universidad Nacional del Altiplano
Ingeniería Estadística e Informática
Student: Edson Bladimir Pinto Luque
Code: 171198

Homework - N° 002
Data Mining [Técnicas]

LANGUAGE MODELING AND TEXT CLASSIFICATION

1. DEFINITION

   Text classification and language modeling are two related but separate NLP (natural language processing) activities. [1]

   Building a statistical or computational model that can anticipate the next word or phrase in a given piece of text is known as language modeling. The basic purpose of language modeling is to capture the structure and grammatical relationships in the text, and develop a model that can generate coherent and meaningful writing. [2] Large text data sets are used to train language models, which can then be applied to a variety of tasks, including automatic text production, word autocompletion on mobile devices, and machine translation.

   On the other hand, text classification focuses on assigning one or more predefined labels to a given piece of text. The goal is to classify the text into specific categories or classes based on its content or theme. In order to do this, a model must first be trained so that it can recognize patterns and features in text and apply that knowledge to categorize new text. Some popular instances of text categorization are email spam detection, social media sentiment analysis [3] [4], dividing news into distinct groups, and document subject detection.

   In short, language modeling focuses on predicting words or phrases based on the context of the text, while text classification focuses on assigning predefined labels or categories to a text based on its content. Both tasks are crucial in natural language processing and are employed in a wide variety of applications and systems.

2. LIST OF TECHNIQUES

   - Methods for modeling languages
     (a) **N-gram models:** These are based on how frequently a corpus of texts contains word sequences of length N.

     (b) **Statistical-Based Language Models:** They use approaches such as Markov chains, hidden Markov models (HMM) and maximum entropy models to capture the probability of occurrence of words or phrases in a particular context.

(c) **Neural Language Models:** Use neural networks, such as recurrent neural networks (RNN) and convolutional neural networks (CNN), to model the context and relationships between words in a text.

(d) **Transformer-Based Language Models:** They are based on the transformer architecture, which allows capturing long-range relationships between words in a text and has proven to be especially effective in languagemodelling.otable examples include GPT (Generative Pre-trained Transformer) and BERT (Bidirectional Encoder Representations).

- Methods for Text Classification:

(a) **Word Vector Representations:** To better capture meaning and semantic similarity between words, use Word2Vec or GloVe methods to represent words as dense numeric vectors.

(b) **Conventional Classification Models:** These models categorize texts into several groups using supervised learning methods like Naive Bayes, Support Vector Machines (SVM), or Random Forests.

(c) **Neural Networks for Text Classification:** They use neural network architectures, such as RNN, CNN, or attention models, to extract relevant features and patterns from text and classify it into different categories.

(d) **Transfer of Learning:** They take pre-trained models on massive data sets, such GPT or BERT, and modify them using fine-tuning methods for particular text categorization tasks.

3. CODE

The code used is the following

```
import pandas as pd
import numpy as np
import tensorflow as tf
from tensorflow import keras
from tensorflow.keras.preprocessing.text import Tokenizer
from tensorflow.keras.preprocessing.sequence import pad_sequences

# Cargar los datos desde el archivo CSV
data = pd.read_csv('datos.csv', encoding='latin-1')

# Dividir los datos en conjuntos de entrenamiento y prueba
train_data = data.sample(frac=0.8, random_state=42)
test_data = data.drop(train_data.index)

```

```
# Obtener las columnas relevantes para el modelado de lenguaje
    yclasificacin de texto
train_sentences = train_data['OriginalTweet'].values
train_labels = train_data['Sentiment'].values
test_sentences = test_data['OriginalTweet'].values
test_labels = test_data['Sentiment'].values

# Crear los diccionarios de etiquetas para convertir las etiquetasde texto
    a nmeros
label_dict = {'Positive': 0, 'Negative': 1, 'Neutral': 2, 'Extremely
    Positive': 3, 'Extremely Negative': 4}
inverse_label_dict = {v: k for k, v in label_dict.items()}

# Convertir las etiquetas de texto a nmeros
train_labels = np.array([label_dict[label] for label in train_labels])
test_labels = np.array([label_dict[label] for label in test_labels])

# Modelado de lenguaje
tokenizer = Tokenizer()
tokenizer.fit_on_texts(train_sentences)
train_sequences = tokenizer.texts_to_sequences(train_sentences)
test_sequences = tokenizer.texts_to_sequences(test_sentences)
train_padded_sequences = pad_sequences(train_sequences)
test_padded_sequences = pad_sequences(test_sequences,
    maxlen=train_padded_sequences.shape[1])

# Definir modelo de lenguaje
model = keras.Sequential([
    keras.layers.Embedding(len(tokenizer.word_index) + 1, 16),
    keras.layers.GlobalAveragePooling1D(),
    keras.layers.Dense(16, activation='relu'),
    keras.layers.Dense(len(label_dict), activation='softmax')
])
model.compile(optimizer='adam', loss='sparse_categorical_crossentropy',
    metrics=['accuracy'])
model.fit(train_padded_sequences, train_labels, epochs=10)

# Evaluacin en el conjunto de prueba
test_loss, test_accuracy = model.evaluate(test_padded_sequences,
    test_labels)
print("Prdida en el conjunto de prueba:", test_loss)
print("Precisin en el conjunto de prueba:", test_accuracy)

# Clasificacin de texto en el conjunto de prueba
test_predictions = model.predict(test_padded_sequences)
test_predictions_classes = np.argmax(test_predictions, axis=1)

