IIMASnlp at GenoVarDis Task: Exploring Zero-shot and CRF Approaches to NER Task in Genomic Variants and Related Diseases

Orlando Ramos-Flores[1],*, Helena Gómez-Adorno[1] and Edgardo Galán-Vásquez[1]

[1] *Instituto de Investigaciones en Matemáticas Aplicadas y en Sistemas, Universidad Nacional Autónoma de México, Mexico City, Mexico*

Abstract

This paper outlines our approach to address Named Entity Recognition (NER) in genomic variants and related diseases. We performed experiments using Conditional Random Fields (CRF), Bi-directional Long Short-Term Memory networks (Bi-LSTM), fine-tuning RoBERTa, and a zero-shot approach. We also used data augmentation with LLaMA3-8b, increasing our training data by approximately 300% . Additionally, we created several dataset variants for our experiments. While developing our approach, we explored a post-processing method to enhance our models' performance. Ultimately, we combined our best-trained CRF model with the zero-shot approach and applied the post-processing method to achieve our highest scores.

Keywords

Named Entity Recognition, Genomic Variants, Diseases, Zero-shot, LLaMA3-8b, CRF, Prompting

1. Introduction

Automatic information extraction has become a crucial task in Natural Language Processing (NLP), proving indispensable across various fields, including the biomedical sector. The task consists of identifying the entities in a raw text and classifying them in their corresponding tags. According to [1], a named entity is, broadly speaking, any item that can be identified with a proper name, such as a person, location, or organization. The task of Named Entity Recognition (NER) involves detecting segments of text that represent these proper names and tagging them according to their specific type.

The NER task in the biomedical field presents significant challenges. One of the foremost difficulties is the need for datasets curated by experts, particularly those in languages other than English, such as Spanish. Addressing this challenge is essential for improving the accuracy and reliability of NER systems in diverse linguistic contexts. Another challenge, equally as important as the first one, is to develop approaches that can identify and classify entities with high performance in low-resource domains. To promote advancements in this area, the

IberLEF 2024, September 2024, Valladolid, Spain

orlando.ramos@aries.iimas.unam.mx (O. Ramos-Flores); helena.gomez@iimas.unam.mx (H. Gómez-Adorno); edgardo.galan@iimas.unam.mx (E. Galán-Vásquez)

0000-0002-8579-4123 (O. Ramos-Flores); 0000-0002-6966-9912 (H. Gómez-Adorno); 0000-0002-9165-1241 (E. Galán-Vásquez)

CEUR Workshop Proceedings (CEUR-WS.org)

GenoVarDis Task [2] was introduced in IberLEF 2024 [3], which encourages researchers to participate and develop innovative approaches to tackle these challenges. This initiative aims to foster collaboration and innovation, ultimately enhancing the performance and applicability of NER systems in the biomedical field.

This paper introduces a methodology for identifying entities related to genomic variants and diseases from electronic health records. Our approach comprises the combination of the Conditional Random Forest (CRF) algorithm [1], a zero-shot approach using LLaMA3-8b [4] (a quantized 4bit model) through the use of prompts. Also, we include a post-processing approach to increase the NER task.

The remainder of this article is structured as follows. In Section 2, we describe the dataset used for the task and briefly discuss its creation. In Section 3, we describe the models proposed for the Named Entity Recognition, Entity Linking, and Entity Classification subtasks, as well as pre-processing and post-processing of the data. In section 4, we describe the experiments carried out with the models and report their results. Finally, in Section 5, we discuss the obtained results and give ideas for future system improvement.

2. Dataset

The GenoVarDis dataset [2] consists of 633 annotated documents, with 427 documents in the training dataset, 70 in the development dataset, and 136 in the test dataset. Table 1 provides a detailed breakdown of the annotations for each split. The dataset exhibits a clear imbalance in the number of annotations per entity type. The most frequently annotated entities are "*Diseases*" and "*Genes*", while "*Transcript*" entities have the fewest annotations. We used data augmentation with LLaMA3-8b to expand the training dataset.

