Transformer based Model for Coherence Evaluation of Scientific Abstracts: Second Fine-tuned BERT

Anyelo-Carlos Gutierrez-Choque[1], Vivian Medina-Mamani[2], Eveling Castro-Gutierrez[3],
Rosa Núñez-Pacheco[4], Ignacio Aguaded[5]
Universidad Nacional de San Agustín de Arequipa, Perú[1,2,3,4]
Universidad de Huelva, España.[5]

***Abstract*—Coherence evaluation is a problem related to the area of natural language processing whose complexity lies mainly in the analysis of the semantics and context of the words in the text. Fortunately, the Bidirectional Encoder Representation from Transformers (BERT) architecture can capture the aforementioned variables and represent them as embeddings to perform Fine-tunings. The present study proposes a Second Fine-Tuned model based on BERT to detect inconsistent sentences (coherence evaluation) in scientific abstracts written in English/Spanish. For this purpose, 2 formal methods for the generation of inconsistent abstracts have been proposed: Random Manipulation (RM) and K-means Random Manipulation (KRM). Six experiments were performed; showing that performing Second Fine-Tuned improves the detection of inconsistent sentences with an accuracy of 71%. This happens even if the new retraining data are of different language or different domain. It was also shown that using several methods for generating inconsistent abstracts and mixing them when performing Second Fine-Tuned does not provide better results than using a single technique.**

***Keywords*—Coherence evaluation; inconsistent sentences detection; BERT; second fine-tuned**

I. Introduction

Natural language processing (NLP) is a subarea of artificial intelligence that involves tasks related to the analysis of text information using computational means. These tasks are: text generation, automatic text summarization, speech analysis and information extraction. Textual coherence modeling belongs to this class of tasks described; it consists of distinguishing coherent documents from incoherent ones [1]. Coherence in NLP is very relevant nowadays, because it is implicitly involved in several applications such as speech generation, text summarization generation, translations etc. The models proposed in text generation must ensure that the results are coherent texts. The automatic evaluation of coherence contributes to the generation of these texts with quality.

According to Charolles [2], coherence operates through the thematic progression which implies that all the ideas of a coherent text must be connected to each other. Each sentence provides a piece of information that ensures thematic continuity. A coherent text must also present consistency of ideas; this implies that no idea in a text should contradict another and neither should it be incongruent with the universe of the text to which it belongs.

Coherence also implies the type of informational and semantic connectivity that a text possesses [3]. A text is considered coherent if it is semantically consistent and provides cognitive integrity [4], therefore, a coherent document is easier to understand than an incoherent document. Coherence is more important when analyzing scientific papers, as it must communicate information effectively to reviewers and researchers. Incoherence in scientific writing directly affects both the reading experience and the comprehensibility of scientific papers [27]. Let us consider the sentence-divided scientific papers in Fig. 1 and 2.

In the left column of Fig. 1, the scientific abstract reports on the effects of candy advertising and consumption reactions of certain additives in children under 12 years old, while in the right column, the third sentence reports on project-based learning that expresses an idea different from the other sentences, evidencing the incoherence of the text, due to the fact that the thematic progression and consistency of ideas have not been met. This phenomenon occurs in the same way in the right column of Fig. 2. The abstract deals with machine learning and data representation, while the sixth sentence reports on image processing.

As we have seen, the incoherence that occurs during scientific writing creates difficulties in transmitting and disseminating the authors ideas [27]. This happens because the sentences produced are not strongly interconnected, but are isolated, managing to label the scientific paper as ”poorly written” or ”difficult to follow” [28]. This kind of problem can be intentional or unintentional. Unfortunately, most of the existing systems that check for errors in scientific papers [29] lack advanced features for coherence quality control.

Given the above consideration, identifying incoherent sentences in a scientific paper becomes a problem of high rigor when evaluating scientific abstracts. The abstract is the only part of the article that is usually published in conference proceedings, and that readers usually review when searching through electronic databases; Likewise, it is a section that a potential referee gives a reading to when they are invited by an editor to review a manuscript [5]. It often contains the following structure: context, methodology, results and conclusions, each of which should provide relevant and semantically consistent information.

The evaluation of coherence requires a thorough analysis of the parts of the text at the structural and semantic level, since being a natural language it does not follow a set of rules like formal languages [4]. It is too abstract a concept [6]; however, it is a problem that has been given attention in different studies

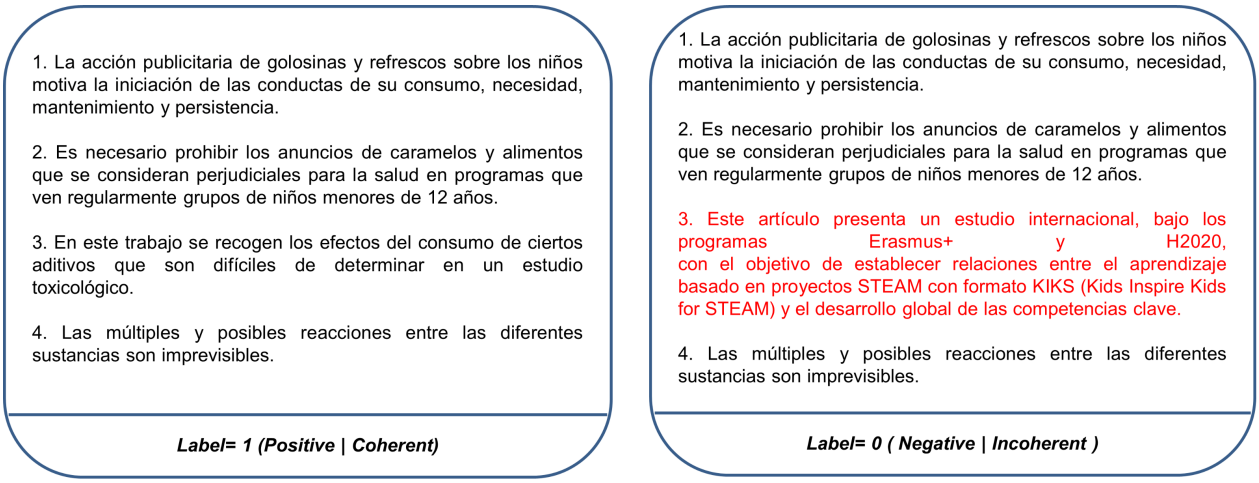

Fig. 1. Coherent and Incoherent Scientific Abstract in Spanish.

