

Mexican Spanish Paraphrase Identification using Data Augmentation

Abdul Meque, Fazlourrahman Balouchzahi, Grigori Sidorov and Alexander Gelbukh

Instituto Politécnico Nacional (IPN), Centro de Investigación en Computación (CIC), Mexico City, Mexico

Abstract

Reorganizing words in a passage using synonyms and different words without changing the main message delivered in the original sentence is called paraphrasing. Simplifying, clarification or taking quotes, etc. In this paper, we address a Paraphrase Identification model for Mexican Spanish text pairs. A data augmentation step was done using Google Translate API, and then three different similarity algorithms, namely: Jaccard, Cosine, and Spacy similarity were used to create a similarity vector for each text pair. The paraphrase identification task was modeled as binary classification of text pairs into two classes, namely: Paraphrases and Not-Paraphrases. The proposed methodology with voting classifier of three machine learning classifiers obtained a F1-score of 0.8754 for paraphrases category.

Keywords

Paraphrase, Spanish, Similarity, Data Augmentation

1. Introduction

A restatement of a passage using different and synonym words to express the same message for various purposes such as simplifying, clarification, etc., is called paraphrasing [1]. Therefore, two sentences are considered as a paraphrase if they express almost the same message with minor differences [2]. The task of automatically identifying whether two sentences convey similar or same meaning is called Paraphrase Identification (PI). The PI task is a primary task in Natural Language Processing (NLP) [2].

Gemma et al., [3] organized a shared task called Paraphrase Identification in Mexican Spanish (PAR-MEX)¹ during the Iberian Languages Evaluation Forum (IberLEF) 2022 conference. The aim of the shared task was a sentence-level PI on Mexican Spanish, and it was modeled as a binary classification in which each sentence pairs were classified into Paraphrase and Non-paraphrase categories.


Tackling the challenge proposed in the mentioned shared task, we proposed a PI model, based on data augmentation and three similarity algorithms, namely: Jaccard, Cosine, and Spacy similarity. The proposed methodology consists of first translating sentences into English and


IberLEF2022, September 2022, A Coruña, Spain.

✉ gmequea@cic.ipn.mx (A. Meque); fbalouchzahi2021@cic.ipn.mx (F. Balouchzahi); sidorov@cic.ipn.mx (G. Sidorov); gelbukh@cic.ipn.mx (A. Gelbukh)

🌐 <https://www.abdulmeque.com/> (A. Meque); <https://sites.google.com/view/fazlfrs/home> (F. Balouchzahi); <https://www.cic.ipn.mx/~sidorov/> (G. Sidorov); <http://www.gelbukh.com/> (A. Gelbukh)

🆔 0000-0002-1068-1810 (A. Meque); 0000-0003-1937-3475 (F. Balouchzahi)

 © 2022 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

 CEUR Workshop Proceedings (CEUR-WS.org)

¹<https://sites.google.com/view/par-mex/home>

French and then translation back to Spanish as data augmentation step using deep-translator ² tool that allows for the use of various translation engines, and then as the second step Spacy, Jaccard and cosine similarity are used to calculate the similarities. The vectors obtained from Term Frequency-Inverse Document Frequency (TF-IDF) (for word n-grams and Bi and Tri syntactic n-grams), Spacy, and transformers for the original sentences and the augmented versions of them are used for similarities calculation.

A function that calculates the degree of similarity between two sets (here a pair of text objects) is called a similarity coefficient, and there are several coefficients explored in the literature such as viz Jaccard, Dice, and Cosine coefficient, etc. [4]

Jaccard Similarity is a common proximity measurement used to compute the similarity between two objects, such as two text documents. Jaccard similarity can be used to find the similarity between two asymmetric binary vectors, or to find the similarity between two sets.

The Jaccard similarity, developed by Paul Jaccard ³ is a similarity measure that calculates similarities between two given sets and provides a value from 0 to 1, the more close to one the more similar are two given sets. Jaccard similarity can be used to compute the similarity between two text documents as well, by finding the similarity between their corresponding vectors. The general equation for Jaccard similarity is given in equation (1), where A and B indicate two sets.

$$Jaccard(A, B) = \frac{|A \cap B|}{|A \cup B|} \quad (1)$$

The Cosine similarity is widely used to compute document similarities in NLP tasks. The Cosine similarity utilizes the cosine of the angle between two vectors to the similarity between them to determine whether the same direction is being pointed by these two vectors [5].

The main difference between these two similarities is that the Cosine similarity divides the number of common attributes by the product of A and B's distance from zero as is represented in Equation (2) while in Jaccard similarity, the number of common attributes is divided by the number of attributes that exist in at least one of the two objects.