Table 1
Entity distribution on the GenoVarDis dataset. The columns **TF**, **DF**, and **ATF** represent the training frequency, development frequency, and augmented training frequency for the entities in each dataset.

No.	TF	DF	ATF	Entity Tag
1	4,028	588	13,048	Disease
2	3,093	550	9,929	Gene
3	496	103	585	DNAMutation
4	271	53	752	OtherMutation
5	139	12	242	DNAAllele
6	120	15	367	SNP
7	51	11	23	NucleotideChange-BaseChange
8	1	1	2	Transcript
Total	8,199	1,333	24,948	

Additionally, we used the Spacy[1] library (version 3.7.4) to split the sentences in each document for both the training, development and the augmented training datasets as well as to get the tokens by each sentence.

[1] https://spacy.io/usage

Table 2 provides a detailed description of the sentences and tokens obtained. The "*Dataset*" column represents the analyzed dataset (ATrain means augmented training dataset), and the "*Sent. Avg.*" column shows the average number of sentences per document, the "*Tokens Avg.*" column displays the average number of tokens per sentence, and the "*Longest Token*" column indicates the longest token by character count in the dataset.

Table 2
Statistics for the training, development, and test datasets

Dataset	Sentences	Sent. Avg.	Tokens Avg.	Longest Token
Train	4,543	10.66	29.61	242
Dev	791	11.30	28.88	109
Test	1,503	11.05	23.07	41
ATrain	14,673	34.44	24.65	242

Table 2 shows that the average number of sentences and tokens in the training, development, and test datasets are similar, but the augmented training dataset is different. Additionally, the longest token length is the same for the training and augmented training datasets, but it differs for the development and test datasets.

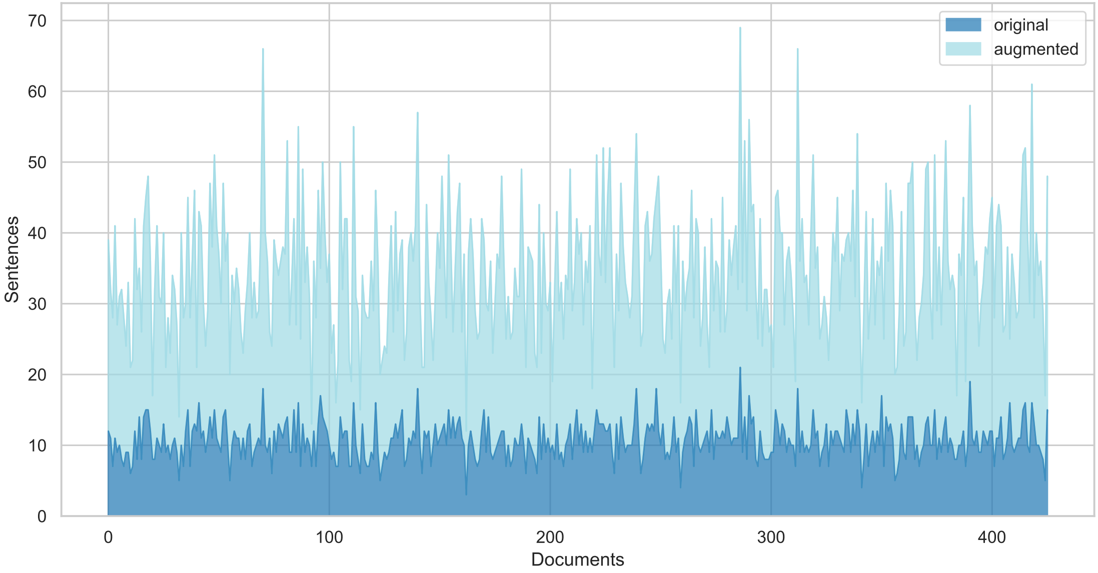


Figure 1: Distribution of augmented and original sentences.

