# Vicomtech at eHealth-KD Challenge 2020: Deep End-to-End Model for Entity and Relation Extraction in Medical Text

Aitor García-Pablos, Naiara Perez, Montse Cuadros and Elena Zotova

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

**Abstract**

This paper describes the participation of the Vicomtech NLP team in the eHealth-KD 2020 shared task about detecting and classifying entities and relations in health-related texts written in Spanish. The proposed system consists of a single end-to-end deep neural network with pre-trained BERT models as the core for the semantic representation of the input texts. We have experimented with two models: BERT-Base Multilingual Cased and BETO, a BERT model pre-trained on Spanish text. Our system models all the output variables—entities and relations—at the same time, modelling the whole problem jointly. Some of the outputs are fed back to latter layers of the model, connecting the outcomes of the different subtasks in a pipeline fashion. Our system shows robust results in all the scenarios of the task. It has achieved the first position in the main scenario of the competition and top-3 in the rest of the scenarios.

**Keywords**

Entity detection, Relation extraction, Health documents



## 1. Introduction

This article describes Vicomtech's participation at the *eHealth Knowledge Discovery challenge* (*eHealth-KD*) 2020 (https://knowledge-learning.github.io/ehealthkd-2020). The challenge involves entity recognition and classification and relation extraction in Spanish health documents, as shown in Figure 1.

The task organisers provided a corpus of 800 sentences for training purposes and 200 sentences for development purposes. The sentences had been manually annotated with a total of 4 entity types and 13 relation types. As Figure 1 shows, entities consist of one or more tokens, which may or may not be contiguous. Furthermore, entities may overlap with each other, and may be the origin and/or target of more than one relation. All these nuances prevent traditional sequence-labelling approaches, based on BIO-tagging or similar schemes, from capturing all the target entities.

The challenge consists of four evaluation scenarios: in the *Main scenario*, systems are tested for both entity recognition and classification and relation extraction; in *Task A*, only entities are evaluated, while in *Task B*, gold entities are provided and only relations are evaluated; finally, the *Transfer scenario* evaluates complete pipelines in documents of a domain different to the training and development data. Further information about eHealth-KD 2020 is provided in the challenge overview article [1].

*Proceedings of the Iberian Languages Evaluation Forum (IberLEF 2020)*

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

ORCID: 0000-0001-9882-7521 (A. García-Pablos); 0000-0001-8648-0428 (N. Perez); 0000-0002-3620-1053 (M. Cuadros); 0000-0002-8350-1331 (E. Zotova)

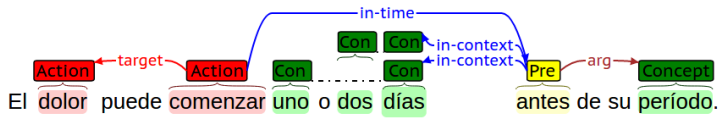


**Figure 1:** Example of eHealth-KD annotations in the sentence "The pain may start a day or two before your period."

In the face of the widespread success of Transformer-based architectures [2] in virtually all Natural Language Processing (NLP) tasks, Vicomtech has implemented a system with BERT [3] that learns to recognise and classify entities and establish relations between them in an end-to-end multi-task fashion. Our system has achieved the best results in the *Main scenario* and ranks among the top-3 results in the rest of the scenarios.

The paper is organised as follows: Section 2 describes the proposed model and the approach followed to represent the data to solve the task; Section 3 presents the results obtained, including a comparison to other competing systems; finally, Sections 4 and 5 comment on several design choices and provide some concluding remarks.

## 2. System description

This section provides a comprehensive description of the system with which the reported results have been obtained. First, we present the architecture of the deep neural network. Next, we explain how the inputs and outputs have been represented and handled in order to solve the task. After, we describe the post-processing rules that help fix potential incongruous outputs of the neural network model. Finally, we present the training settings.