The other similarity used in this study is Spacy similarity, that is done by finding similarity between word Spacy vectors in the vector space.

$$\cos(\mathbf{A}, \mathbf{B}) = \frac{\mathbf{A}\mathbf{B}}{\|\mathbf{A}\|\|\mathbf{B}\|} = \frac{\sum_{i=1}^n \mathbf{A}_i\mathbf{B}_i}{\sqrt{\sum_{i=1}^n (\mathbf{A}_i)^2} \sqrt{\sum_{i=1}^n (\mathbf{B}_i)^2}} \quad (2)$$

The proposed method obtained a F1-score of 0.8754 on blinded test set provided by shared task organizers.

The rest of the paper is arranged as follows: A review of related work is included in Section 2, and the methodology is discussed in Section 3. Experiments, and results are described in Section 4 followed by concluding the paper with future work in Section 5.

²<https://pypi.org/project/deep-translator/>

³https://en.wikipedia.org/wiki/Paul_Jaccard

2. Related Work

In NLP, Paraphrase Identification has been the subject of various studies, using different approaches, most notably:

In [6], a new approach to Language Model for Few-Shot learner, the Entailment as Few-Shot Learner, which at its core does a conversion of a given NLP task into an entailment task and fine-tune the model with fewer K examples that would originally be required. The authors conducted experiments and reported performance improvements in various tasks and corpora, including QQP (F1-Score 89.2% on the full dataset and 67.3% with K=8) and MRCP (F1-Score 91.0% on the full dataset, 76.2% with K=8) for Paraphrase Identification and STS-B (F1-Score 75.7% on full dataset and 71.0% with K=8) for Sentence Similarity, compared against Glue Benchmark.

In [7] a new term-weighting metric called TF-KLD is proposed, which combined with n-grams and latent features and reportedly shows some improvement against the state-of-the-art at the time. The TF-KLD allows for the discriminative reweighting of distributional features before factorization, thus impacting the induction of the latent representation, which is then transformed into a vector for learning. When tested on the MRCP dataset, this approach reportedly achieves 73.58% accuracy with an 80.55% F1-Score.

An unsupervised transformer-based approach called Transformer-based Sequential Denoising Auto-Encoder (TSDAE) was presented in [8] and tested on various datasets and tasks on heterogeneous domains, including the TwitterPara Paraphrase Identification task. On the TwitterPara task consisting of two similarity datasets, namely the Twitter Paraphrase Corpus and the Twitter News URL Corpus, TSDAE showed an improvement in the Average Precision against the gold confidence scores and the similarity scores from the models, achieving around 66.0% in the evaluation metric of Spearman's rank correlation.

STRUCTBERT, an extension of the BERT language model that incorporates language structures into the pre-training, presented in [9] outperformed the SOTA in the Glue benchmark which includes the QQP (74.4/90.7), STS-B (92.8/92.4) and MRCP (93.6/91.5) datasets. To achieve these results the authors extended the BERT model by incorporating two objectives, the word objective, and sentence objective, the latter specifically aimed at the sentence pair task, revolves around the expansion of the sentence prediction task to include both next and previous sentence prediction.

Previous work leveraging Machine Translations to aid in Paraphrase Identification has been presented before, such in [10] where the author reported comparable results when using ensemble methods and translation engines on Russian corpora translated to English. In [11] new SOTA is achieved by both Paraphrase Identification and Paraphrase Generation using a combination of data sampling and a fine-tuned Text-To-Text Transfer Transformer (T5) model.

3. Methodology

The proposed method contain 4 major phases, namely: Data augmentation, Vectorization, Similarity Calculation, and Model Construction.

3.1. Data Augmentation

In this study, Google Translate is used for data augmentation using the deep-translator tool. The deep-translator tool provides a flexible API that utilizes various translation platforms such as Google Translate, Microsoft Translator, Yandex translator, etc., that enable translation among many languages ⁴. The original texts from the dataset were translated to the French and English languages (separately) and then translated back to Spanish and used for further processing. Figure 1 represents the process followed for data augmentation using translation. The process was done for both text columns (text and paraphrased version of it in dataset) and original and augmented texts are passed to the next phase for vectorization.

