Vicomtech at CAPITEL 2020: Facing Entity Recognition and Universal Dependency Parsing of Spanish News Articles with BERT Models

Aitor García-Pablos[a], Montse Cuadros[a] and Elena Zotova[a]

[a]*SNLT group at Vicomtech Foundation, Basque Research and Technology Alliance (BRTA), Mikeletegi Pasealekua 57, Donostia/San-Sebastián, 20009, Spain*

Abstract

These working notes describe the participation of Vicomtech NLP team in the CAPITEL task, which is part of the IberLEF 2020. CAPITEL task included two sub-tasks: Named Entity Recognition and Classification (NERC) and Universal Dependency (UD) Parsing for Spanish news articles. A specific system has been designed for each task based on BERT architectures. Both systems have been tested with different settings, where the best ones have been selected for being submitted to the shared task. The resulting systems show robust and competitive systems in both tasks with simple architectures.

Keywords

NERC, dependency parsing, deep learning,

These working notes present an overview of Vicomtech's systems presented in CAPITEL 2020[1] tasks. CAPITEL 2020 is a shared task organized at IberLEF (https://sites.google.com/view/iberlef2020) 2020 campaign.

The *Corpus del Plan de Impulso a las Tecnologías del Lenguaje* (CAPITEL) corpus is an agreement between the Royal Spanish Academy (RAE) and the Secretariat of State for Digital Advancement (SEAD) of the Ministry of Economy within the framework of PlanTL (https://www.plantl.gob.es/Paginas/index.aspx). This corpus is composed of contemporary news articles thanks to agreements with a number of news media providers. CAPITEL has three levels of linguistic annotation: morphosyntactic (with lemmas and Universal Dependencies-style POS tags and features), syntactic (following Universal Dependencies v2 (https://universaldependencies.org/u/dep/index.html), and named entities. The named entity and syntactic layers of revised annotations comprise about 1 million words for the former, and roughly 300,000 for the latter. Due to the size of the corpus and the nature of the annotations, two IberLEF sub-tasks under the more general, umbrella task of CAPITEL @ IberLEF 2020, have been proposed:

- Named Entity Recognition and Classification
- Universal Dependency Parsing

Named Entity Recognition and Classification sub-task aims to challenge participants to apply their systems or solutions to the problem of identifying and classifying name entities

Proceedings of the Iberian Languages Evaluation Forum (IberLEF 2020)

EMAIL: agarciap@vicomtech.org (A. García-Pablos); mcuadros@vicomtech.org (M. Cuadros); ezotova@vicomtech.org (E. Zotova)

ORCID: 0000-0001-9882-7521 (A. García-Pablos); 0000-0002-3620-1053 (M. Cuadros); 0000-0002-8350-1331 (E. Zotova)

(NEs) in Spanish news articles. The following NE categories will be evaluated: Person (PER), Location (LOC), Organization (ORG) and Other (OTH). The metrics used for the evaluation will be Precision, Recall, and F-measure (Micro and Macro average).

Universal Dependency Parsing sub-task aims to challenge participants to apply their systems or solutions to the problem of Universal Dependency parsing of Spanish news articles. The metrics used for the evaluation would be the Unlabelled Attachment Score (UAS) and Labelled Attachment Score (LAS). UAS stands for the percentage of words that have the correct head and LAS, the percentage of words that have the correct head and dependency label.

We have participated in both sub-tasks with two different systems making use of simple architectures with BERT at their base.

These working notes are organised as follows: Section 1 describes the systems presented in both tasks, with all the details concerning architecture and training setup. Section 2 shows the results obtained in the participation of both tasks and Section 3 draws some conclusions and future work.

1. System description

This section provides a description of the systems that we have developed to participate in CAPITEL's task. In the face of the widespread success of Transformer-based architectures [2] in virtually all Natural Language Processing (NLP) tasks, Vicomtech has implemented both systems, one that learns to recognise and classify entities and other to establish relations between sentence elements, based on BERT [3]. The first task, NERC, is approached using a traditional sequence-labelling approach relying on BERT contextual representation for words. The second task, syntactic dependency parsing, is also based on BERT at its core, combining the semantic representation of the tokens to detect the syntactic relations among them. We have tried the same architecture with different pre-trained BERT models.