### 2.1. Architecture

The model is a deep neural network that receives the input tokens and jointly emits predictions for several different output variables. These predictions can be grouped into two tasks: *a)* classifying individual tokens, and *b)* classifying relations—the presence, absence or type of a relation—between pairs of tokens. The output variables to be predicted by the model are the following:

- **Entities:** the classification of each individual token into one of the task's entity types or 'O' (from 'Out', meaning that the token is not part of any entity at all, such as "puede" in Figure 1).
- **`multiword` relations:** whether token pairs belong to the same entity (such as "uno" and "días" or "dos" and "días" in Figure 1).
- **`same-as` relations:** whether token pairs are related by the `same-as` relation.
- **Directed relations:** whether token pairs are related by any of the other relation types described in the task.

Unlike the rest of the relations considered, `multiword` and `same-as` relations are bidirectional. In view of several preliminary experiments, which indicated that modelling all the relations together caused noise when predicting directed relations, we decided to model bidirectional relations separately.

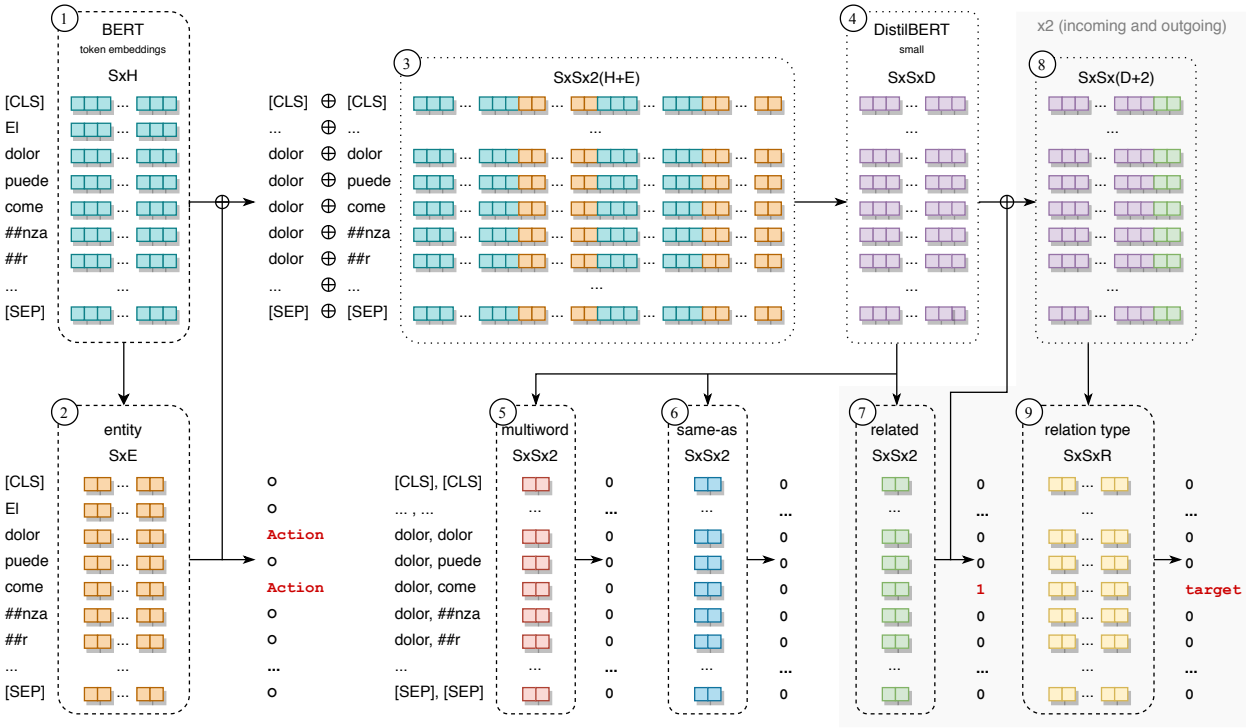


**Figure 2:** High-level diagram of the model, including the shapes output by each layer: H=768 (BERT contextual embedding size); E=5 (number of different entity types plus 'O'), D=768 (contextual embedding of the custom DistilBERT model), R=13 (number of different relation types plus 'O', and minus the `same-as` relation that is modelled separately). `[CLS]` and `[SEP]` are special tokens used by BERT.