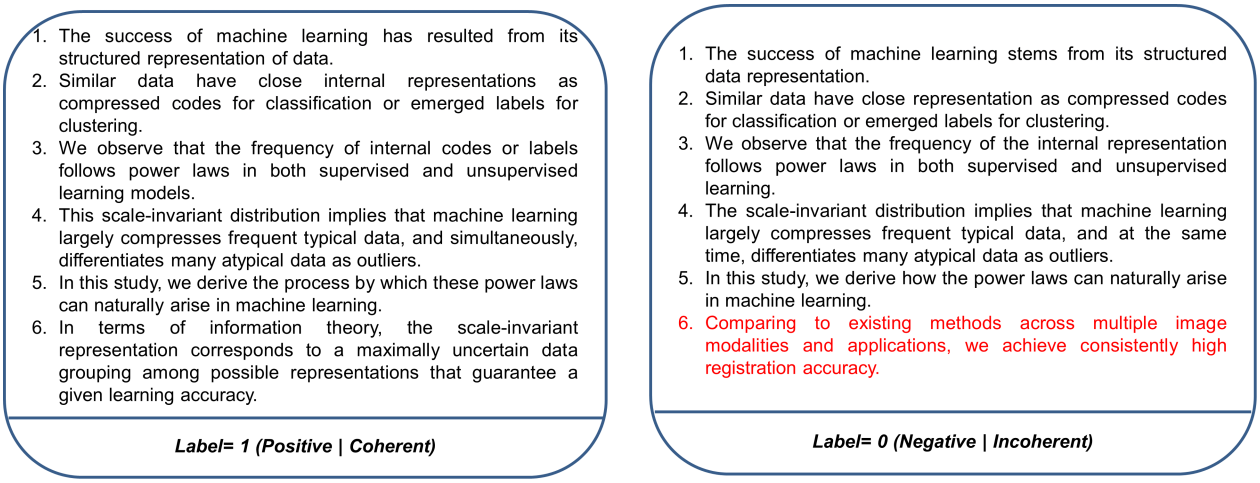

Fig. 2. Coherent and Incoherent Scientific Abstract in English.

and/or solutions since the twentieth century, as mentioned in the following paragraphs.

Foltz in 1998 [7] proposed the first coherence evaluation method using machine. This method is based on the study of latent semantic analysis (LSA), a method that compares units of textual information and determines their semantic relationship. In the following years, several coherence analysis methods were proposed by various researchers; however, no method has proved to be perfect [8].

Relational models such as the ”Rhetorical Structure Theory” [9] define relationships that hierarchically structure texts. Thus, in the work of Barzila and Lapata (2008) a model called Entity Grid [10] was proposed to evaluate local cohesion. This approach was based on the centering theory [11], which models a text as a set of segments and utterances that produce centers of attention.

Unlike the Entity Grid model [10], which is a method for evaluating coherence at the local level, in the work of Guinaudeau and Strube (2013) a graphical model called Entity Graph [12] was proposed to measure text coherence at the global level. This bipartite graph allows relating non-adjacent sentences of a text.

In the work of Li and Hovy (2014) it has been shown that recurrent and recursive neural networks are designed to estimate the coherence of a text [13]. Recurrent neural networks simulate the processing of a text according to a reading process: word by word; while in the recursive neural network the processing is represented through a binary tree.

Li and Jurafsky (2016) developed a discriminative neural model that can distinguish coherent and incoherent text. They also created 2 generative models that produce coherent text, one is based on SEQ2SEQ and the other is a Markovian model. These models capture the latent discourse dependencies of a given text [14]

Based on the foundations of the Entity Graph model [12], a semantic similarity graph model was proposed in the work of Putra and Tokunaga (2017) to address coherence from a cohesion perspective [15]. They argue that the coherence of a text is built by the cohesion between its sentences. This method employs an unsupervised learning approach.

In the work of Baiyun Cui *et al.* (2017) a deep coherence

model (DCM) was proposed making use of a convolutional neural network architecture to capture text coherence [6]. The model captures the interactions between sentences by calculating the similarities of their distributional representations.

In the work of Mesgar and Strube [21], a local coherence model was developed using a unidirectional standardized LSTM architecture to encode the context of an input sequence of words, then the relationships between adjacent sentences were encoded using LSTM. Finally, a vector representing the coherence of the text was produced.

The use of Recurrent or Recursive Neural Networks allows a vector representation of an input sequence. Based on this, a series of dense (linear) layers can be applied to classify whether the input sequence is coherent or incoherent. Thus, in the work of Moon *et al.* [22], the Bidirectional LSTM (BiLSTM) sentence encoder was applied to capture the grammar of each sentence. Given the numerical representations of the sentences, the local coherence model and the global coherence model extract the respective features.

Bao *et al.* [23] used Recurrent Neural Networks (RNN) for the model to semantically represent a text. For this purpose, they used bidirectional closed recurrent units (BiGRU) in conjunction with the pretrained language model Word2Vec to represent this semantics. The results show that a complete analysis of the coherence of a text can be represented, which favors the task of binary text classification.

The creation of a dataset for training, validation and testing is also indispensable for the evaluation of coherence. Because of this, the work of Mohammadi *et al.* [24] proposes different techniques for generating incoherent or negative documents to be added to the coherent documents in order to train a Convolutional Neural Network. Their results indicate that artificially generating incoherent documents does not guarantee "sufficiently incoherent" documents, which negatively influences the accuracy of the model.

As seen, RNN, LSTM, gated recurrent neural network (GRNN) and BiLSTM are some of the sequence models for NLP tasks such as natural language modeling [17]. In 2017, the Google research team presented the Transformer architecture that replaces the complex RNN and CNN (Convolutional Neural Network) architectures because of its better results: parallel training capability with several GPUs and self-attention mechanism, which allows to "remember" the information in the long term [18]. Fig. 3 shows the architecture of the Transformer and the Bidirectional Encoder Representations from Transformers model. In the Transformer architecture, the encoder maps an input sequence of symbols $(x_1, x_2, ..., x_{n-1}, x_n)$ to a sequence of continuous representations $z = (z_1, z_2, ..., z_{n-1}, z_n)$. Given z, the decoder generates a sequence of output symbols $(y_1, y_2, ..., y_{m-1}, y_m)$, considering one element at a time. The general architecture of Transformer comprises the self-attention mechanism, the encoder and decoder, which are fully connected.