Figure 1 presents the distribution of augmented sentences by document. The sentence augmentation process selected only those sentences containing at least one entity. As a result, 3,444 sentences from the original 4,543 in the training dataset were used. For each of these sentences, three similar sentences were generated, with the corresponding entities identified in them. The x-axis represents the 427 documents in the training dataset, while the y-axis displays the number of sentences in each document. In the chart, original sentence counts are represented by the color blue, while the light blue color represents the original count plus the augmented sentences.

3. Methodology

The methodology involves the following steps: The first step is pre-processing, which includes extracting the text, and annotations for training and development datasets by aligning the tokens and labels. Following pre-processing, we move to the training of the CRF, the Bi-LSTM, and the RoBERTa models. Alternatively, we apply a data augmentation technique to increase the number of sentences in the training dataset. Additionally, we employ a zero-shot approach using LLaMA3-8b [4] for the NER task. At last, in the post-processing step, we incorporate the predictions from a selected trained model, eliminate any incorrect entities using specialized lexicons, apply a set of regular expression patterns, and then combine the results with the zero-shot results (only for the "Disease" class) to generate the final system predictions.

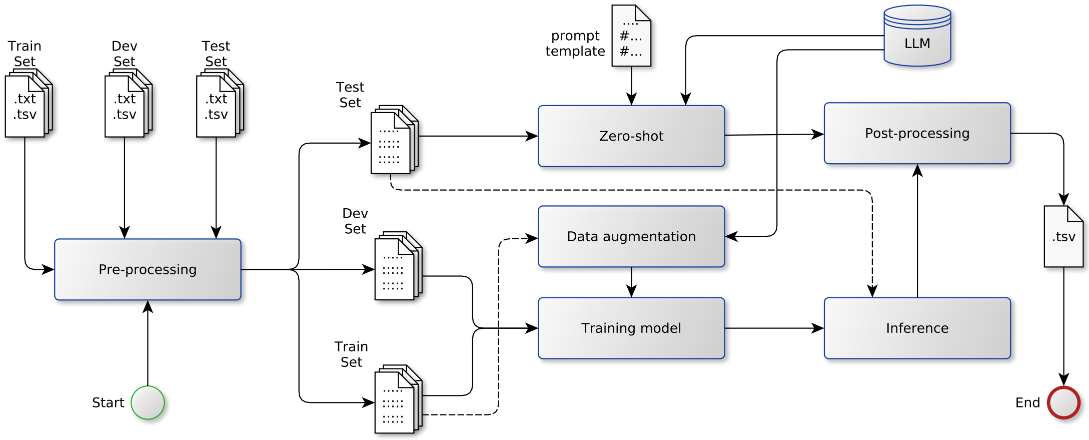


Figure 2: Methodology.

3.1. Pre-processing

The first step in our process is extracting sentences from the documents using the Spacy library. The second step involves tokenizing each sentence. However, we encountered issues with Spacy's tokenization. For example, the term “IL-6” is labeled as a “Gene” and it appears in the text as “IL-6-174.”, the Spacy tokenizer does not separate the string “IL-6-174.” into “IL-6”, “-174” and “.”; the tokenizer preserves "IL-6-174." as is. Other similar examples are: the string “colangitis intrahepática familia” labeled as “Disease” omitting the last character “r” at the end, in the same aspect, the string “enfermedad coronaria prematur” labeled as “Disease” omits the character “a” at the end. Initially, we tried to resolve this by setting up special cases in Spacy, but we found this approach to be complicated, conflicting, and laborious.

To resolve the issue, we inserted blank spaces in cases similar to the one described above, before the tokenization step. Following this, we proceed with the original steps one and two without defining special tokenization cases.

In the last step, we aligned all sentence elements (tokens, Part-of-Speech (POS) tags, and labels) according to the BIO scheme. These steps were applied to both the training and development datasets. For the test dataset, we apply the standard tokenization process from the Spacy library.

3.2. Data augmentation

After discovering that the Bi-LSTM and RoBERTa models yielded poor results in the development stage, we tried to enhance our models by increasing the training dataset.