```

```python
# Convertir las etiquetas numricas predichas a texto
inverse_label_func = np.vectorize(lambda x: inverse_label_dict[x])
predicted_labels = inverse_label_func(test_predictions_classes)

# Comparar las etiquetas reales con las predichas
for i in range(len(test_sentences)):
    print("Texto:", test_sentences[i])
    print("Etiqueta real:", test_labels[i])
    print("Etiqueta predicha:", predicted_labels[i])
    print("----------------------------------------")
```

Listing 1: Code For Language Modeling and Text Classification

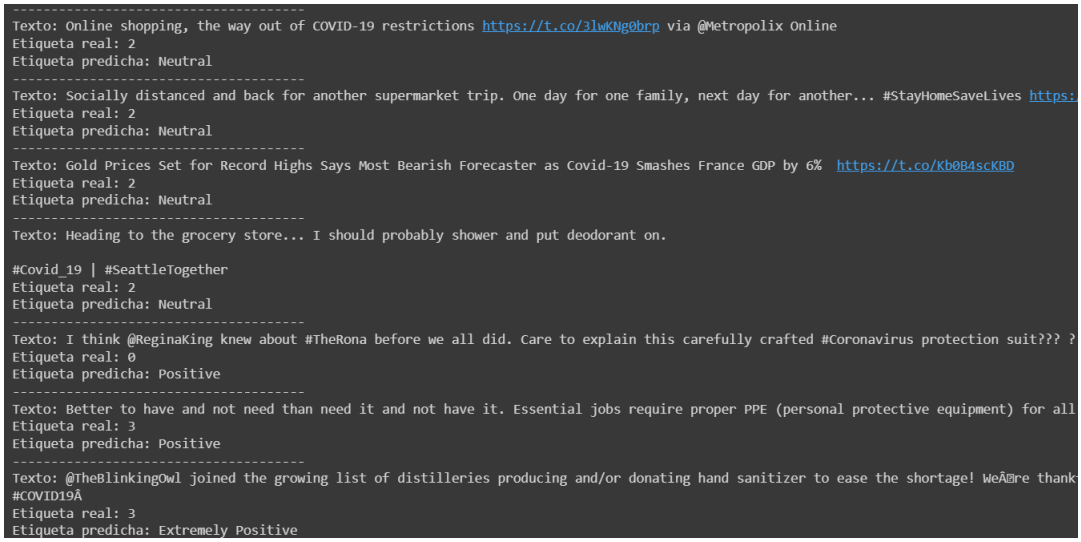

Figure 1: Result for Predicction

4. ARTICLES EXAMPLES

- *"A THREAT MODELING LANGUAGE TO GENERATE ATTACK GRAPHICS OF SUBSTATION AUTOMATION SYSTEMS"*

  The article discusses existing security measures to protect substation automation systems, such as firewalls, user authentication, and data encryption. However, the need to improve security in this area is emphasized, since cyber attacks are constantly evolving and becoming more and more sophisticated. [5], In summary, the article advocates to improve the security of substation automation systems and proposes the use of a specific threat modeling language for this environment, with the aim of identifying and reducing possible vulnerabilities and cybersecurity risks.

- *"CAN AI LANGUAGE MODELS REPLACE HUMAN PARTICIPANTS?"*

The article explores the possibility of using language models as participants in psychological research, particularly in terms of capturing human-like moral judgments. The advantages and limitations of language models in relation to human participants are discussed, and areas in which they can complement research are highlighted. While the models show promise in certain respects, it is concluded that a direct understanding of human minds is still needed and that language models cannot fully replace human participants in psychological science. [2]

- *"TEXT CLASSIFICATION OF ON-BOARD EQUIPMENT FAILURES IN HIGH-SPEED TRAINS BASED ON LABELING-DOC2VEC AND BIGRU"*

  The topic discusses rail vehicle-related fault text categorization. The objective is to use natural language processing techniques to increase the efficiency and accuracy of manual categorization of defect text. They provide a technique to achieve this goal that involves the creation of phrase embeddings using Labeled-Doc2vec, feature extraction using a bidirectional closed recursive unit (BiGRU), and the application of an enhanced attention mechanism (IAtt).The suggested technique may categorize the fault text by taking into consideration the operating condition of the train before and after the fault and studying the semantics of the fault language. When compared to other approaches, the suggested method achieves the highest level of accuracy in fault text classification, lowers labor expenses, and increases fault text classification efficiency. [6] describes a technique for classifying rail vehicle defect messages utilizing Labeled-Doc2vec, BiGRU, and an enhanced attention mechanism. According to the experimental findings, the defect text may be classified with a high degree of efficiency and precision.
- *"TOPICSTRIKER: A TOPIC KERNELS-POWERED APPROACH FOR TEXT CLASSIFICATION"*

