FRE at GenoVarDis: A sane approach to Disease and Genomic Variant NER

Ander Martínez[1]

[1] *AI & Computing Research Group, Fujitsu Research of Europe Ltd., Spain*

Abstract

Fujitsu Research of Europe (FRE) has participated in the GenoVarDis [1] competition on Named Entity Recognition (NER) of variants, genes and associated diseases. This competition was part of the IberLEF 2024 [2] campaign. In this paper, we describe our approach to the challenge and an analysis of our results. Our approach consisted of a combination of Pretrained Language Model fine-tuning, Conditional Random Fields (CRF), Byte-Pair Encoding dropout (BPE dropout) and model ensembling. With this solution, we ranked first in the competition. In this paper, we analyze the benchmark dataset and our results. Now that the gold data for the test set has been released, we can consider how our results could have been better.

Keywords

Natural Language Processing, Named Entity Recognition, Conditional Random Fields, RoBERTa

1. Introduction

Among other tasks proposed by the IberLEF 2024 [2] campaign, GenoVarDis fell in the *Biomedical NLP* category, with the full title of *GenoVarDis: NER in Genomic Variants and related Diseases* [1]. The task released a much-needed benchmark dataset in Spanish on Biomedical Named Entity Recognition (NER). The presentation of the dataset cited tmVar3 [3] and BERN2 [4] as similar datasets in English, both of them well known albeit limited in size.

Named Entity Recognition (NER) is one of the cornerstones of text mining, a necessary step to go from unstructured to structured data. In the recent years, fine-tuning Large Language Models (LLM), such as BERT[5] or RoBERTa[6], has become the most popular approach to NER. When compared to previous non-LLM deep learning approaches[7], the newer LLM approaches require less data to train. However, hand-labeled (or *gold-standard*) data is still necessary, and so, the dataset released for this competition makes a great contribution to Spanish language Biomedical NLP.

Our team has been interested on working with unstructured biomedical data, and we recently participated [8] in the SympTEMIST challenge [9], on Spanish language symptom NER. For the GenoVarDis competition we have used a solution similar to the one that we used in that occasion.

For our submission, we have used a sane combination of well known techniques that we think delivers best results on most occasions. The techniques that we used are LLM fine-tuning for NER, Conditional Random Fields (CRF), BPE-Dropout, and model ensembling with majority voting.

Using this approach we ranked first in the GenoVarDis competition. In this paper, we analyze the benchmark dataset and our results. Now that the gold data for the test set has been released, we can consider how our results could have been better.

IberLEF 2024, September 2024, Valladolid, Spain
EMAIL: ander.martinez@fujitsu.com (A. Martínez)
ORCID: 0000-0003-2290-8194 (A. Martínez)

Table 1
Dataset text statistics. Size of each partition. The rows *Mean length in characters* and *Mean length in words* represent the mean size of the documents contained.

	Train	Dev	Test
Total documents	427	70	136
Mean length in characters	1,809.65	1,885.11	1,503.50
Mean length in words	276.78	287.46	219.37

Table 2
Dataset statistics for the three partitions. The *entities* column represents the total number of entities in the partition. The *chars* and *words* columns represent the average length of the mentions in chars and words separated by space. The row `Nucl...eChange` stands for `NucleotideChange-BaseChange`.

	Train			Dev			Test		
category	entities	chars	words	entities	chars	words	entities	chars	words
`Disease`	4028	15.98	2.15	588	15.53	2.07	1433	20.1	2.43
`Gene`	3093	7.27	1.33	550	7.95	1.38	514	6.46	1.22
`DNAMutation`	496	14.31	2.31	103	15.46	2.61	73	7.92	1.05
`OtherMutation`	271	25.3	4.3	53	21.15	3.0	22	19.86	2.68
`DNAAllele`	139	9.27	1.71	12	8	1.58	15	7.2	2.2
`SNP`	120	8.52	1.03	15	8.47	1	42	8.69	1
`Nucl...eChange`	51	11.43	2.43	11	9.82	2	1	3	1
`Transcript`	1	11	1	1	11	1	1	11	1

2. Dataset

The GenoVarDis dataset is a collection of texts that have been labeled with the spans of mentions of various genes, mutations and diseases. Each text or document consists of a title and a block of text such as a paragraph. The blocks of text normally contain multiple sentences. The dataset was distributed partitioned into `train`, `development` and `test`, with the gold data of the test partition only released after the competition concluded.