An overview of the inner workings of the network is given in Figure 2. The computation of the model starts with the input tokens. The tokens are fed into a BERT model to obtain their contextual embeddings (1). These embeddings are passed to a classification layer that emits logits with the prediction about each token being or not an entity of a certain type (2).

Next, the entity logits are concatenated back to the contextual embeddings, and a tensor operation is performed to obtain an all-vs-all combination of token vectors (3). This generates $S \times S$ combined embeddings that represent all the possible token pairs, $S$ being the length of the input sequence. Further, these embeddings are passed to a small randomly initialised DistilBERT model [4] with only two layers of two attention heads each (4). The objective of this model is to further capture interactions between the token pairs via self-attention.

The resulting token-pair representations are then passed to several classification layers to make predictions about the relation between the tokens in each pair. The pairs are categorised by four binary classifiers that decide respectively whether the tokens that form the pair are connected by a `multiword` relation (5), a `same-as` relation (6), or one of the directed relations (7).

The directed relations are modelled as outgoing arcs and as incoming arcs. That is, if the pair $token_i \oplus token_j$ is linked by a relation of type $R$, the pair $token_j \oplus token_i$ will have the same relation $R$. This is represented in the neural network as a fork of two equal branches of layers, one to model the outgoing relations, and another to model the incoming relations. Notice that Figure 2 has been simplified to show one of the branches only.

At this step, the output of the classification layers tells whether there is a relation or not between a pair of tokens. The `multiword` and `same-as` relations do not require further processing. In the

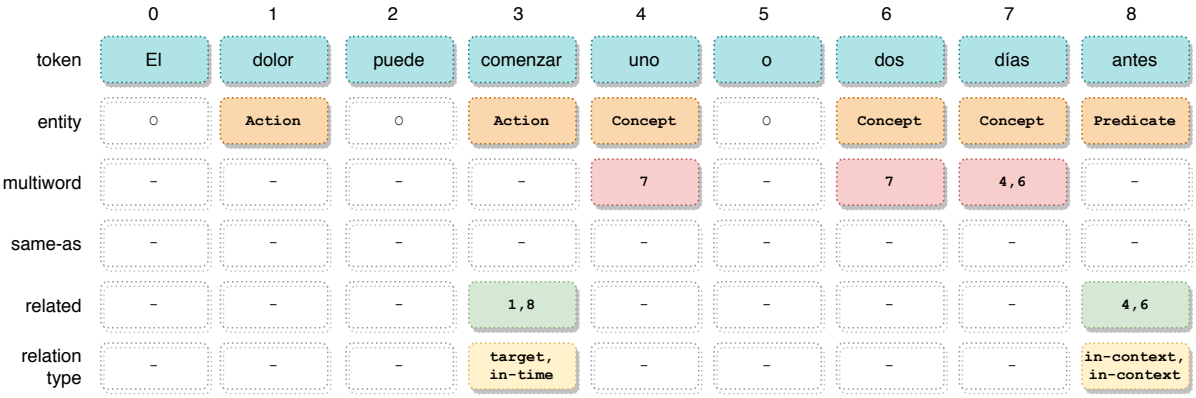


**Figure 3:** Example of data representation of entities and their relations

case of directed relations, however, a relation type or label must be assigned. To that end, the logits of the directed relation classifiers are concatenated back with DistilBERT's layer (8), and the resulting representation passed to a final classification layer to obtain the type of relation for each token pair among the types defined in the task (9). Again, this is done twice: once for the outgoing and another for the incoming arcs.

Overall, the network has seven classifiers, which are built using the same stack of layers: a fully connected linear transformation layer, followed by a dropout layer and a non-linear activation function, and a final linear transformation that outputs the logits for the given output variable. We arbitrarily decided to use Mish [5], not having experimented with other activation functions.