BERT is a model of the pretrained open source language introduced in 2018 [19]. It is based on Google's Transformer architecture. Also, it is designed to pre-train text representations in a bidirectional (left-to-right) manner from unlabeled texts [20]. BERT has two pre-trained models: BERT Base and BERT Large. The first model consists of 12 encoders

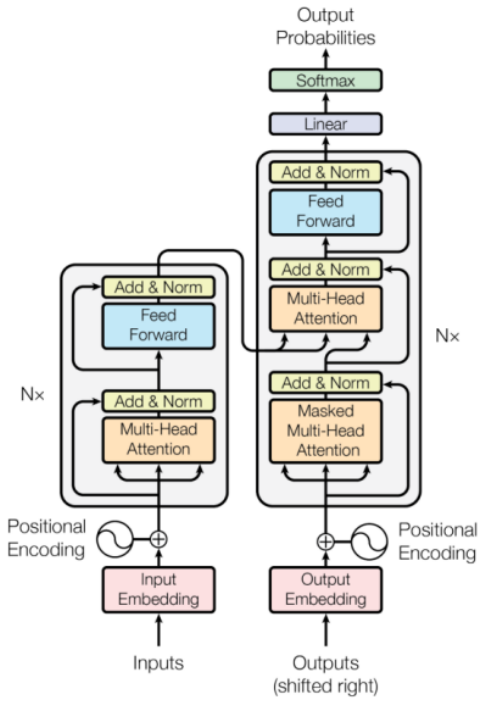

Fig. 3. The Transformer Model Architecture [18].

and a bidirectional self-attention mechanism; while the second model consists of 24 encoders and 16 bidirectional heads. The BERT model is pre-trained with 800 million words from BooksCorpus and unlabeled text from English Wikipedia with 2.5 billion words. This model is suited for small datasets related to specific NLP tasks, for example, the evaluation of coherence in scientific papers.

Fig. 4 shows the neural network architecture of the deep bidirectional BERT and unidirectional (from left to right) OpenAI GPT contextual models [17], in which the unidirectional model generates a representation for each word based on other words in the same sentence. The bidirectional BERT model represents both the preceding and following context in a sentence. However, the context-free models Word2vec and Glove generate a word representation based on each vocabulary word.

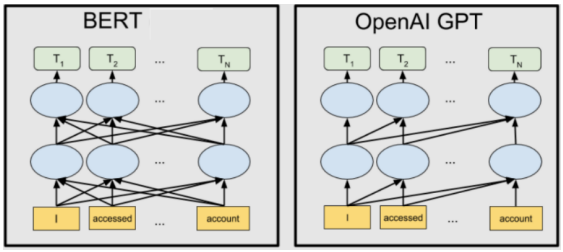

Fig. 4. BERT and OpenAI GPT Neural Network Architecture [18].

According to the historical review, BERT has revolutionized the field of NLP by enabling transfer learning of large language models that can capture complex textual patterns [36]. It also has the advantage of offering better performance and scalability over recurrent neural network architectures

since the latter operate sequentially, while BERT can be parallelized. This research work proposes a Second Fine-Tuned model based on BERT to detect inconsistent sentences to evaluate the coherence in abstracts written in Spanish/English. Two formal techniques for generating incoherent abstracts are also proposed in order to improve the training and validation of the aforementioned model.

This paper is organized as follows: in Section II the most recent and important research on the evaluation of coherence is mentioned. In Section III the methodology and the research proposal are described. Section IV presents the experiment with the results. Finally, Section V presents the discussion, conclusions and future work.

II. Related Work

Discovering semantic progression and consistency of ideas is indispensable for understanding coherence. Previous research has relied on the RNN, LSTM and BiLSTM architecture to evaluate coherence; however, these networks do not use a self-attention mechanism to encode sentences and some information is lost [16]. The Transformer-based architecture allows receiving input sequences in parallel making it more efficient; and, specifically, BERT allows capturing the context of a sentence based on bidirectional analysis [18]. Some work based on BERT to evaluate coherence is shown below.

In the work of Muangkammuen *et al.* (2020), a scoring method based on BERT was proposed to score text clarity using local coherence between adjacent sentences. Cause-effect relationship and contrast were considered [25]. First, a local coherence model was trained according to BERT; then the model was retrained to evaluate the clarity of a text. The results show that retraining provides positive results even if the data on which both trainings were performed were not domain related.

In the work of Callan and Foster (2021), a corpus of narrative stories automatically generated by the pre-trained Transformer GPT-NEO model was proposed, which were analyzed by humans and by 2 automated metrics: BERT Score y BERT NSP. This was done in order to evaluate the coherence and level of interest of narrative texts. The results show that greater emphasis should be placed on BERT evaluation techniques and that generative models do not always produce coherent texts; the more natural and coherent the generated text is, the higher its quality [26].

In the work of *et al.* (2021), a comparative analysis of 3 different types of models for the evaluation of coherence in Polish documents has been developed. The first one is based on Semantic Similarity Graph (SSG); the second one is based on Long Short Term Memory (LSTM); and the third one is based on BERT. The results show that the neural network related methods offer better accuracy than the SSG related methods; and within the neural networks, although the LSTM based method shows better accuracy compared to the BERT based method, it is emphasized that the latter can increase the value of said metric with an additional Fine-Tuned [4].

In the work of Nguyen and Zaslavskiy (2021), a Fine-Tuned method based on the BERT model in conjunction with a clustering algorithm was proposed for the detection of discordant sentences in a corpus of scientific documents written in English and Russian, in order to detect incoherence in scientific writing. Primero generaron ejemplos negativos mediante la métrica BERT Score para calcular la similitud semántica entre pares de oraciones. They first generated negative examples using the BERT Score metric to compute semantic similarity between sentence pairs. Then they trained a model with coherent and incoherent sentence pairs. Finally, they retrained this model with whole paragraph training. The results were positive [27].