Table 1 shows some statistics of the text contained in dataset. We see that of a total of 633 documents (and 1,109,153 characters), about 70% was directed to the training data, 12% to the development and 18% to the test. We also observe that the documents in the training and development partitions are similar in length but a bit longer than the texts in the test partition.

Table 2 shows the statistics of the entities contained in each of the dataset partitions. The dataset contains entities of eight different categories, all of varying frequencies. In the table, we sorted the categories from most to least frequent. While the categories `Disease` and `Gene` have a fair amount of mentions, mutations are found in different categories and some of them do not have enough data to successfully train a NER model. As an example, the `Transcript` category contains a single mention in the training data. We observe that the `Disease`- and `Gene`-category mentions make up about 90% of the annotations (92.7% of the test annotations), so when micro-averaging the f1-scores of the different entities these will make most of the score of the competition, rendering the mutation annotations not so relevant.

The average length of the annotations shows that the mentions of the `OtherMutation` category are particularly long. These are descriptions of the mutations such as "*insertion introduced eight additional amino acids*". We have extracted this English translation from the official task description. All samples of the `Transcript` category are single word, 11-character long: `"NM_203475.1"`, `"NG_008724.1"` and `"NM_000747.2"`. These examples are extracted from the training, development and test respectively. They follow a regular pattern and could be extracted using regular expressions.

3. Approach

In the introduction, we have described our technology as a combination of a few well-known techniques: Language Model Fine-tuning and Conditional Random Fields, BPE dropout and Model ensembling. In this section, we will provide a short description of each of them and some details on how they were implemented.

3.1. Language Model Fine-tuning and Conditional Random Fields

The NER task is usually reduced to a token classification task, where each of the tokens (words or subwords) in a text (sentence or paragraph) are classified to a BIO, Beginning-Inside-Outside [10], schema class. This schema represents each mention in the text as a `B-` label followed by zero or more `I-` labels. Tokens that do not belong to a mention are classified as `O` (outside). For eight entity categories we need to have $8 \times 2 + 1 = 17$ classes. That is the `O` class and `B-Disease`, `I-Disease`, `B-Gene`.... Other popular schemas are `SBIO` and `BIOES`.

Fine-tuning of LLM is a very popular approach to train NER models. This approach adds a classification layer on top of the token representations learned by the LLM. The LLM models can be trained on raw (not annotated) text, so they can leverage large amounts of data. An early example of this approach is the original BERT paper [5].

The performance of these models can be improved using Conditional Random Fields (CRFs), although CRFs have been popular long before the introduction of LLMs. [11, Souza et al.] is an example of combining LLMs NER with CRF.

The contribution of CRFs is that they can model the probability of transitioning from one output label to the next one by training an additional matrix. This is usually useful because `I-` labels cannot come without a preceding `B-` label. CRF can help avoid impossible transitions. On prediction time, the Viterbi algorithm[12] is used to produce the most likely sequence of labels after considering the transition probability.

3.2. Subword Representation and BPE Dropout

Texts can be represented as strings of characters. Characters form a closed set of symbols, whether these are Latin characters, or any other set of Unicode characters, which means one can enumerate all of them in a list. However, representing texts as strings or sequences of characters results in very long sequences. Word sequences have been used instead, resulting in shorter sequences for the same text. But words cannot form a closed set of symbols, which means we can encounter words that we have never seen before (out-of-vocabulary). This is a problem when training deep learning (DL) models, particularly for NER, where we expect to encounter many new words. A compromise is using subwords (or wordpieces) that consists in defining a closed set of substrings that can be used to compose words. The closed vocabulary is selected to produce shorter sequences. Byte-pair encoding was originally formulated as a compression algorithm, and later repurposed to represent texts for DL models[13], originally in the context of Neural Machine Translation. After that, BPE has found wide adoption, and it is widely used as the method to present text to DL models.

An alternative to BPE was introduced by [14, Kudo]. A benefit of this approach is that it can produce multiple representations for the same text by segmenting it differently. For this, it requires training a unigram model, and it uses Expectation–Maximization (EM) and Viterbi [12] algorithms to sample segmentations, adding some complexity and being a drawback to its adoption.