The process involved selecting sentences that contained at least one entity tag. We used the Large Language Model (LLM) LLama3-8b[2] to generate three sentences similar to the original and identify the corresponding entities within them. We built an augmented training dataset from the generated and training sentences by aligning sentence elements (tokens, Part-of-Speech tags, and labels) according to the BIO scheme. Data augmentation was only applied to the training dataset.

3.3. NER models training

We trained various supervised models. We first chose the Conditional Random Fields (CRF) algorithm, which has demonstrated good results with small datasets [5]. Additionally, we implemented other algorithms, including Bidirectional Long Short-Term Memory (Bi-LSTM) with and without word embeddings and fine-tuning a RoBERTa model.

3.3.1. CRF model

We utilized two different CRF algorithms. The first CRF model received a defined set of features for each token, including the token's form (whether it is lowercase, uppercase, title case, or a digit), its POS tag, and specific token segments (the last two and three characters of the token, and the two first characters of the POS tag).

The second CRF model received word embeddings obtained from BERT [6]. We employed the "last hidden state" to extract word embeddings from BERT. These embeddings are dynamic and capture the nuances of each word in its specific context, making them highly valuable for a wide range of NLP tasks. The training dataset contains a total of 134,523 tokens, with 12,275 unique tokens, for which we obtained word embeddings from the pre-trained BERT[3] to train the CRF.

3.3.2. Bi-LSTM model

We also implemented a Bi-LSTM model and conducted experiments using an embeddings layer with random embeddings and word embeddings, similar to our approach with the CRF model. In the first configuration, the general hyperparameters included embeddings with a dimension

[2] https://huggingface.co/unsloth/llama-3-8b-Instruct-bnb-4bit

[3] https://huggingface.co/google-bert/bert-base-multilingual-cased

of 300, 128 neurons in the hidden layer, and 10 epochs. In the second configuration, we used word embeddings from BERT, resulting in hyperparameters of embeddings with a dimension of 768, 128 neurons in the hidden layer, and 10 epochs.

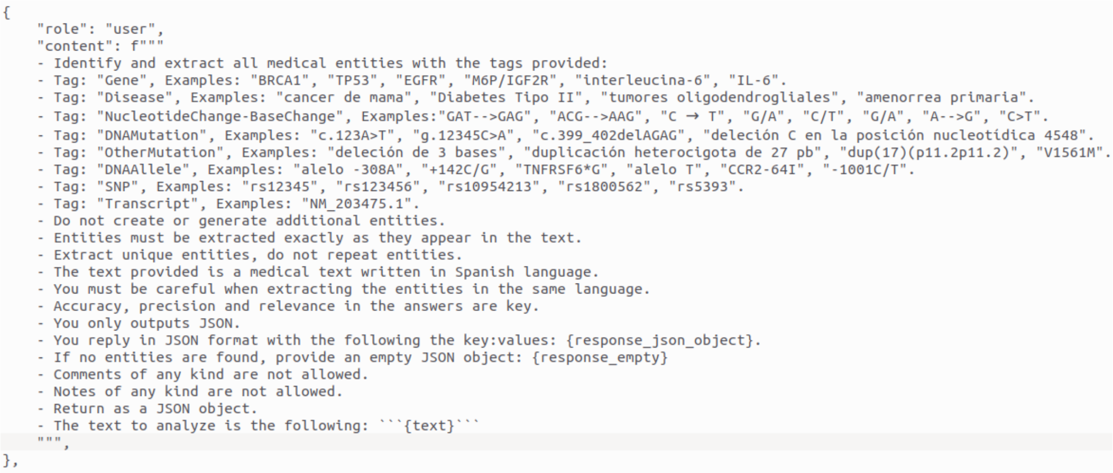
3.3.3. RoBERTa fine-tuning

We used the Biomedical pre-trained language model of the RoBERTa [7] model for Spanish from [8]. We fine-tuned this model using the same hyperparameters as the original, with three epochs. The HuggingFace model we used was developed by Barcelona Super Computing (BSC[4]).

It is worth noting that experiments with the models described above, with slight modifications, were conducted both during the development and evaluation stages.