1.1. Architecture for the NERC system

The NERC system is a Deep Learning model based on BERT (Bidirectional Encoder Representation of Transformers). The model is the simplest approach in which BERT can be used to perform NERC. It makes use of BERT to encode the input, obtaining a contextual embedding for each input token. These contextual embeddings are the input to a fully connected feed-forward layer that helps to classify each token as one of the possible output tags. Figure 1 shows a diagram with the architecture.

We have experimented with two different pre-trained BERT models in our experiments. On the one hand, we have used the BERT-Base Multilingual Cased (https://github.com/google-research/bert/blob/master/multilingual.md) shared by Google. On the other hand, we have used BETO [4], a BERT-base architecture pre-trained only using Spanish texts.

1.2. Architecture for the dependency parsing system

The dependency parsing system is, again, a Deep Learning model relying on BERT. The model uses BERT to encode the input, obtaining contextual embeddings. Then, a tensor operation is

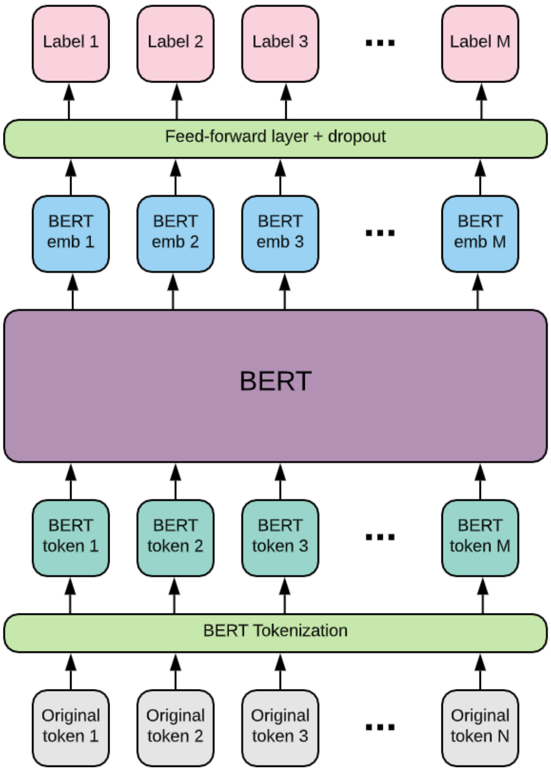

Figure 1: Diagram of the NERC model based on BERT.

performed over the contextual embeddings to obtain an all-vs-all combination of token vectors. This generates $S \times S$ combined embeddings that represent all the possible token pairs, S being the length of the input sequence.

The resulting token-pair representations are then passed to several classification layers to make predictions about the relation between the tokens in each pair. First, the pairs are categorised by a binary classifier that decides whether the tokens that form the pair are connected by a relation (an arc of the dependency tree). The logits resulting from the relation classifier are concatenated with each token-pair embeddings. The resulting representation is passed to a final classification layer to obtain the type of relation for each token pair among the Universal Dependencies types.

Note that this all-vs-all token combination strategy has an exponential computational cost w.r.t. the length of the sequence. This approach could not be applied to full documents. However, the scope of dependency parsing is limited to individual sentences, and since the only operations with the all-vs-all pairs consist of concatenation and a simple matrix multiplication (the feed-forward classification layer) the overall computational cost is feasible.

Figure 2 shows a diagram of the described architecture.

Similar to the NERC task, we have experimented with the BERT multilingual pre-trained model, and with its Spanish pre-trained counterpart, BETO.