BPE dropout[15] was introduced as a simpler alternative to Kudo's unigram approach. It can be applied to existing BPE vocabularies, and so it can also be applied to many of the pretrained language models available at *HuggingFace*, such as RoBERTa. In comparison, the unigram language model subword regularization method uses a statistical model and dynamic

programming to be able to sample different segmentations from the same sequence. BPE dropout uses random noise to discard certain merge-operations, randomly generating a different sequence of subwords each time. This is so because BPE does not store the frequencies of each subword, only the order of the merge-operations. Merge-operations are discarded with a probability p, which is usually 0.1. Provilkov et al. [15] concluded through several experiments that BPE dropout achieves better results. Our systems used BPE dropout during training, with a dropout probability p of 0.1.

3.3. Model Ensembling

Because we are only fine-tuning our models from pretrained LM (and not training from scratch), and because the data available for training is relatively scarce, we can only train our models for a limited number of iterations before they start to overfit.

Training a single model does not take a long time, but its predictions depend on the initialization that was used for the classification layer. Under these circumstances, we can easily combine a few models to make more robust predictions.

We combined five models that were initialized with different seeds but using the same base LLM model. We used a majority voting strategy ensemble the models. Under this simple strategy, each model makes a prediction for each label and the label that got more prediction votes is selected. We observed that this strategy improved the final f1-score.

4. Results

The competition was held on Codalab[16] [1], with a development phase preceding the evaluation phase. During the development phase, the annotations for the development partition were not available.

We trained a model based on the `PlanTL-GOB-ES/bsc-bio-ehr-es` model [17] available at HuggingFace [2]. We submitted the predictions of this model to the Codalab system to make sure that the format of our predictions was correct. The system reports the score of the submission right away and keeps a ranking of all submissions live. We observed that the submissions were ranked with respect to the micro average of the f1-score. With this in mind, we decided it was not worth to optimize for the minority classes and focus on `Gene` and `Disease`.

After the development annotations were released, we trained five models using both training and development partition for 1 epoch. We combined the predictions of the five models as we described in Subsection 3.3. Our submission got an F1-score of 0.820977 and ranked first until the completion of the competition.

The Codalab system only reports the average scores, but after the annotations of the test data were released, we could analyze the errors in our submission. Table 3 shows the scores that we got for each of the categories. Our submission got good scores for both `Gene` and `Disease`, that made 92.67% of the entities (and the score), as shown in Table 2. We also got good results for the `DNAMutation` category: 91.39% F1-score. This category was the third most common category in the training data with 496 mentions. For all the other categories our model did not get good results, but this did not impact the final score. The fourth most common category in the training data was `OtherMutation`, with 271 mentions. Although the number of mentions is more than half of those for `DNAMutation`, our system could only get 16% F1-score for this category, the reason being that these mentions were considerably longer and more complex than the others, as shown in Table 2.

[1] URL: https://codalab.lisn.upsaclay.fr/competitions/17733
[2] URL: https://huggingface.co/PlanTL-GOB-ES/bsc-bio-ehr-es

Table 3
Full results for the submitted prediction. The categories are ordered alphabetically.

category	precision	recall	f1-score	support
DNAAllele	1.0000	0.0667	0.1250	15
DNAMutation	0.8846	0.9452	0.9139	73
Disease	0.8074	0.8193	0.8133	1433
Gene	0.8444	0.8444	0.8444	514
NucleotideChange-BaseChange	0.0000	0.0000	0.0000	1
OtherMutation	0.6667	0.0909	0.1600	22
SNP	1.0000	1.0000	1.0000	42
Transcript	1.0000	0.0000	0.0000	1
micro avg	0.8223	0.8196	0.8210	2101
macro avg	0.7754	0.4708	0.4821	2101
weighted avg	0.8226	0.8196	0.8156	2101

5. Conclusions

We participated in the *GenoVarDis* competition and ranked first. We showed that our standard approach to NER works well for different settings when provided enough training data. Still, there are cases where not enough annotated data is available to train a reliable model. In these cases using dictionary matching and regular expressions is a better option.

References

[1] M. M. Agüero-Torales, C. R. Abellán, M. C. Mata, J. I. D. Hernández, M. S. López, A. Miranda-Escalada, S. López-Alvárez, J. M. Prats, C. C. 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).

[2] 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.