In the work of Bendevski *et al.* (2021), a comparative analysis of different artificial intelligence methods for predicting the coherence score of narrative documents was proposed, where it was evidenced that BERT produces better results compared to traditional machine learning methods such as: Linear Regression, Support Vector Machine, Random Forest. They establish dimensions for coherence which are: Context, Chronology and Theme, each of these dimensions possess narrative texts with a coherence score of 0-3 (4-class classification) [30].

According to the work of Noji and Takamura [31], negative examples contribute to a neural language model's ability to robustly handle complex syntactic constructs and improve its robustness.

III. Proposed Model and Methodology

The principal objective of this study is to build a Second Fine-Tuned model based on BERT to detect inconsistent sentences of abstracts in English/Spanish (coherence evaluation). Two negative example generation techniques have been used for this coherent scientific abstracts are positive examples ($label = 1$), while incoherent scientific abstracts are negative examples ($label = 0$).

A. Data Recollection and Preprocessing

First of all, a program has been developed in Python with the beautifulsoup4 library to perform web scraping to the website of the journal "Comunicar" [32]. From this journal 1,493 scientific abstracts written in Spanish were extracted. Second, the corpus of "Medical Semantic Indexing in Spanish" (MESINESP) was downloaded [33], from which 51,390 scientific abstracts written in Spanish were collected. Thirdly, a corpus of arXiv was downloaded through Kaggle [34], from which 56,181 scientific documents written in English were collected.

In addition, a corpus of 448 scientific abstracts from the "International Conference on Machine Learning and Applications" (ICMLA) were added to this corpus. [35]. As a result, a corpus of 56,629 scientific documents was constructed, 3 corpus were collected, 2 in Spanish and 1 in English. It should be added that the downloaded corpus were subjected to 2 preprocessing stages: Remove blanks and Remove duplicates. Two formal techniques were developed for the generation of negative examples (incoherent abstracts):

1) Random Manipulation (RM): In the first method, every abstract T_i of a corpus D is tokenized in N sentences. This $T_i$is represented as a set of sentences $T_i = \{S_1, S_2, S_3, ..., S_{N1}, S_N\}$. Every S_j is a sentence where j is

the position of the sentence in T_j, it must be fulfilled that $1 \leq j \leq N$. The variable i represents the position of a scientific abstract in the corpus D, it must be fulfilled that $1 \leq i \leq size(D)$.Then a S_j is randomly selected, knowing that: $S_j \in T_i$, $T_i \in D$. Therefore, this S_j is replaced with a $S_{j'}$; knowing that: $S_{j'} \in T_{i'}$, $T_{i'} \in D$, and $S_j \neq S_{j'}$.

2) Manipulation Random based K-means (KRM): In the second method, embeddings with BERT are generated from all abstracts of a corpus D. Then by clustering with K-Means ($\#clusters : K = 10$) As a partial result, there are 10 clusters of scientific abstracts labeled as C_i, being i the cluster number in the corpus D. These clusters are grouped by similar embeddings. The K-means algorithm minimizes the principle of inertia, according to the following equation 1:

$$\sum_{i=0}^{n} \min_{\mu_j \in C} (\| x_i - \mu_j \|^2) \quad (1)$$

This measure 1 indicates the internal coherence between the samples belonging to the different clusters. Taking as a reference the variables of the first method; to generate negative examples a $S_k \in T_j$is randomly chosen, considering that: $T_j \in C_i$ y, $C_i \in D$. LThen this S_k is randomly replaced with a $S_{k'}$ knowing that: $S_{k'} \in T_{j'}$, $T_{j'} \in C_{i'}$ y $C_{i'} \in D$. It must be fulfilled that: $S_k \neq S_{k'}$ y $C_i \neq C_j$. Having clusters whose similar scientific documents, knowing that the groups among themselves are different, it is ensured that negative examples are explicitly more incoherent than the first RM method. Finally, the corpus are summarized in the following table I.

TABLE I. CORPUS SUMMARY

Features	Comunicar Corpus		MESINESP Corpus	arXiv + ICMLA Corpus
Language	Spanish		Spanish	English
Coherent abstracts	1,493		50,047	56,181
Incoherent abstracts	RM Method	KRM Method	RM Method	RM Method
	1,491	1,491	49,953	41,559
Total abstracts	2,984	2,984	100,000	97,740

B. Proposed BERT Model

A Second Fine-Tuned model based on BERT is proposed to detect inconsistent sentences in scientific abstracts written in English/Spanish (coherence evaluation), six different experiments have been carried out for this purpose, the same ones as detailed below:

1) First Fine-Tuned to the original pre-trained model of BERT with the Spanish-language dataset of the journal "Comunicar". The dataset has been divided into three segments: training, validation and testing. The technique used to generate negative examples was RM.
2) Second Fine-Tuned to the model trained in experiment 1 by mixing the MESINESP Spanish dataset and "Comunicar" dataset, the same testing set as experiment 1 has been maintained. The technique used to generate negative examples was RM.
3) Second Fine-Tuned to the model trained in experiment 1 by mixing the dataset in English of "arXiv + ICMLA" with that of "Communicate". The same testing segment has been maintained as experiment 1. The technique used to generate negative examples was RM.
4) First Fine-Tuned to the original pre-trained model of BERT with the Spanish-language dataset of the journal "Comunicar". The dataset has been divided into 2 segments: training and validation. The same testing segment has been maintained as experiment 1. The technique used to generate negative examples was KRM.
5) Second Fine-Tuned to the model trained in experiment 4. The same data segments of experiment 2 and also the same testing segment of experiment 1 have been maintained.
6) Second Fine-Tuned to the model trained in experiment 4. The same data segments of experiment 3 and also the same testing segment of experiment 1 have been maintained.