## 2.2. Input and output handling

The training and development corpora have been provided in Brat [6] standoff format. This format is character span-based, while our network works at token level. Furthermore, the output of the neural network needs to be converted back to Brat's format and the annotation schema proposed in the task (i.e., multi-word entities must be reconstructed from the `multiword` relations, and so on). Consequently, our system relies on a set of pre-processing and post-processing transformation steps, explained below.

### 2.2.1. Data representation

Starting from the provided Brat representation, the different pieces of information must be adapted in a manageable way according to our objectives. Figure 3 shows an example of the information representation designed with all the network's output variables.

As mentioned in previous sections, the entities to be detected are not necessarily continuous and they may even overlap. For example, the text span "uno o dos días" contains two independent entities: "uno días" and "dos días". In order to represent this information, we assign to each individual token its corresponding entity label according to the spans from the Brat annotations. The tokens that belong to the same entity span are marked as linked by a `multiword` relation. This approach allows to represent tokens as part of one or more entities regardless of their original position in the text. All the tokens which are part of the same entity are inter-linked via `multiword` relations among them.

A similar approach is followed for `same-as` relations and the directed relations. When entities that span several tokens are connected by one of these relations, only the first token of each entity (in the

order they appear in the text) is marked as being part of the relation. In addition, directed relations are described by an additional output variable to indicate the type of the relation among the different relation types defined in the task.

The described model performs all the tasks end-to-end, using its own entity predictions as input to detect relations. However, in *Task B*, gold entity annotations are provided by the task organisers; systems need to focus on the relations only. In this case, our model accepts gold entity labels along the input tokens, and replaces the predicted entities with a one-hot encoding of the gold ones as the input for detecting relations.

### 2.2.2. Interpreting and reconstructing the model output

The output of the model has to be interpreted to obtain a correct and meaningful label or relation arc for each of the tokens in the original input text. The output of the entity classifier is straightforwardly interpreted as a regular sequence-labelling task selecting the most probable prediction for each individual token.

The reconstruction of relations is more elaborate. The network's outcome for each modelled relation variable forms a $S \times S$ matrix, $S$ being the length of the token sequence, where each position $i$ and $j$; $i, j \in [0, S]$ contains the prediction for the relation between $token_i$ and $token_j$. We implemented two strategies to select the most probable label:

- With *inferencer 1*, only the predicted outgoing relations are used, ignoring the incoming arcs' predictions. In the ideal case, they should be symmetric and, thus, redundant.
- With *inferencer 2*, the prediction values for any outgoing arc $i \longrightarrow j$ and its counterpart incoming arc $i \longleftarrow j$ are summed up before selecting the most probable outcome. This is done for all the modelled relations. In the case of the relation types, this is only possible because the relation types vocabulary is shared between the outgoing and incoming relations, so the same index refers to the same decoded relation type.

Finally, token positions must be corrected to account for deviations and extra offsets introduced by BERT's tokenization (BERT uses WordPiece tokenization [7], which breaks original tokens into subtokens; in addition, it requires that extra special tokens be added which distort the token positions w.r.t. the original input).

### 2.2.3. Post-processing rules

The predictions obtained from interpreting the model's output are still token-based and must be processed further to obtain a representation in Brat standoff format that is compliant with eHealth-KD's annotation scheme. In doing so, we also apply several rules that correct potential inconsistencies produced by the neural network's several classifiers. The post-processing consists in the following steps:

1. Align entity annotations of individual tokens to the original text with a tool provided by Brat developers (https://github.com/nlplab/brat/blob/master/tools/annalign.py).
2. Merge the annotations connected by a `multiword` arc, effectively generating multi-word entities. We decided to keep disjoint sets of tokens: if token $i$ and $j$ are both connected to token $k$ but not with each other, we generate the multi-word entities $i, k$ and $j, k$ instead of $i, j, k$. We also decided to disregard `multiword` arcs to/from tokens that are not classified as entities.
3. Re-assign the `same-as` and directed relations of the tokens in a multi-word entity to the latter.