[3] C.-H. Wei, A. Allot, K. Riehle, A. Milosavljevic, Z. Lu, tmVar 3.0: an improved variant concept recognition and normalization tool, Bioinformatics 38 (2022) 4449–4451. URL: https://doi.org/10.1093/bioinformatics/btac537. doi:10.1093/bioinformatics/btac537.

[4] M. Sung, M. Jeong, Y. Choi, D. Kim, J. Lee, J. Kang, BERN2: an advanced neural biomedical namedentity recognition and normalization tool (2022). _eprint: 2201.02080.

[5] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding, 2019. URL: http://arxiv.org/abs/1810.04805. doi:10.48550/arXiv.1810.04805, arXiv:1810.04805 [cs].

[6] 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, 2019. URL: http://arxiv.org/abs/1907.11692. doi:10.48550/arXiv.1907.11692, arXiv:1907.11692 [cs].

[7] S. Chowdhury, X. Dong, L. Qian, X. Li, Y. Guan, J. Yang, Q. Yu, A multitask bi-directional RNN model for named entity recognition on Chinese electronic medical records, BMC bioinformatics 19 (2018) 499. doi:10.1186/s12859-018-2467-9.

[8] A. Martínez, N. García-Santa, FRE @ BC8 SympTEMIST track: Named Entity Recognition, 2023. URL: https://doi.org/10.5281/zenodo.10103882. doi:10.5281/zenodo.10103882.

[9] S. L. López, L. G. Sánchez, E. Farré, L. V. Gimenez, M. Krallinger, SympTEMIST Corpus: Gold Standard annotations for clinical symptoms, signs and findings information extrac-

tion, 2024. URL: https://doi.org/10.5281/zenodo.10635215. doi:10.5281/zenodo.10635215, version Number: 4.

[10] L. A. Ramshaw, M. P. Marcus, Text Chunking using Transformation-Based Learning, 1995. URL: http://arxiv.org/abs/cmp-lg/9505040. doi:10.48550/arXiv.cmp-lg/9505040, arXiv:cmp-lg/9505040.

[11] F. Souza, R. Nogueira, R. Lotufo, Portuguese Named Entity Recognition using BERT-CRF, 2020. URL: http://arxiv.org/abs/1909.10649. doi:10.48550/arXiv.1909.10649, arXiv:1909.10649 [cs].

[12] A. Viterbi, Error bounds for convolutional codes and an asymptotically optimum decoding algorithm, IEEE Transactions on Information Theory 13 (1967) 260–269. URL: https://ieeexplore.ieee.org/document/1054010. doi:10.1109/TIT.1967.1054010, conference Name: IEEE Transactions on Information Theory.

[13] R. Sennrich, B. Haddow, A. Birch, Neural Machine Translation of Rare Words with Subword Units, arXiv:1508.07909 [cs] (2015). URL: http://arxiv.org/abs/1508.07909, arXiv:1508.07909.

[14] T. Kudo, Subword Regularization: Improving Neural Network Translation Models with Multiple Subword Candidates, in: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Association for Computational Linguistics, Melbourne, Australia, 2018, pp. 66–75. URL: https://www.aclweb.org/anthology/P18-1007. doi:10.18653/v1/P18-1007.

[15] I. Provilkov, D. Emelianenko, E. Voita, BPE-Dropout: Simple and Effective Subword Regularization, 2020. URL: http://arxiv.org/abs/1910.13267. doi:10.48550/arXiv.1910.13267, arXiv:1910.13267 [cs].

[16] A. Pavao, I. Guyon, A.-C. Letournel, D.-T. Tran, X. Baro, H. J. Escalante, S. Escalera, T. Thomas, Z. Xu, CodaLab Competitions: An Open Source Platform to Organize Scientific Challenges, Journal of Machine Learning Research 24 (2023) 1–6. URL: http://jmlr.org/papers/v24/21-1436.html.

[17] C. P. Carrino, J. Llop, M. Pàmies, A. Gutiérrez-Fandiño, J. Armengol-Estapé, J. Silveira-Ocampo, A. Valencia, A. Gonzalez-Agirre, M. Villegas, Pretrained Biomedical Language Models for Clinical NLP in Spanish, in: Proceedings of the 21st Workshop on Biomedical Language Processing, Association for Computational Linguistics, Dublin, Ireland, 2022, pp. 193–199. URL: https://aclanthology.org/2022.bionlp-1.19. doi:10.18653/v1/2022.bionlp-1.19.