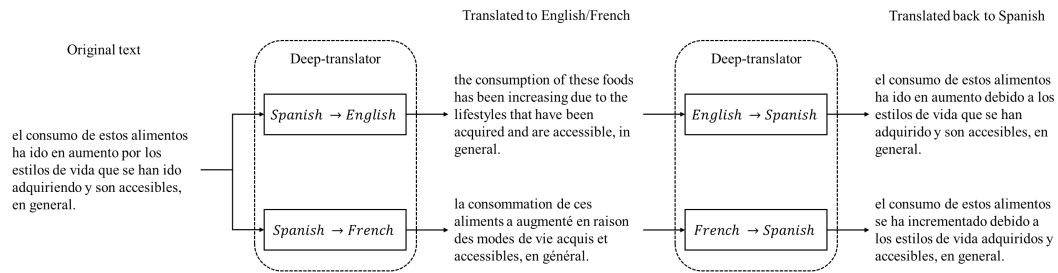


Figure 1: Data augmentation process

3.2. Vectorization

The vectorization phase enables us to calculate the similarity between two corresponding texts later. Therefore, three types of vectors such as vectors from Spacy, sentence transformers, and TF-IDF are generated from texts. Spacy library with en_core_web_lg ⁵ and es_core_news_lg ⁶ models for English and Spanish respectively were used to obtain the respective vectors.

The sentence-transformers ⁷ framework was used to compute vector representation for sentences. It supports various transformers such as: BERT, RoBERTa, XLM-RoBERTa, and etc. The eduardofv/stsb-m-mt-es-distiluse-base-multilingual-cased-v1 that is pre-trained model on a Semantic Textual Similarity (STS) dataset in Spanish language ⁸ was used to generate the other set of vectors in this study.

In addition to the mentioned vectors, traditional word n-grams and bi-tri syntactic n-grams (sn-grams) [12] are also extracted from texts and were vectorized using TF-IDF. The process of producing vectors for texts are illustrated in Figure 2.

⁴list of all translation platforms and languages here: <https://deep-translator.readthedocs.io/en/latest/README.html>

⁵<https://spacy.io/models/en>

⁶<https://spacy.io/models/es>

⁷<https://pypi.org/project/sentence-transformers/>

⁸https://huggingface.co/datasets/stsb_multi_mt

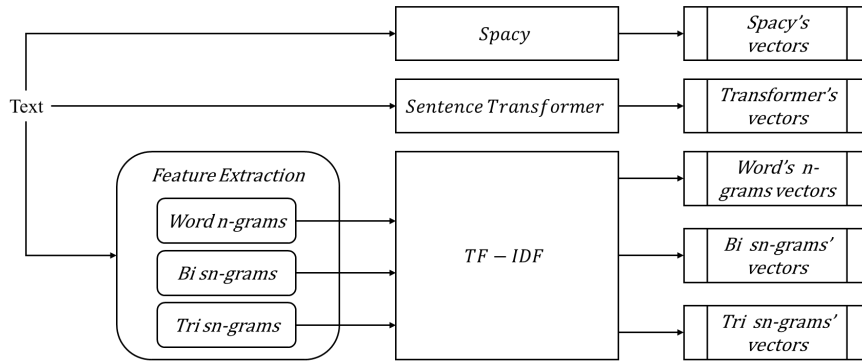


Figure 2: Vectorization process

3.3. Similarity Calculation

Once all vectors for all text pairs (original and augmented ones) were generated, three similarities, namely: Spacy, Cosine, and Jaccard similarities in turn were calculated for each vector pair. The obtained similarity scores for different vector pairs are used to build a new vector called similarity vector for the input text pair. In the other words, for each text pair in the dataset, a similarity vector was crated using similarity score obtained from their generated vectors from previous phase. Figure 3 represents the structure of similarity vector, and it is shown that Spacy, Jaccard and Cosine similarities were computed for various vector types for original and augmented ones (Spacy similarity only were computed for Spacy word vectors for Original and augmented texts).

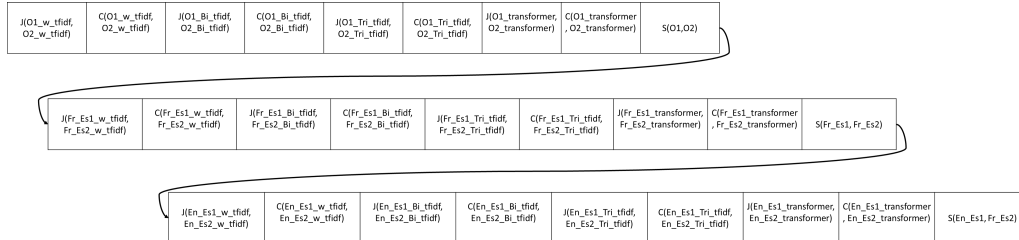


Figure 3: Similarity vector (J: Jaccard, C: Cosine, S: Spacy, O: Original, Fr_Es: Augmented through French to Spanish translation, En_Es: Augmented through English to Spanish translation, w_tfidf: word n-grams, Bi_tfidf: Bi-sngrams, Tri_tfidf: Tri-sngrams)

3.4. Model Construction

Three traditional Machine Learning (ML) classifiers, namely: Multi-layer Perceptron (MLP), Logistic Regression (LR), and Support Vector Machine (SVM) were used as estimators to build a robust Voting Classifier (VC) based on soft voting. The idea of ensembling ML classifiers is to utilize the strength of each individual classifier to improve the performance of the classification model [13].