3.4. Zero-shot NER

As part of our research to address this task, we propose a zero-shot approach to identify and classify entities. In this stage, we experiment with Mistral-7b[5] and LLaMA3-8b models. However, the Mistral-7b LLM showed inferior results compared to LLaMA3-8b. Therefore, we decided to use LLaMA3-8b for all experiments. As mentioned earlier, the quantized 4-bit versions of both LLMs were used in the experiments.



````
{
    "role": "user",
    "content": f"""
    - Identify and extract all medical entities with the tags provided:
    - Tag: "Gene", Examples: "BRCA1", "TP53", "EGFR", "M6P/IGF2R", "interleucina-6", "IL-6".
    - Tag: "Disease", Examples: "cancer de mama", "Diabetes Tipo II", "tumores oligodendrogliales", "amenorrea primaria".
    - Tag: "NucleotideChange-BaseChange", Examples:"GAT-->GAG", "ACG-->AAG", "C → T", "G/A", "C/T", "G/A", "A-->G", "C>T".
    - Tag: "DNAMutation", Examples: "c.123A>T", "g.12345C>A", "c.399_402delAGAG", "deleción C en la posición nucleotídica 4548".
    - Tag: "OtherMutation", Examples: "deleción de 3 bases", "duplicación heterocigota de 27 pb", "dup(17)(p11.2p11.2)", "V1561M".
    - Tag: "DNAAllele", Examples: "alelo -308A", "+142C/G", "TNFRSF6*G", "alelo T", "CCR2-64I", "-1001C/T".
    - Tag: "SNP", Examples: "rs12345", "rs123456", "rs10954213", "rs1800562", "rs5393".
    - Tag: "Transcript", Examples: "NM_203475.1".
    - Do not create or generate additional entities.
    - Entities must be extracted exactly as they appear in the text.
    - Extract unique entities, do not repeat entities.
    - The text provided is a medical text written in Spanish language.
    - You must be careful when extracting the entities in the same language.
    - Accuracy, precision and relevance in the answers are key.
    - You only outputs JSON.
    - You reply in JSON format with the following the key:values: {response_json_object}.
    - If no entities are found, provide an empty JSON object: {response_empty}
    - Comments of any kind are not allowed.
    - Notes of any kind are not allowed.
    - Return as a JSON object.
    - The text to analyze is the following: ```{text}```
    """,
},
````

Figure 3: Zero-shot LLaMA3-8b prompt template

The template prompt used is as follows: for the system, the prompt is {"**role**": "*system*", "**content**": "*You are an expert in medicine and an expert annotator of medical entities. Follow the instructions below to extract the entities from the text.*"}.

The user message is illustrated in Figure 3, where the required tagged entities are described with examples and instructions to return either a JSON object or an empty JSON object if no entities are found. The variable *response_json_object* in Figure 3 has the structure {'**tag1**':

[4] https://huggingface.co/PlanTL-GOB-ES/roberta-base-biomedical-clinical-es

[5] https://mistral.ai/news/announcing-mistral-7b/

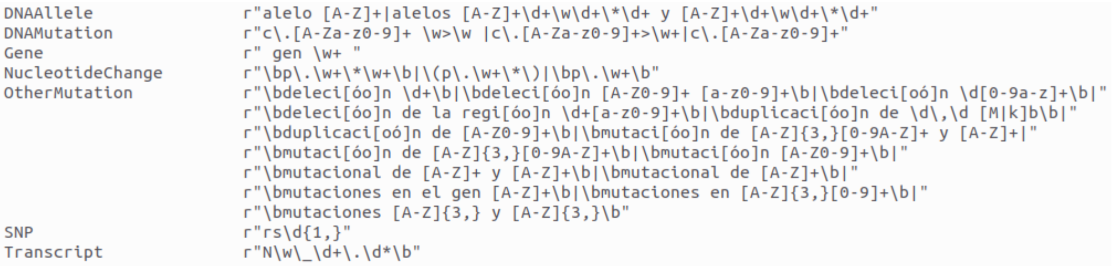
['entity1', 'entity2'], '**tag2**': ['entity1', 'entity2']}. The *response_empty* variable represents an empty JSON object: { }. The variable *text* contains each document to be analyzed.