The purpose of the experiments described above is to determine which model is best for evaluating coherence in scientific abstracts written in English/Spanish. Once the positive and negative examples (datasets) have been generated, these experiments have followed the framework proposed in Fig. 5

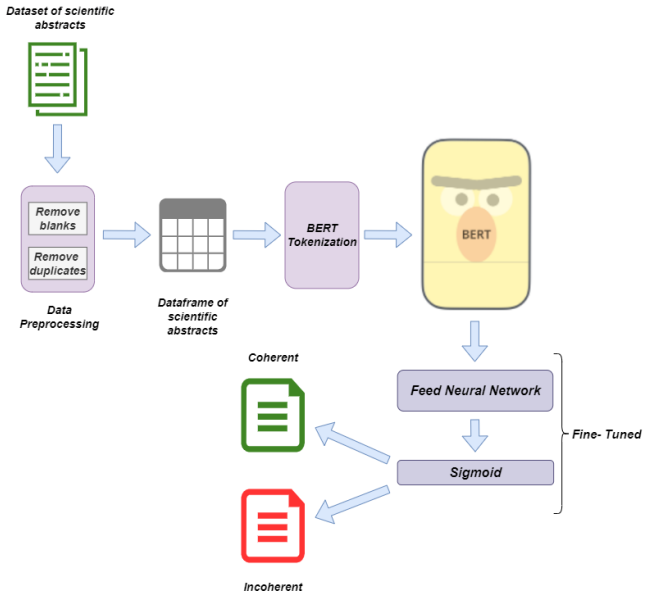

Fig. 5. BERT-Large Coherence Evaluation Framework.

During the tokenization process, an input sequence must be preprocessed to produce tokens (prayer units). In this process, the CASED preprocessor BERT was used, which is a multilingual preprocessor that can process special characters or capital letters/minuscule. A sorting token [CLS] is always included at the beginning of a sentence, each word being a token. To separate one sentence from another, the separation token [SEP] is included.[37]. It is known that an abstract T_i is represented as $T_i = \{S_1, S_2, S_3, ..., S_{N1}, S_N\}$, applying the preprocessing you get: $\{S'_1, S'_2, ...\}$.

The BERT Large pre-trained model was used to perform the numerical coding process of input sequences. This is a more complete version with 24 encoders and 340 million parameters [37]. It was ensured that this model used is multilingual and also that it is CASED. For each token 3 representations of embeddings were applied, the first is called token embeddings (h_{s1}), is responsible for representing a token as a numerical vector. The second is called segment embeddings (h_{s2}), this indicates to which segment a token embeddings belongs. It is known that a segment is that delimited by the [SEP] separator of another segment. The third is called positional embeddings (h_{s3}), this indicates the relative position of a token embeddings in the sentence. Each word is processed simultaneously [37]. Finally, each numerical representation is added to produce a single resulting vector h_c that will be used to train the Fine-Tuned model. The vector h_c can be represented by the following equation:

$$h_c = [h_{s1}, h_{s2}, h_{s3}] \tag{2}$$

Also, Fig. 6 represents how these embedding layers work with a pre-processed input. Fig. 7 shows the general architecture of the BERT Large model.

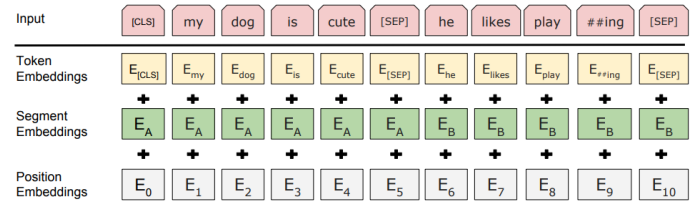

Fig. 6. BERT Input Representation [37].

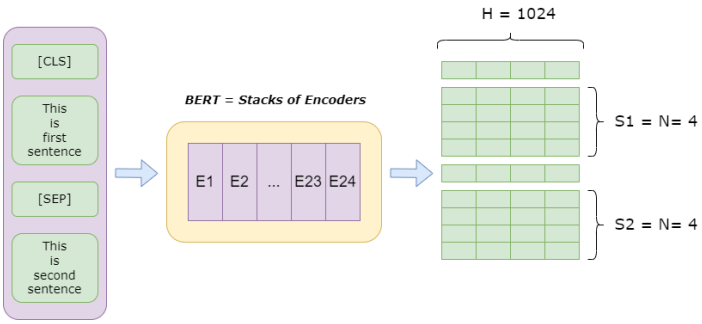

Fig. 7. Architecture BERT LARGE Encoders.

To perform the Fine-Tuned of the pre-trained model, following the six experiments described above, the dataset of the journal "Comunicar" for testing has been divided by 10%. Then 10% was divided for validation and 80% for training. Each experiment has the same percentage of data division. It should be remembered that each experiment was tested with the testing data of the journal "Comunicar". After defining the data sets, a Neural Network Layer Dense has been built with a Rectified Linear Unit activation function (ReLU). This activation function will generate a positive or zero output (if the input is negative). This function is optimal to accelerate the training of deep neural networks, it is defined with the following equation 3:

$$f(x) = max(0, x), x = input \tag{3}$$

After the ReLU layer, a Dropout layer was added to avoid overfitting the model during training; as several studies have shown that it reduces this overfitting and improves the performance of deep neural networks for tasks such as document classification [38]. After implementing the Neural Network layer, a simple layer has been added with a sigmoid activation function. This function is commonly used for binary classifications. It has a prediction range of $[0-1]$, with those closest to zero being those incoherent abstracts (rounded to 0) and those closest to 1 being the coherent ones (rounded to 1). The mathematical representation of this function is shown in the following equation 4:

$$S(x) = \frac{1}{1+e^{-x}}, x = input \tag{4}$$

The latter simple layer contains a single neuron, whose output represents the probability of coherence of a group of abstract sentences. It can be mathematically defined by the following equations 5 and 6. Each experiment has followed the same procedure so far mentioned. Considering the above, the 6 experiments possess the same architecture. This architecture is observed in the following Fig. 8.

$$q_c = f(W_{sen}^T h_c + b_{sen}) \tag{5}$$

$$output = sigmoid(U^T q_c + b) \tag{6}$$

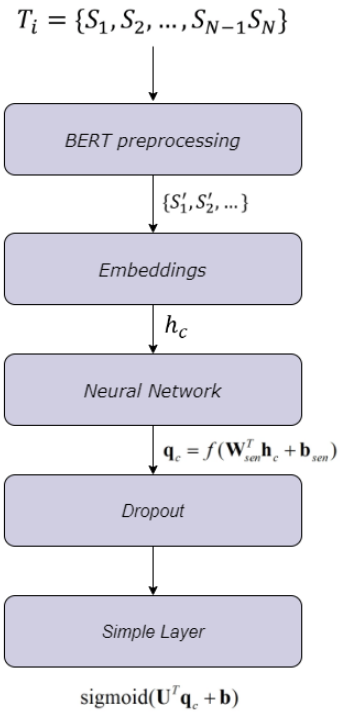

Fig. 8. Based BERT Model Architecture.