4. Split into two multi-word entities that contain conjunctions or certain punctuation marks, such as commas, semi-colons, parenthesis, and so on.
5. Discard multi-word entities that start or end with a stopword.
6. If the tokens in a multi-word entity have been assigned different entity types, assign to the multi-word entity the most frequent type among the tokens composing the multi-word; in case of a tie, choose the most frequent label in the corpus (i.e., `Concept`).
7. Discard entities that are wholly contained within another entity.
8. Discard `same-as` and directed relations from/to a token that is not an entity or part of an entity.
9. Discard reflexive relations, which might have arisen during the generation of multi-word annotations.

### 2.3. Training setup

The system has been implemented in Python 3.7 with HuggingFace's transformers library [8] (https://github.com/huggingface/transformers). 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 (henceforth, mBERT; https://github.com/google-research/bert/blob/master/multilingual.md) and BETO [9], a BERT model pre-trained on Spanish text. We did not perform any in-domain 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 data provided for the task, consisting of 800 training sentences and 200 development sentences.

The training of the different variants was carried out on 2 Nvidia GeForce RTX 2080 GPUs with ~11GB of memory. The model requires a considerable amount of memory for training, so the batch size was adjusted to 2, while the sequence length was adjusted to ~100 tokens (the maximum length encountered in the training set after BERT WordPiece tokenization, which varies from mBERT to BETO). We applied the AdamW optimiser [10] with a base learning rate of $2^{-5}$, combined with a linear LR scheduling to warm-up the learning rate during the first 5,000 training steps. The dropout probability was arbitrarily set to 0.2 across the whole network.

The training monitored the F1-score of several of the classifiers in the development set and it was 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 balance of development metrics, which for BETO was at the epoch 148 and for mBERT was at the epoch 262.

## 3. Results

Vicomtech participated in all the scenarios, submitting the same three runs:

- Run 1: mBERT + *inferencer 1*
- Run 2: BETO + *inferencer 1*
- Run 3: BETO + *inferencer 2*

The results for each scenario and run are shown in Table 1. We provide the results on the development data and the officially published results on the training data. In addition, the best results obtained among all the participants in the challenge are also included per scenario for benchmarking purposes.

**Table 1**
Results of the submitted runs and the best system in each scenario

| | Development | | | Testing | | |
|---|---|---|---|---|---|---|
| | P | R | F1 | P | R | F1 |
| *Scenario 1 - Main* | | | | | | |
| Run 1: mBERT + inf 1 | 72.32 | 67.74 | 69.95 | 66.78 | **65.23** | 66.00 |
| Run 2: BETO + inf 1 | 73.98 | **71.00** | 72.46 | 66.26 | 65.09 | 65.67 |
| Run 3: BETO + inf 2 (best) | **74.39** | 70.81 | **72.55** | **67.94** | **65.23** | **66.56** |
| *Scenario 2 - Task A (entity recognition and classification)* | | | | | | |
| Run 1: mBERT + inf 1 | 84.69 | 85.40 | 85.04 | 82.16 | **82.01** | 82.09 |
| Run 2: BETO + inf 1 | 86.12 | 86.78 | 86.45 | 81.95 | 81.65 | 81.80 |
| Run 3: BETO + inf 2 | 85.97 | 87.09 | 86.52 | 82.16 | **82.01** | 82.09 |
| SINAI [11] (best) | **87.06** | **87.39** | **87.23** | **84.46** | 80.67 | **82.52** |
| *Scenario 3 - Task B (relation extraction)* | | | | | | |
| Run 1: mBERT + inf 1 | 67.78 | 54.15 | 60.20 | 64.86 | 50.77 | 56.96 |
| Run 2: BETO + inf 1 | 69.50 | 59.05 | 63.85 | 65.61 | 51.73 | 57.85 |
| Run 3: BETO + inf 2 | **71.54** | 58.47 | 64.35 | **67.17** | 51.54 | 58.32 |
| IXA-NER-RE [12] (best) | 70.67 | **67.44** | **69.02** | 64.79 | **61.92** | **63.32** |
| *Scenario 4 - Transfer* | | | | | | |
| Run 1: mBERT + inf 1 | - | - | - | 58.45 | 52.18 | 55.14 |
| Run 2: BETO + inf 1 | - | - | - | 57.94 | 53.05 | 55.39 |
| Run 3: BETO + inf 2 | - | - | - | 59.40 | 53.55 | 56.33 |
| Talp-UPC [13] (best) | - | - | - | **60.47** | **56.38** | **58.35** |
| *Scenario 4 - Transfer, unofficial* | | | | | | |
| BETO + inf 2 + domain FT | - | - | - | 68.42 | 54.47 | 60.65 |