The zero-shot approach is applied to the development and test dataset to identify the eight entities listed in Table 1.

3.5. Post-processing

In the post-processing, we aim to identify entities that the supervised models nor the LLM could capture. For this purpose, we used lexicons of DNA mutations (5,877,319 items), Human Genes from KEGG (100,429 items), and an SNP database (34,225,003 items). Additionally, we defined a set of regular expression patterns.

Primarily, the lexicons were used to verify previously extracted entities. We focused on analyzing text with challenging entities for our trained models to identify. Figure 4 lists the customized pattern sets for each entity, except for the *Disease* entity.



```
DNAAllele            r"alelo [A-Z]+|alelos [A-Z]+\d+\w\d+\*\d+ y [A-Z]+\d+\w\d+\*\d+"
DNAMutation          r"c\.[A-Za-z0-9]+ \w>\w |c\.[A-Za-z0-9]+>\w+|c\.[A-Za-z0-9]+"
Gene                 r" gen \w+ "
NucleotideChange     r"\bp\.\w+\*\w+\b|\(p\.\w+\*\)|\bp\.\w+\b"
OtherMutation        r"\bdeleci[óo]n \d+\b|\bdeleci[óo]n [A-Z0-9]+ [a-z0-9]+\b|\bdeleci[oó]n \d[0-9a-z]+\b|"
                     r"\bdeleci[óo]n de la regi[óo]n \d+[a-z0-9]+\b|\bduplicaci[óo]n de \d\,\d [M|k]b\b|"
                     r"\bduplicaci[oó]n de [A-Z0-9]+\b|\bmutaci[óo]n de [A-Z]{3,}[0-9A-Z]+ y [A-Z]+|"
                     r"\bmutaci[óo]n de [A-Z]{3,}[0-9A-Z]+\b|\bmutaci[óo]n [A-Z0-9]+\b|"
                     r"\bmutacional de [A-Z]+ y [A-Z]+\b|\bmutacional de [A-Z]+\b|"
                     r"\bmutaciones en el gen [A-Z]+\b|\bmutaciones en [A-Z]{3,}[0-9]+\b|"
                     r"\bmutaciones [A-Z]{3,} y [A-Z]{3,}\b"
SNP                  r"rs\d{1,}"
Transcript           r"N\w\_\d+\.\d*\b"
```

Figure 4: Regular expression patterns designed

For the *Disease* entity, we also experimented with the CIE-10 lexicon (14,497 items), which yielded poor results. Nevertheless, we achieved significant results with the zero-shot approach, so we decided to accept the *Disease* entities identified by this approach. The same cannot be said for the other entities retrieved in the zero-shot approach.

The final process involves the following steps: 1) Obtain results from a trained model. 2) Use lexicons to verify recognized entities and remove any incorrect ones. 3) Apply a set of regular expressions to find new entities. 4) Utilize the zero-shot result for the "Disease" class and combine it with step 3. Finally, we created the output file in the required .tsv format for submission to the shared task.

4. Experiments and Results

We describe the experiments conducted with the trained models, zero-shot approaches, and post-processing methods during both the development and evaluation stages.

4.1. Experimental datasets

We created various datasets for the experiments, as listed in Table 3. The (*) symbol denotes the original dataset, meaning the data preserves its original form, i.e., as it was provided by the organizers without mixing or splitting. The prefix "*dev*" or "*eval*" in the "Dataset name" column indicates the stage during which the dataset was used. The Genovardis_dev_ori dataset comprises the original training dataset, but we further divide it into training and validation during the development stage. In the same way, the Genovardis_dev_ori_aug is the augmented dataset from the original training data. The prefix "*ori*" means that the original data was not merged, while "*mix*" indicates that the original training and development sets were merged and shuffled before splitting. The splits of the original training/development datasets were used in Genovardis_eval_ori during the evaluation stage.