IV. Experimental Settings and Results

In this section, the 6 experiments mentioned in the previous section are in depth. Six models with exact architecture have been generated to Fig. 8. The only difference between the experiments are the positive and negative data sets with which their models have been trained. The following subsections are detailed: The Datasets, Experimental Environments, Parameters Fine-tuning and Performance measurements:

A. The Datasets

According to Table I and described in previous sections; each experiment was assigned training, validation and testing data segments. This is shown in Table II. These datasets feed each model to perform the corresponding Fine-Tuned. It should be mentioned that each dataset generated for the experiments has passed through a preprocessing stage explained in Section III.

TABLE II. Prepared Dataset

	Training (80%)	Validation (10%)	Testing (10%)	Dataset	Method
Experiment 1	2,416	269	299	Comunicar	RM
Experiment 2	90, 000 + 2,416	10,000 + 269	299	MESINESP + Comunicar	RM
Experiment 3	87,966 + 2,416	9774 + 269	299	arXiv + ICMLA + Comunicar	RM
Experiment 4	2416	269	299	Comunicar	KRM
Experiment 5	90, 000 + 2,416	10,000 + 269	299	MESINESP + Comunicar	RM + KRM
Experiment 6	87,966 + 2,416	9774 + 269	299	arXiv + ICMLA + Comunicar	RM + KRM

According to Table II, 2 formal methods of generating negative examples have been applied as explained in Section III. Experiment 1 uses only the dataset of the journal "Comunicar". Experiments 2 and 3 depend on experiment 1, in that sense; experiment 2 first uses the dataset of MESINESP and, secondly, the data of the journal "Comunicar" is added (repeating exactly the same training and validation data). Experiment 3 follows the same flow of experiment 2, only that it uses its own dataset as detailed in Table II. Incoherent abstracts of experiments 1,2 and 3 were generated with the RM method.

Experiment 4 is similar to experiment 1 with the difference that its incoherent abstracts were generated by the KRM method. Experiments 5 and 6 depend on experiment 4. In that sense, experiment 5 is similar to experiment 2, with the difference that when performing the training, the techniques of RM and KRM are combined to create a varied dataset. The RM method was applied to the MESINESP dataset and the KRM to the "Communicar" dataset. Experiment 6 follows the same flow as experiment 5 with the difference that uses its own dataset as detailed in Table II.

In summary, the incoherent abstracts of experiment 4 were generated with the KRM method, therefore, in experiments 5 and 6 they ended up using both KRM and RM methods. It should be remembered that the testing data was generated with the RM method. This set belongs only to the magazine "Comunicar", repeating in each of the experiments without exception.

B. Experimental Environments

Experiments 1 and 4 were executed in the Google Colab environment. This environment serves to create automatic machine learning models for free and with powerful hardware resources such as: Graphic Processing Unit (GPU) and Tension Processing Unit (TPU) [39]. On the other hand, experiments 2, 3, 5 and 6 were executed on a standard server. Resources used for experiments are described in Table III.

TABLE III. Experimental Components and Environments

Components	Details	
Dataset prepared	Libraries: Pandas, Beautiful Soup 4	
Preprocessador	bert_multicased_preprocess of Tensorflow.	
Model	bert_multi_cased_L-12_H-768_A-12 of Tensorflow.	
Language Programming & Tools	Python 3.9.6, Jupyter Notebook, Colab Notebook.	
Libraries & Frameworks	Tensorflow, Keras, Numpy, Pandas, Scikit-learn, Matplotlib, Seaborn, Nltk.	
Server	**Environment 1 (Experiments 1,4)**	**Environment 2 (Experiments 2,3,5,6)**
	Google Colab Cloud, GPU Tesla K80 11 GB Memory.	Ubuntu 64-bit S.O., Intel(R) Xeon(R) Gold 5115, CPU @ 2.40GHz, GPU Quadro P5000 16 GB Memory.

C. Parameters Fine-Tuning

The parameters, hyper-parameters and other configurations of the 6 experiments are detailed in the following Table IV.

TABLE IV. Parameters Settings

Name	Parameters and Hiper-Parameters	Value
Experiments 1, 2, 3, 4, 5, 6	- Model	- bert_multi_cased
	- Platform	- Tensorflow
	- Activation Function	- ReLU and Sigmoid
	- Dropout rate	- 0.1
	- Class_weight	- Balanced
	- Callback Model CheckPoint	- Max val_accuracy
	- Optimizer	- Adam
	- Learning rate	- 1e-05
	- Loss Function	- Binary Cross Entropy
	- Epochs	- 5
	- BatchSize	- 64

D. Performance Measures

The basic components that have been used for the evaluation of the models developed during the six experiments are the following:

- True Positive (TP): When the model correctly predicts the positive class. This means it correctly predicts a coherent abstract.
- True Negative (TN): When the model correctly predicts the negative class. This means it correctly predicts an incoherent abstract.
- False Positive (FP): When the model incorrectly predicts the positive class. This means it predicts an incoherent abstract as coherent.
- False Negative (FN): When the model incorrectly predicts the negative class. This means it predicts a coherent abstract as incoherent.

The basic components of the 6 experiments are detailed in the Table V:

TABLE V. EVALUATION COMPONENTS

Models	True Positive (TP)	False Positive (FP)	True Negative (TN)	False Negative (FN)
Experiment 1	118	70	79	32
Experiment 2	124	60	89	26
Experiment 3	114	55	94	36
Experiment 4	111	62	87	39
Experiment 5	137	83	66	13
Experiment 6	127	78	71	23

The Accuracy, F1-score, precision and recall were the most frequently used metrics to report model performance on benchmark datasets. As metrics for binary classification problems, they can be derived from a confusion matrix, a two by two contingency table of the predicted and observed class labels [40]. Once the evaluation components have been defined, performance measurements were calculated using the following equations: 7 8 9 y 10:

$$Precision = \frac{TP}{TP + FP} \tag{7}$$

$$Recall = \frac{TP}{TP + FN} \tag{8}$$

$$F1Score = 2 * \frac{Precision * Recall}{Precision + Recall} \tag{9}$$

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \tag{10}$$

Applying performance measurements equations to the 6 generated models, the following results were obtained from the Table VI.