Table 1

Hyperparameters for the ML classifiers

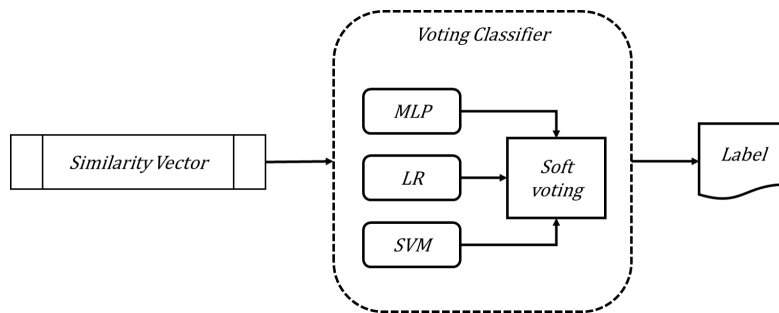
Classifier	Parameters
MLP	hidden_layer_sizes= (150,100,50), activation= 'relu', solver= 'adam', random_state=1
LR	Default parameters
SVM	C= 0.6, degree= 2, gamma= 1, kernel= 'linear', probability= True

Table 2

Performance of proposed methodology

Set	F1-score
Validation	0.8889
Test	0.8754

We fed the similarity vectors obtained from the procedure described in 3.3, to the ensemble classifier tuned using the hyperparameters shown in 1, each of the classifiers predictions are weighted by the classifier's importance and summed up, using the soft voting scheme, the label with the greatest sum of weighted probabilities is selected. The hyperparameters for each estimator are set according to 1. The architecture of VC classifier is graphically presented in Figure 4.

**Figure 4:** Voting classifier architecture

4. Experiments and Results

The dataset provided by PAR-MEX shared task [3] organizers consists of 7382 text pairs as training set that was distributed into two classes, namely: Paraphrases (P) and Not paraphrases (NP). The organizers also provided the participants with blinded validation and test sets for experiments. The validation and test sets contain 97 and 2819 text pairs respectively. The objective of the task was to identify the paraphrased pairs, therefore, the F1-score of P class was used for final evaluation and ranking of teams.

The results presented in Figure 5 illustrate the final leaderboard that reveal a very competitive performance among teams, where the difference between best performing team and last ranking team is only 0.098. The complete performance of our proposed model is presented in Table 2.

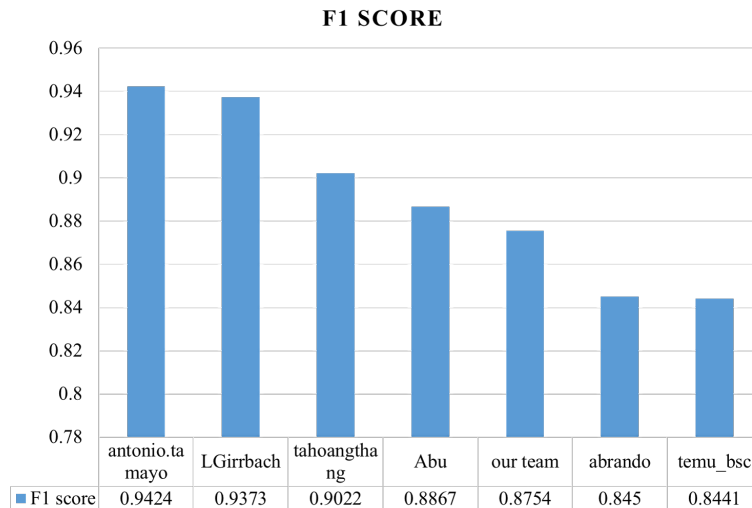


Figure 5: F1-score comparison for Paraphrases class among all participants

5. Conclusion and Future Work

In the present paper, we reported our proposed model submitted to PAR-MEX shared task on paraphrase identification on Mexican Spanish text pairs. In the proposed methodology, we utilized translation as a data augmentation and create a similarity vector for each text pair. Word n-grams, Bi/tri n-grams and Spacy vectors were used to compute the similarities and the similarity vector were obtained using three algorithms namely: Jaccard, Cosine, and Spacy similarity. The generated vectors were used to train a soft VC and results were reported as a F1-score of 0.8754 on P class. As future work, we would like to enhance our performance using different feature sets and statistical algorithms.