Each dataset includes the following columns: id, texts, tokens, ner_tags, and pos_tags. The "id" corresponds to the PMID column from the original data, while the other items represent sentences.

Table 3
Experimental datasets

Dataset name	Training (%)	Validation (%)
Genovardis_dev_ori	4,088 (90%)	455 (10%)
Genovardis_dev_ori_aug	13,205 (90%)	1,468 (90%)
Genovardis_eval_ori	4543*	791*
Genovardis_eval_mix	4,800 (90%)	534 (10%)
Genovardis_eval_ori_aug	14,673*	791*
Genovardis_eval_mix_aug	12,371 (80%)	3,093 (20%)

4.2. Development stage

We experimented with two models: the CRF and fine-tuned RoBERTa models. For both approaches, we tested the original training dataset and the augmented training dataset. Table 4 presents the results from the CRF and RoBERTa models after training. When using the original dataset, the RoBERTa model performed the best. However, when using an augmented training dataset, the CRF model showed better performance than the RoBERTa model. It's evident that data augmentation improves performance, but after applying the CRF trained on the development dataset for the shared task, the model's performance was lower in the metrics.

Table 4
Results of the models (trained in the development stage) on the internal validation set.

Dataset	Model	Precision	Recall	F1-score
Genovardis_dev_ori	CRF	**0.79**	0.72	0.75
	RoBERTa	0.76	**0.80**	**0.78**
Genovardis_dev_ori_aug	CRF	**0.91**	**0.92**	**0.91**
	RoBERTa	0.85	**0.92**	0.88

The original training dataset consisted of 4,088 sentences (90%) for training and 455 sentences (10%) for validation. The augmented training dataset included 13,205 sentences (90%) for training and 1,468 sentences (10%) for validation. The best results, obtained after submission, were achieved with the CRF model using the original training dataset, yielding a **precision of 0.628571**, a **recall of 0.379595**, and an **F1 score of 0.473340**.

4.3. Evaluation stage

In this stage, we experimented with the CRF, RoBERTa, and the Bi-LSTM models with different datasets and word embeddings (see prefix "**-e**" in the model name). During our experiments, we tried the zero-shot approach. However, the results were not promising, except for the "Disease" class. Results are presented in Table 5, with the best scores for precision, recall, and f1 metrics highlighted in bold.

Table 5
Results of the models (trained on the evaluation stage) on the internal validation set.

Dataset	Model	Precision	Recall	F1-score
Genovardis_eval_ori	CRF	**0.73**	0.55	0.63
	CRF-e	0.66	0.62	0.64
	Bi-LSTM	0.58	0.57	0.58
	RoBERTa	**0.73**	**0.74**	**0.73**
Genovardis_eval_ori_aug	CRF	**0.73**	0.56	0.63
	Bi-LSTM	0.53	0.60	0.56
	RoBERTa	0.70	**0.76**	**0.73**
Genovardis_eval_mix	CRF	**0.78**	0.72	0.75
	CRF-e	**0.78**	0.73	0.75
	Bi-LSTM	0.68	0.69	0.68
	RoBERTa	0.77	**0.82**	**0.80**
Genovardis_eval_mix_aug	CRF	**0.90**	**0.90**	**0.90**
	Bi-LSTM	0.84	0.88	0.86
	RoBERTa	0.85	0.91	0.88

The experiments presented in Table 5 detail the datasets and models used. The RoBERTa model performed well in the experiments but produced disappointing results in the submissions to CodaLab during both the development and evaluation stages. Although data augmentation improved the model performance, the scores in the CodaLab submissions were not encouraging.

To ensure accuracy, we implemented a post-processing step to validate the entities identified by the CRF model. This involved using specialized lexicons and applying regular expression patterns to identify all entities except those classified as "Disease". Lastly, we integrated only the diseases identified through the zero-shot approach to produce the final file in .tsv format.

4.4. Shared Task results

Finally, we present our best scores for each stage of the GenoVarDis competition. Table 6 lists our final rank position, the stages, and the results for each metric.