  The article describes the application of topic modeling and string kernels, two well-known machine learning and natural language processing techniques. The objective is to provide better text representations for later tasks, including text categorization. While string kernels are mathematical operations on text sequences that enable feature matrices to be constructed for classification, topic modeling is used to uncover latent semantic patterns in vast collections of texts. The method suggested in the text blends supervised string nuclei with unsupervised topic modeling to maximize its benefits. When used in practical applications like text classification, reference data sets are used to validate experimental results. The examination of both methodologies, the investigation of next-generation machine learning strategies, and the introduction of a method that combines topic modeling with string kernels to produce superior text representations are the paper's key contributions. [7]
- *"COMPARISON OF PRE-TRAINED LANGUAGE MODELS IN TERMS OF CARBON EMISSIONS, TIME AND ACCURACY IN MULTI-LABEL TEXT*

CLASSIFICATION USING AUTOML"

Using a 250K data set, this study assessed the outcomes of various language models that had been pretrained for Turkish text categorization.employing a 250K-data set for text categorization. Turkish is a language binder with unique linguistic characteristics, including as repeated phrases, idioms, and metaphors, that make robotic word processing difficult. having unique linguistic characteristics, such as repetitions, idioms, and metaphors, which make robotic word processing difficult.The evaluation of ConvBERTurk mC4 and BERTurk (unshelled, 128K) language models. The outcomes demonstrated that the BERTurk model performed better in terms of accuracy on the data set, with a training duration of 66 minutes and a reasonably low CO2 emission. Additionally, the ConvBERTurk mC4 model produced effective results.produced effective effects.

The study provides further information on Turkish natural language processing capabilities of pretrained language models and their use in text classification.the use of pretrained models to text categorization. [8] This study, which focuses on assessing pretrained language models for Turkish text categorization, establishes the viability of the BERTurk and ConvBERTurk models. The research offers data on word processing performance, training duration, CO2 emissions, and other pertinent parameters in Turkish.

- *"A COMPARATIVE ANALYSIS OF TRANSFORMER BASED MODELS FOR FIGURATIVE LANGUAGE CLASSIFICATION"*

The essay discusses the necessity to create effective techniques for extracting various feelings from significant amounts of text. Researchers evaluate the efficacy of several transformer architecture-based models for figurative language processing. The performance of the transformers in metaphorical language is evaluated using a data set. The study emphasizes how crucial it is for computers to comprehend figurative language, how difficult it is to classify, and how it varies from literal language. Advances in the subject are given, and results from various models are compared. The researchers also investigate how these models generalize to other comparable forms of figurative language after being trained on a particular type. [9] The models based on transformer topologies are used by the researchers to evaluate figurative language processing ability. To demonstrate how transformers function in figurative language and their capacity to generalize across diverse figurative language kinds, the findings from several models are compared and developments in the area are given.

References

[1] S. Bhattacharjee, D. Delen, M. Ghasemaghaei, A. Kumar, and E. W. Ngai, "Business and government applications of text mining natural language processing (nlp) for societal benefit: Introduction to the special issue on "text mining nlp"," *Decision Support Systems*, vol. 162, p. 113867, 11 2022.

[2] D. Dillion, N. Tandon, Y. Gu, and K. Gray, "Can ai language models replace human participants?," *Trends in Cognitive Sciences*, 5 2023.

[3] T. Shaik, X. Tao, C. Dann, H. Xie, Y. Li, and L. Galligan, "Sentiment analysis and opinion mining on educational data: A survey," *Natural Language Processing Journal*, vol. 2, p. 100003, 3 2023.

[4] K. Denecke and D. Reichenpfader, "Sentiment analysis of clinical narratives: A scoping review," *Journal of Biomedical Informatics*, vol. 140, p. 104336, 4 2023.

[5] E. R. Ling and M. Ekstedt, "A threat modeling language for generating attack graphs of substation automation systems," *International Journal of Critical Infrastructure Protection*, vol. 41, p. 100601, 7 2023.

[6] W. Wei and X. Zhao, "Fault text classification of on-board equipment in high-speed railway based on labeled-doc2vec and bigru," *Journal of Rail Transport Planning Management*, vol. 26, p. 100372, 6 2023.

[7] N. V. Chandran, V. S. Anoop, and S. Asharaf, "Topicstriker: A topic kernels-powered approach for text classification," *Results in Engineering*, vol. 17, p. 100949, 3 2023.

[8] P. Savci and B. Das, "Comparison of pre-trained language models in terms of carbon emissions, time and accuracy in multi-label text classification using automl," *Heliyon*, vol. 9, p. e15670, 5 2023.

[9] T. Junaid, D. Sumathi, A. N. Sasikumar, S. Suthir, J. Manikandan, R. Khilar, P. G. Kuppusamy, and M. J. Raju, "A comparative analysis of transformer based models for figurative language classification," *Computers and Electrical Engineering*, vol. 101, p. 108051, 7 2022.