TABLE VI. MODELS PERFORMANCE MEASURES

Models	Accuracy	Precision	Recall	F1-Score	Loss
Experiment 1	0.65	0.63	0.79	0.70	0.66
Experiment 2	**0.71**	0.67	0.83	0.74	0.55
Experiment 3	0.70	0.67	0.76	0.71	0.59
Experiment 4	0.66	0.64	0.74	0.69	0.62
Experiment 5	**0.68**	0.62	0.91	0.74	0.61
Experiment 6	0.68	0.62	0.84	0.71	0.64

The experiment 2 model offers a better Accuracy (0.71) to detect inconsistent sentences in scientific abstracts. This shows that performing a Second Fine-Tuned mixing data from the same language, but different domain (MESINESP + Comunicar), improves the Accuracy to evaluate coherence.

Experiment 3 shows that performing a Second Fine-Tuned by mixing data from a different language (English) and different domain (arXiv + ICMLA + Communicar) improves accuracy to detect inconsistent sentences (0.70). This aspect is important, as it is shown that you can increase training data from different languages without worrying about results to evaluate coherence. This is because a multilingual pre-trained model of BERT has been used.

In experiments 4, 5 and 6 it is shown that the KRM method works slightly better than the RM method with few data. The KRM method, at first, better classifies incoherent abstracts, but if a large set of data is increased whose negative examples were generated with RM, it does not offer higher performance. This happens because the testing data were not created with the KRM method but with the RM method, which negatively impacts predictions.

The authors agree with the work of Bendevski *et al.* [30] when he mentions that BERT offers better results for the evaluation of coherence than traditional machine learning methods, since the latter cannot understand the context of a text as does BERT, in addition to that BERT offers greater scalability and guarantees the Transfer learning process.

In these studies [4], [6] and [23] they have focused on generating incoherent examples varying the order of sentences (Sentence Ordering Task) unlike the present research that has focused on the detection of inconsistent sentences for the evaluation of coherence in scientific abstracts written in English/Spanish.

V. CONCLUSION AND FUTURE WORK

In this study, it has been shown that abstracts written in different languages/domains can be trained to detect inconsistent sentences of test data whose language and domain is also different from training data. Experiment 2 has proved to be better for the detection of inconsistent sentences of abstracts written in Spanish. Experiment 3 has proved to be more optimal for the evaluation of abstracts written in Spanish using combined training and validation data written in Spanish and English. Also, the variety of incoherent abstracts generated with RM and KRM during the Second Fine-Tuned has proven to deliver no better results than to train with a single method of generating incoherent abstracts when you want to detect inconsistent sentences for coherence evaluation.

Future research will renew the current clustering method (KRM) to a BERT Score-based method for the detection of inconsistent sentences. It will also address the evaluation of coherence as a multi-classification problem taking into account the types of incoherence: contradiction, redundancy and thematic discontinuity.

ACKNOWLEDGMENT

To the Universidad Nacional de San Agustín de Arequipa for the funding granted to the project "Transmedia, Gamification and Video games to promote scientific writing in Engineering students", under Contract No. IBA-IB-38-2020-UNSA. We would like to thank to the "Research Center, Transfer of Technologies and Software Development R+D+i" – CiTeSoft-EC-0003-2017-UNSA, for their collaboration in the use of their equipment and facilities, for the development of this research work.

REFERENCES

[1] Pishdad L., Fancellu F., Zhang R., Fazly A, "How coherent are neural models of coherence?", In Proceedings of the 28th International Conference on Computational Linguistics, Barcelona, Spain, pp. 6126–6138, December, 2020.

[2] Charolles M, "Introduction aux problèmes de la cohérence des textes: Approche théorique et étude des pratiques pédagogiques", Langue française, 1978.

[3] Xiong H., He Z., Wu H., Wang H, "Modeling coherence for discourse neural machine translation", In Proceedings of the AAAI Conference on Artificial Intelligence, Honolulu, HI, USA, Vol. 33, pp. 7338–7345, July, 2019.

[4] Telenyk S., Pogorilyy S., Kramov A, "Evaluation of the Coherence of Polish Texts Using Neural Network Models", Appl. Sci, April, 2021.

[5] Andrade C, "How to write a good abstract for a scientific paper or conference presentation", Indian J Psychiatry, April, 2011.

[6] Baiyun Cui, Yingming Li, Yaqing Zhang, and Zhongfei Zhang, "Text Coherence Analysis Based on Deep Neural Network", Proceedings of the 2017 ACM on Conference on Information and Knowledge Management. 10.1145/3132847.3133047. October, 2017.

[7] Peter W. Foltz, Walter Kintsch and Thomas K Landauer, "The measurement of textual coherence with latent semantic analysis", Discourse Processes, Vol. 25, 1998.

[8] Md. Anwar Hussen Wadud and Md. Rashadul Hasan Rakib, "Text Coherence Analysis based on Misspelling Oblivious Word Embeddings and Deep Neural Network", International Journal of Advanced Computer Science and Applications(IJACSA), 2021.

[9] William C. Mann and Sandra A. Thompson, "Rhetorical structure theory: Toward a functional theory of text organization", Text - Interdisciplinary Journal for the Study of Discourse, Vol. 8, 1988.

[10] Regina Barzilay and Mirella Lapata, "Modeling local coherence: An entity-based approach", Association for Computational Linguistics, Vol. 34, 2008.

[11] Barbara J. Grosz, Scott Weinstein and Aravind K Joshi, "Centering: A framework for modeling the local coherence of discourse", Association for Computational Linguistics, Vol. 21, 1995.

[12] Guinaudeau C., Strube M., "Graph-based Local Coherence Modeling", In Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics, Sofia, Bulgaria, Vol. 1, pp. 93-103, August, 2013.

[13] Li J., Hovy E, "A Model of Coherence Based on Distributed Sentence Representation", In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Doha, Qatar, pp. 25–29, October, 2014.