Table 6
Final shared task scores

Rank	Stage	Team	F1	Precision	Recall
1	Development	ander.martinez	**0.740269**	0.700603	**0.784696**
2		VictorMov	0.668254	**0.709351**	0.631658
3		**Ours**	0.473340	0.628571	0.379595
1	Evaluation	ander.martinez	**0.820977**	**0.822350**	**0.819610**
2		VictorMov	0.793455	0.790643	0.796287
3		ELiRF-VRAIN	0.734940	0.777483	0.696811
4		Milimeter98	0.548269	0.610754	0.497382
5		**Ours**	0.530055	0.731769	0.415516

5. Conclusions

Addressing the NER task presents a challenge due to the limited data resources involving genomic variants, genes, and their related diseases, leading to the following conclusions. The best outcome was achieved by integrating the CRF, post-processing, and zero-shot, resulting in scores of **F1**: 0.53, **Precision**: 0.73, and **Recall**: 0.41. Our data augmentation technique did not enhance performance, likely because the augmented sentences were too similar to the original ones, with only slight word changes. Both the RoBERTa fine-tuning and Bi-LSTM experiments also yielded poor results, likely due to the limited data available for experimentation.

We found that the CRF model was best at generalizing across both the development and evaluation stages. Furthermore, the zero-shot approach with LLaMA3-8b showed promising results in recognizing diseases but not with the other entities, and our post-processing showed some improvement in the overall scores.

For future work, we plan to use embeddings from RoBERTa within a Bi-LSTM-CRF architecture. We also aim to explore fine-tuning LLaMA3-8b instead of using a zero-shot approach. Additionally, we plan to develop a post-processing method that integrates both techniques and expands the set of regular expressions.

6. Acknowledgments

This paper has been supported by PAPIIT projects IN104424, TA101722, and CONAHCYT CF-2023-G-64. We also want to thank Ricardo Villareal and Rita Rodríguez for their help with the computing resources and Eng. Roman Osorio, thank you for assisting with the student administration of the project.

References

[1] D. Jurafsky, J. H. Martin, Speech and language processing. vol. 3, 2014.

[2] M. M. Agüero-Torales, C. Rodríguez Abellán, M. Carcajona Mata, J. I. Díaz Hernández, M. Solís López, A. Miranda-Escalada, S. López-Alvárez, J. Mira Prats, C. Castaño Moraga,

D. Vilares, L. Chiruzzo, Overview of GenoVarDis at IberLEF 2024: NER of Genomic Variants and Related Diseases in Spanish, Procesamiento del Lenguaje Natural 73 (2024).

[3] L. Chiruzzo, S. M. Jiménez-Zafra, F. Rangel, Overview of IberLEF 2024: Natural Language Processing Challenges for Spanish and other Iberian Languages, in: Proceedings of the Iberian Languages Evaluation Forum (IberLEF 2024), co-located with the 40th Conference of the Spanish Society for Natural Language Processing (SEPLN 2024), CEUR-WS.org, 2024.

[4] AI@Meta, Llama 3 model card (2024). URL: https://github.com/meta-llama/llama3/blob/main/MODEL_CARD.md.

[5] K. Liu, Q. Hu, J. Liu, C. Xing, Named entity recognition in chinese electronic medical records based on crf, in: 2017 14th Web Information Systems and Applications Conference (WISA), 2017, pp. 105–110. doi:10.1109/WISA.2017.8.

[6] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, Bert: Pre-training of deep bidirectional transformers for language understanding, in: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), 2019, pp. 4171–4186.

[7] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, V. Stoyanov, Roberta: A robustly optimized bert pretraining approach, arXiv preprint arXiv:1907.11692 (2019).

[8] C. P. Carrino, J. Armengol-Estapé, A. Gutiérrez-Fandiño, J. Llop-Palao, M. Pàmies, A. Gonzalez-Agirre, M. Villegas, Biomedical and clinical language models for spanish: On the benefits of domain-specific pretraining in a mid-resource scenario, 2021. arXiv:2109.03570.