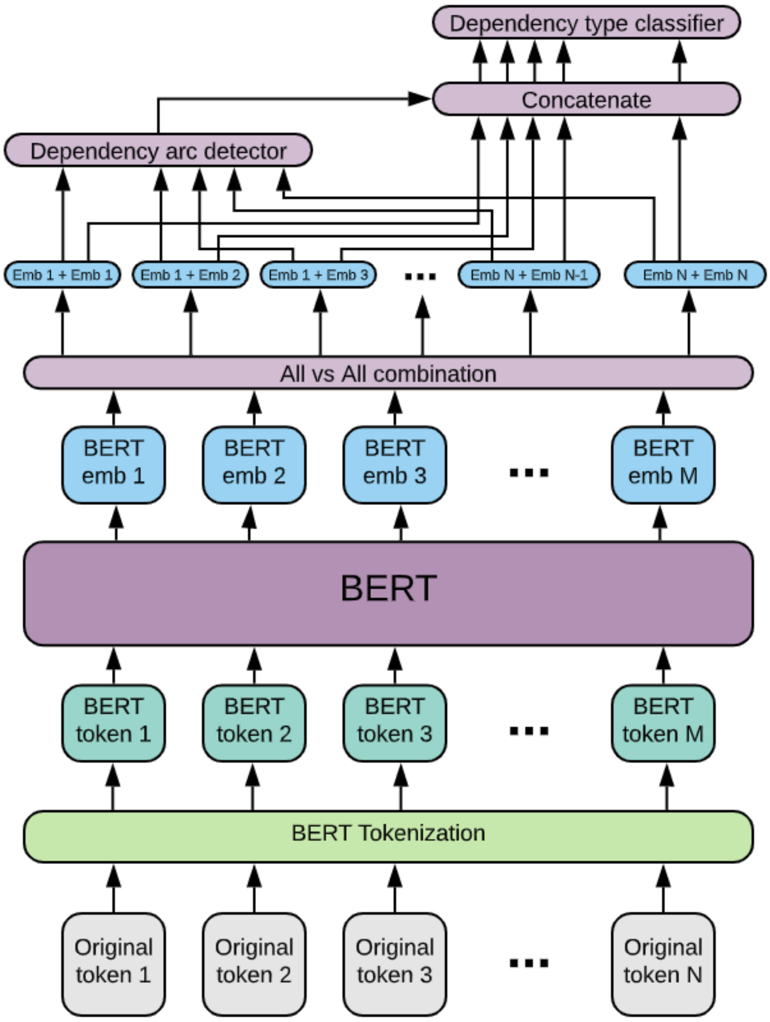

Figure 2: Diagram of the UD model based on BERT.

1.3. Input and output handling

The input for the task comes already tokenized. However, these tokens are not equivalent to what a BERT model expects. Each pre-trained BERT model needs the tokens as they are obtained after using its own tokenizer. Otherwise the input would be meaningless to the pre-trained model. This poses the additional challenge of keeping the alignment between the resulting tokens and the provided labels.

For that, each original token is retokenized with the corresponding BERT tokenizer. This results in additional tokens, since BERT uses WordPiece tokenization that breaks words into word-pieces (e.g. ”Jim Henson” would become ”Jim Hen ##son”). The provided labels are mapped to the head of each token (i.e. the first piece of a sub-word) and the rest of the subtokens are assigned with an special label 'X'. A mapping indicating the correspondence between head subwords and original tokens is stored, so the original token space can be rebuilt at the end of the process, resulting in a token and label sequences of the same length than the original.

1.3.1. BETO vocabulary issues

While using BETO, we realised than the special '[UNK]' token was being hit too often. This token is the default representation used for out of vocabulary values (OOV). When using more traditional tokenization approaches it is usual to encounter OOVs in NLP tasks, due to the limited size of the whole-word vocabularies. But with modern tokenization approaches like WordPiece or BPE this is less common. We discovered that BETO tokenizer's WordPiece vocabulary was missing some common punctuation marks like semicolon ';', or percentage symbol '%'. Also, we noticed that any word containing certain diacritic marks, like 'cigüeña', 'piragüista' or 'Düsseldorf', were automatically marked as unknown. The same happened with the words containing the character 'Ç', rather common in Catalan or French nouns.