[14] J. Li and D. Jurafsky, "Neural net models for open-domain discourse coherence," arXiv preprint arXiv:1606.01545, 2016.

[15] Putra J. W. G. and Tokunaga, T, "Evaluating text coherence based on semantic similarity graph", In Proceedings of TextGraphs-11: the Workshop on Graph-based Methods for Natural Language Processing, pp. 76–85, August, 2017.

[16] Parikh A. P., Täckström O., Das D. and Uszkoreit J., "A decomposable attention model for natural language inference", In: Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, Austin, Texas, Association for Computational Linguistics, pp. 2249–2255, November, 2016.

[17] Connor Holmes, Daniel Mawhirter, Yuxiong He, Feng Yan, Bo Wu, "GRNN: Low-Latency and Scalable RNN Inference on GPUs", In Fourteenth Eurosys Conference (Eurosys '19), March, 2019.

[18] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakon Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, Illia Polosukhin, "Attention is all you need", 31st Conferencence on Neural Information Processing Systems (NIPS), Long Beach, CA, USA, December, 2017.

[19] Jacob Devlin and Ming-Wei Chnag, Research Scientists Google AI Language, "https://ai.googleblog.com/2018/11/open-sourcing-bertstate-of-art-pre.html", November, 2018.

[20] Jacob Devlin, Ming-Wei Chang, Kenton Lee, Kristina Toutanova, "BERT:Pre-training of Deep Bidirectional Transformers for Language Understanding" Google AI Language, arXiv:1810.04805v2 [cs.CL], 24 May, 2019.

[21] Mohsen Mesgar, Michael Strube, "A Neural Local Coherence Model for Text Quality Assessment", In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, Association for Computational Linguistics, pp. 4328–4339, October, 2018.

[22] Moon H. C., Mohiuddin T., Joty S. and Chi X., "A Unified Neural Coherence Model", Conference on Empirical Methods in Natural Language Processing and 9th International Joint Conference on Natural Language Processing, arXiv e-prints, 2019.

[23] M. Bao, J. Li, J. Zhang, H. Peng and X. Liu, "Learning Semantic Coherence for Machine Generated Spam Text Detection", The 2019 International Joint Conference on Neural Networks (IJCNN), pp. 1-8 2019.

[24] Elham Mohammadi, Timothe Beiko and Leila Kosseim, "On the Creation of a Corpus for Coherence Evaluation of Discursive Units", In Proceedings of the 12th Language Resources and Evaluation Conference, Marseille, France, European Language Resources Association, pp. 1067–1072, 2020.

[25] Panitan Muangkammuen, Sheng Xu, Fumiyo Fukumoto, Kanda Runapongsa Saikaew, and Jiyi Li, "A Neural Local Coherence Analysis Model for Clarity Text Scoring", In Proceedings of the 28th International Conference on Computational Linguistics, Barcelona, Spain (Online), International Committee on Computational Linguistics, pp. 2138–2143, 2020.

[26] Callan D. and Foster, J, "Evaluation of Interest and Coherence in Machine Generated Stories", In CEUR Workshop Proceedings, Vol. 3105, pp. 212–223, 2021.

[27] Q. H. Nguyen and M. Zaslavskiy, "Incoherent Sentence Detection in Scientific Articles in Russian and English," 2021 29th Conference of Open Innovations Association (FRUCT). 10.23919/FRUCT52173.2021.9435478. pp. 267-273, 2021.

[28] D. J. Pierson, "The Top 10 Reasons Why Manuscripts Are Not Accepted for Publication", Respiratory care, October, 2004.

[29] E.I. Blees and M.M. Zaslavskiy, "Criteria for text conformity to scientific style", Scientific and Technical Journal of Information Technologies, Mechanics and Optics, vol.19, pp.299-305, April, 2019.

[30] Filip Bendevski, Jumana Ibrahim, Tina Krulec, Theodore Waters, Nizar Habash, Hanan Salam, Himadri Mukherjee, and Christin Camia, "Towards Automatic Narrative Coherence Prediction", In Proceedings of the 2021 International Conference on Multimodal Interaction (ICMI '21), https://doi.org/10.1145/3462244.3479895, pp. 18-22, October, 2021.

[31] Noji H. and Takamura H., "An Analysis of the Utility of Explicit Negative Examples to Improve the Syntactic Abilities of Neural Language Models", arXiv e-prints, 2020.

[32] Comunicar, Revista Científica de Comunicación y Educación, https://doi.org/10.3916/comunicar, 2021.

[33] Rana Ankush, Gonzalez-Agirre Aitor, Miranda-Escalada Antonio, and Krallinger Martin, "MESINESP: Medical Semantic Indexing in Spanish - Train dataset (1.0)", Zenodo, https://doi.org/10.5281/zenodo.3826492, 2020.

[34] Sayak, "arXiv Paper Abstracts: arXiv paper abstract dataset for building multi-label text classifiers", Kaggle, https://www.kaggle.com/datasets/spsayakpaul/arxiv-paper-abstracts, 2021.

[35] Vallejo Diego, Morillo Paulina, Ferri Cèsar, "ICMLA 2014/2015/2016/2017 Accepted Papers Data Set", Mendeley Data, V2, 10.17632/wj5vb6h9jy.2, 2019.

[36] Souza F.D. and Filho J.B. de O. e S., "BERT for Sentiment Analysis: Pre-trained and Fine-Tuned Alternatives", Computational Processing of the Portuguese Language. https://doi.org/10.1007/978-3-030-98305-5_20. pp. 209–218, 2022.

[37] Devlin J., Chang M. W., Lee K. and Toutanova, K., "Bert: Pre-training of deep bidirectional transformers for language understanding", In NAACL HLT 2019 - 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies - Proceedings of the Conference, Association for Computational Linguistics, Vol. 1, pp. 4171–4186, 2019.

[38] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever and Ruslan Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting", Journal of Machine Learning Research.10.5555/2627435.2670313, January, 2014.

[39] Bisong, E, "Building Machine Learning and Deep Learning Models on Google Cloud Platform", Apress. https://doi.org/10.1007/978-1-4842-4470-8_7, 2019.

[40] Blagec K., Dorffner G., Moradi M. and Samwald M, "A critical analysis of metrics used for measuring progress in artificial intelligence", arXiv preprint arXiv:2008.02577, 2020.