Furthermore, after submitting our predictions we learned that the the task organisers had provided 100 out-of-domain sentences to fine-tune systems for the *Transfer scenario*. Our official results for this scenario did not involve any kind of fine-tuning for the new domain, and thus rely exclusively on zero-shot transfer-learning. For the sake of completeness, we have fine-tuned our best performing model with the provided extra sentences and report the results at the bottom of the results table.

As Table 1 shows, our approach has achieved the best scores of the challenge in the *Main* scenario, obtaining the best balance between *Task A*—entity recognition and classification—and *Task B*—relation extraction—, despite being surpassed by other participants in the individual tasks. The proposed approach yields both better precision and recall metrics, improving the second best system by more than 2 F1-score points. In *Task A*, our system is in second position, having improved the recall of the winner system (82.01 vs 80.67) but not its precision (82.16 vs 84.46). As for *Task B*, our approach yields remarkably lower recall scores than the best system (51.73 vs 61.92), but manages to win third place in the scenario with the best precision (67.17). Finally, our system has won third place in the *Transfer* task, despite not having been fine-tuned with the data available for the new domain. Our system would have achieved the best F1-score had the extra 100 sentences been used, as the last row in Table 1 shows.

Regarding the differences between the submitted runs, little difference is observed. BETO seems to

be a slightly better choice than mBERT in all the scenarios. However, the variation of these differences in the development and the test set suggests that the observed differences may not be statistically significant, specially due to the limited size of the datasets. As for the differences between *inferencer 1* and *2*, the latter seems to help improve the scores for the relation detection. Specifically, taking into account incoming and outgoing arcs appears to help produce more precise predictions by dropping mostly false positive predictions in comparison to *inferencer 1*.

## 4. Discussion

Due to time constraints, the presented model is the result of many arbitrary design choices and contains arguable components that may require further research and experimentation. To enumerate some of the potential flaws:

- The intermediate custom DistilBERT model is an addition based on intuition. We have not performed enough experiments to prove it useful. Further, it implies a non-negligible amount of extra computational and memory requirements.
- It is not clear whether modelling relations as outgoing and incoming arcs helps improve the results. We have not experimented with other representation variations to gather enough evidence to reach a conclusion in this regard.
- The directed relations are detected in two steps: *1)* whether a relation exists or not, and *2)* the type of relation. This can been done in a single step. We have not made experiments to know which approach yields better results.

All in all, our system appears to be a good entity recogniser with the capability to produce quite precise relations between the entities—while missing almost half them—, and to be suitable for transfer learning scenarios. The joint modelling of both entities and relations has allowed the system to achieve a good balance between *Task A* and *B*, but the system does not excel in any of them individually. The presence of a pre-trained BERT model helps in the domain transfer scenario. Since the results obtained suggest that specific pre-training on Spanish text (i.e., BETO) achieves better scores, additional pre-training on more relevant data would probably help improve the results.