Having so many unknown values is inconvenient because that means that all the occurrences of such words will share the same vector representation. The contextual information coming from the surrounding tokens may alleviate the problem, but relevant information is being lost.

In order to deal with these issues we manually added missing punctuation marks and symbols to the BETO vocabulary, using the unused slots of the vocabulary that are reserved for the addition of new words. The newly added symbols would have a randomly initialised value because they were not part of the BETO pre-training, but at least this gives them the chance of learning a meaningful representation during the fine-tuning of the model for the downstream task. The problem with the diacritics was solved replacing the offending characters by their diacritics-free counterparts, e.g. 'Düsseldorf' was converted into 'Dusseldorf'.

1.3.2. Post-processing IOBES tagging

The gold labels for the NERC task follow a *IOBES* tagging scheme, which indicates if a token is at the *B*eginning, *I*nside, *O*utside or *E*nd of a named entity, or if it is an entity spanning a *S*ingle token.

This means that for each given entity type *ENT*, there are four possible labels: $B - ENT$, $I - ENT$, $E - ENT$ or $S - ENT$. Sometimes the model selects a tag that correctly predict the entity type, but it is incorrect with respect the IOBES tagging. Some of these mistakes can be corrected with a simple post-processing step.

A *B* must be followed by an *I* tag or an *E* tag, otherwise it becomes and *S* tag. If an *I* tag is followed by an *O* tag, then the *I* must become an *E* tag, and vice versa.

1.4. Training setup

We have experimented with two different pre-trained BERT models as the core for the semantic representation of the input tokens: BERT-Base Multilingual Cased (https://github.com/google-research/bert/blob/master/multilingual.md) (mBERT) and BETO [4], a BERT model pre-trained on Spanish text. We have used the implementations from the HuggingFace Transformers library (https://huggingface.co/transformers/. We did not perform any in-domain

Table 1
Results of the submitted system compared with the top-scored participant in NERC task (TestSet) and results of Vicomtech system on the development set comparing BERT and BETO.

Dataset	Team	Metric	PER	LOC	ORG	OTH.	Micro	Macro
TestSet	Agerri&Rigau[6]	P	96.40	90.47	**88.63**	**83.36**	**90.50**	**90.43**
		R	**97.46**	**91.74**	87.31	80.69	90.17	90.27
		F1	96.93	91.10	87.96	82.00	90.34	90.30
	Vicomtech (BETO)	P	93.48	89.36	85.76	79.63	87.88	87.81
		R	96.70	88.03	85.76	77.34	88.89	88.09
		F1	95.06	88.69	85.82	78.47	87.99	87.94
DevSet	Vicomtech (BETO)	P	94.93	90.43	85.36	81.74	88.67	88.68
		R	95.66	89.60	87.12	81.19	89.11	89.11
		F1	95.29	90.01	86.23	81.46	88.89	88.89
	Vicomtech (mBERT)	P	94.12	89.89	96.38	78.14	87.94	87.99
		R	95.45	89.96	85.87	81.70	88.86	88.86
		F1	94.78	89.92	86.13	79.88	88.39	88.42

language model fine-tuning for the base models. In this sense, the approach is general and domain-agnostic. The only resource used for fine-tuning the whole system is the training data provided for the tasks. For the NERC task the training data consisted in 22,647 sentences with a validation set of 7,549 sentences, while for the UD task the training set contained 7,086 sentences with a validation set of 2,362 sentences.

The training of the different variants was carried out on 2 Nvidia GeForce RTX 2080 GPUs with ~11GB of memory. We applied the AdamW optimiser [5] with a base learning rate of $2E{-}5$, combined with a linear LR scheduling to warm-up the learning rate during the first 5,000 training steps.

For each trained model, the training monitored the weighted F1-score for the model predictions against the development set (i.e. the entity tags for the NERC system, and the syntactic dependency relations for the dependency parsing system). They were run for a maximum of 500 epochs with an early-stopping patience of 150 epochs. Finally, we chose the model checkpoints that had the best development metrics.