Acknowledgments

The work was done with partial support from the Mexican Government through the grant A1-S-47854 of CONACYT, Mexico, grants 20220852 and 20220859 of the Secretaría de Investigación y Posgrado of the Instituto Politécnico Nacional, Mexico. The authors thank the CONACYT for the computing resources brought to them through the Plataforma de Aprendizaje Profundo para Tecnologías del Lenguaje of the Laboratorio de Supercómputo of the INAOE, Mexico and acknowledge the support of Microsoft through the Microsoft Latin America PhD Award.

References

- [1] V. Rus, R. Banjade, M. Lintean, On Paraphrase Identification Corpora, in: Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14), 2014, pp. 2422–2429.

- [2] H. Wang, F. Ma, Y. Wang, J. Gao, Knowledge-Guided Paraphrase Identification, in: Findings of the Association for Computational Linguistics: EMNLP 2021, 2021, pp. 843–853.
- [3] G. Bel-Enguix, H. Gomez-Adorno, G. Sierra, J.-M. Torres-Moreno, J.-G. Ortiz-Barajas, J. Vasquez, Overview of PAR-MEX at Iberlef 2022: Paraphrase Detection in Spanish Shared Task, *Procesamiento del Lenguaje Natural* 69 (2022).
- [4] V. Thada, V. Jaglan, Comparison of Jaccard, Dice, Cosine Similarity Coefficient to Find Best Fitness Value for Web Retrieved Documents using Genetic Algorithm, *International Journal of Innovations in Engineering and Technology* 2 (2013) 202–205.
- [5] J. Han, M. Kamber, J. Pei, Getting to Know Your Data, in: J. Han, M. Kamber, J. Pei (Eds.), *Data Mining (Third Edition)*, The Morgan Kaufmann Series in Data Management Systems, third edition ed., Morgan Kaufmann, Boston, 2012, pp. 39–82. URL: <https://www.sciencedirect.com/science/article/pii/B9780123814791000022>. doi:<https://doi.org/10.1016/B978-0-12-381479-1.00002-2>.
- [6] S. Wang, H. Fang, M. Khabsa, H. Mao, H. Ma, Entailment as Few-Shot Learner, 2021. URL: <https://arxiv.org/abs/2104.14690>. doi:10.48550/ARXIV.2104.14690.
- [7] Y. Ji, J. Eisenstein, Discriminative Improvements to Distributional Sentence Similarity, in: *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, Association for Computational Linguistics, Seattle, Washington, USA, 2013, pp. 891–896. URL: <https://aclanthology.org/D13-1090>.
- [8] K. Wang, N. Reimers, I. Gurevych, TSDAE: Using Transformer-based Sequential Denoising Auto-Encoder for Unsupervised Sentence Embedding Learning, 2021. URL: <https://arxiv.org/abs/2104.06979>. doi:10.48550/ARXIV.2104.06979.
- [9] W. Wang, B. Bi, M. Yan, C. Wu, Z. Bao, J. Xia, L. Peng, L. Si, StructBERT: Incorporating Language Structures into Pre-training for Deep Language Understanding, 2019. URL: <https://arxiv.org/abs/1908.04577>. doi:10.48550/ARXIV.1908.04577.
- [10] D. Kravchenko, Paraphrase Detection Using Machine Translation and Textual Similarity Algorithms, 2018, pp. 277–292. doi:10.1007/978-3-319-71746-3_22.
- [11] H. Palivela, Optimization of Paraphrase Generation and Identification using Language Models in Natural Language Processing, *International Journal of Information Management Data Insights* 1 (2021) 100025. URL: <https://www.sciencedirect.com/science/article/pii/S2667096821000185>. doi:<https://doi.org/10.1016/j.jjime.2021.100025>.
- [12] G. Sidorov, F. Velasquez, E. Stamatatos, A. Gelbukh, L. Chanona-Hernández, Syntactic n-grams as Machine Learning Features for Natural Language Processing, *Expert Systems with Applications* 41 (2014) 853–860.
- [13] F. Balouchzahi, A. B K, H. L. Shashirekha, MUCS@LT-EDI-EACL2021:CoHope-Hope Speech Detection for Equality, Diversity, and Inclusion in Code-Mixed Texts, in: *Proceedings of the First Workshop on Language Technology for Equality, Diversity and Inclusion*, Association for Computational Linguistics, Kyiv, 2021, pp. 180–187. URL: <https://aclanthology.org/2021.ltedi-1.27>.