On the whole, the task is far from being solved, in particular for relation extraction, despite the reasonably good results obtained. We leave the issues and open questions discussed to future work.

## 5. Conclusions

In this working notes we have described our participation in the eHealth-KD 2020 shared task. We have presented the end-to-end deep-learning-based architecture of our system, which relies on pre-trained BERT models as the base for semantic representation of the texts, and jointly models the entities and relations proposed in the competition. We have described our data representation, which allows to model discontinuous and overlapping entities in an integrated manner. We also explained how we interpret and post-process the output of the neural network. The proposed system has won the competition, achieving the first place in the *Main scenario* and ranking within the top-3 in the other three scenarios. Still, further experimentation is required to understand the impact of the network's components and how to improve them, which we will explore in future work.

## Acknowledgments

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

## References

[1] A. Piad-Morffis, Y. Gutiérrez, H. Cañizares-Diaz, S. Estevez-Velarde, Y. Almeida-Cruz, R. Muñoz, A. Montoyo, Overview of the eHealth Knowledge Discovery Challenge at IberLEF 2020, in: Proceedings of the Iberian Languages Evaluation Forum co-located with 36th Conference of the Spanish Society for Natural Language Processing, IberLEF@SEPLN 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] V. Sanh, L. Debut, J. Chaumond, T. Wolf, DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter, in: Proceedings of the 5th Workshop on Energy Efficient Machine Learning and Cognitive Computing (EMC2) co-located with the Thirty-third Conference on Neural Information Processing Systems (NeurIPS 2019), 2019, pp. 1–5.

[5] D. Misra, Mish: A Self Regularized Non-Monotonic Neural Activation Function, arXiv:1908.08681 (2019) 1–13.

[6] P. Stenetorp, S. Pyysalo, G. Topić, T. Ohta, S. Ananiadou, J. Tsujii, BRAT: A Web-based Tool for NLP-assisted Text Annotation, in: Proceedings of the Demonstrations at the 13th Conference of the European Chapter of the Association for Computational Linguistics (EACL '12), 2012, pp. 102–107.

[7] Y. Wu, M. Schuster, Z. Chen, Q. V. Le, M. Norouzi, W. Macherey, M. Krikun, Y. Cao, Q. Gao, K. Macherey, et al., Google's neural machine translation system: Bridging the gap between human and machine translation, arXiv:1609.08144 (2016) 1–23.

[8] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, J. Brew, HuggingFace's Transformers: State-of-the-art Natural Language Processing, arXiv:1910.03771 (2019) 1–11.

[9] 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.

[10] I. Loshchilov, F. Hutter, Decoupled Weight Decay Regularization, in: Proceedings of the Seventh International Conference on Learning Representations (ICLR 2019), 2019, pp. 1–18.

[11] P. López-Ubeda, J. M. Perea-Ortega, D.-G. Manuel C., M. T. Martín-Valdivia, L. A. Ureña-López, SINAI at eHealth-KD Challenge 2020: Combining Word Embeddings for Named Entity Recognition in Spanish Medical Records, in: Proceedings of the Iberian Languages Evaluation Forum co-located with 36th Conference of the Spanish Society for Natural Language Processing, IberLEF@SEPLN 2020, 2020.

[12] E. Andrés, O. Sainz, A. Atutxa, O. Lopez de Lacalle, IXA-NER-RE at eHealth-KD Challenge 2020: Cross-Lingual Transfer Learning for Medical Relation Extraction, in: Proceedings of the Iberian

Languages Evaluation Forum co-located with 36th Conference of the Spanish Society for Natural Language Processing, IberLEF@SEPLN 2020, 2020.

[13] S. Medina, J. Turmo, TALP at eHealth-KD Challenge 2020: Multi-Level Recurrent and Convolutional Neural Networks for Joint Classification of Key-Phrases and Relations , in: Proceedings of the Iberian Languages Evaluation Forum co-located with 36th Conference of the Spanish Society for Natural Language Processing, IberLEF@SEPLN 2020, 2020.