2. Results

Table 1 shows the top-results of the NERC task evaluated on the test set, and the results of our training evaluated on the development set. The first system belongs to the task winner (ragerri), while the second system is ours. In the official ranking of the competition our system appears in the 4th position, after the three runs of ragerri and out of 9 different submissions belonging to 5 different participants. We have used BETO-based system for the submission because in the development set it achieved better results than mBERT, as it is shown in the table.

The results show that our system obtains high scores for all the entity types. The overall results are, on average, 2-3% lower than the best performing system on the task.

Table 2
Results of the submitted system compared with the top-scored participant in UD task (TestSet) and results of Vicomtech system in the development set comparing BERT and BETO.

Dataset	Team	Type	Metric
Test Set	Lendínez[7]	UAS	**91.935**
		LAS	**88.660**
	Vicomtech (BETO)	UAS	91.875
		LAS	88.600
Dev Set	Vicomtech (BETO)	UAS	91.540
		LAS	88.410
	Vicomtech (mBERT)	UAS	91.220
		LAS	87.860

Regarding UD task, Table 2 shows the top-results of the UD task evaluated on the test set and the results of our training evaluated on the development set. Our system achieves the second position in this task out of four different submissions. The results show very similar scores in both metrics, UAS and LAS compared to the winner of the task (0.06 points less in LAS). Again, as for the NERC task, our submitted system is based on BETO instead of mBERT because it achieved better results in the development set.

3. Conclusions

In these working notes we have described our participation in the CAPITEL shared task, for the two available subtasks: NERC and dependency parsing based on Universal Dependencies (UD). We have presented the deep-learning-based architecture of our systems, which rely on pre-trained BERT models as the base for semantic representation of the texts. We have tried different pre-trained BERT models, multilingual-BERT and Spanish-BERT (BETO). Despite the presented systems are simple and domain agnostic they obtain high scores. For the NERC subtask our system is the 4th best performing submission, and our team achieves the 2nd position among five participants. For the UD subtask our system ranks the 2nd achieving only 0.06 points less than the best performing system.

The described systems can almost be considered baselines based on BERT. As future work, we may experiment with other novel transformer architectures, and additional in-domain pre-training or more sophisticated pre-training objectives. We would also experiment with additional layer on top of these basic architectures (from the well-known CRF for NERC to additional self-attention layers). Also, in particular for the NERC model, researching and designing an extensible way of injecting world-knowledge about existing entities would be very interesting.

Acknowledgements

This work has been supported by Vicomtech and partially funded by the project DeepReading (RTI2018-096846-B-C21, MCIU/AEI/FEDER,UE).

References

[1] J. Porta-Zamorano, L. Espinosa-Anke, Overview of CAPITEL Shared Tasks at IberLEF 2020: NERC and Universal Dependencies Parsing, in: Proceedings of the Iberian Languages Evaluation Forum (IberLEF 2020), 2020.

[2] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, I. Polosukhin, Attention Is All You Need, in: Proceedings of the Thirty-first Conference on Advances in Neural Information Processing Systems (NeurIPS 2017), 2017, pp. 5998–6008.

[3] 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.

[4] J. Cañete, G. Chaperon, R. Fuentes, J. Pérez, Spanish Pre-Trained BERT Model and Evaluation Data, in: Proceedings of the Practical ML for Developing Countries Workshop at the Eighth International Conference on Learning Representations (ICLR 2020), 2020, pp. 1–9.

[5] I. Loshchilov, F. Hutter, Decoupled Weight Decay Regularization, in: Proceedings of the Seventh International Conference on Learning Representations (ICLR 2019), 2019.

[6] R. Agerri, G. Rigau, Projecting Heterogeneous Annotations for Named Entity Recognition, in: Proceedings of the Iberian Languages Evaluation Forum (IberLEF 2020), 2020.

[7] F. Sánchez-León, Combining Different Parsers and Datasets for CAPITEL UD Parsing, in: Proceedings of the Iberian Languages Evaluation Forum (IberLEF 2020), 2020.