Vicomtech at CANTEMIST 2020

Aitor García-Pablos, Naiara Perez and Montse Cuadros

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 CANTEMIST shared task, consisting in the automatic assignment of ICD-O-3 tumour morphology codes to health-related documents in Spanish language. The submitted systems are based on pre-trained BERT models. The contextual embeddings obtained for each token are used in a multitask sequence-labelling approach that takes advantage of ICD-O-3 code's structure. We have experimented with different pre-trained BERT models and combinations, as well as several ensemble structures. The three task tracks—tumour morphology mention recognition, normalisation and document coding—have been approached at the same time, based on the outputs of the proposed models and some post-processing steps. The reported results are robust and perform well across different subsets of data. The official results also indicate that the ensemble models outperform individual models.

Keywords

Clinical Text Coding, ICD-O-3, Oncology

1. Introduction

These working notes describe Vicomtech's participation in *CANTEMIST: CANcer TExt MIning Shared Task - tumor named entity recognition*. CANTEMIST is the first shared task focused on tumor morphology mining and coding in Spanish text with the International Classification of Diseases for Oncology, 3rd Edition (ICD-O-3). The task consists of three independent tracks:

- NER: finding automatically tumour morphology mentions.
- NORM: NER + assigning to each recognized mention their corresponding ICD-O-3 code.
- CODING: suggesting a ranked list of ICD-O-3 codes per document.

The CANTEMIST gold standard corpus consists of manually annotated clinical cases in BRAT standoff format [1], sourced from the SPACCC corpus[1]. 501 and 500 clinical cases have been made available for training and development purposes, respectively. The development data is split in 2 sets of 250 documents each. The test dataset consists of 300 unlabelled clinical cases that come mixed within a background set of 5,323 documents to difficult manual revision on the predicted labels for the competition. Detailed information about CANTEMIST, including a detailed description of the corpus, the annotation guidelines and evaluation metrics, is provided in the shared task overview article [2] and website[2].

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)

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

CEUR Workshop Proceedings (CEUR-WS.org)

[1] https://github.com/PlanTL-SANIDAD/SPACCC

[2] https://temu.bsc.es/cantemist

The Vicomtech team has submitted multiple systems to all CANTEMIST tracks. The systems have been developed with state-of-the-art deep learning architectures, featuring different BERT-flavoured embeddings [3]. The final submitted systems consist of voting ensemble models.

The paper is organised as follows: Section 2 provides a detailed explanation of the submitted systems' architectures and training setups; Section 3 presents the results obtained in the different task tracks and provides a preliminary error analysis; Section 4 poses several open questions; finally, Section 5 outlines our main conclusions.

2. System Description

The submitted systems are designed primarily to solve the NORM track, i.e., to detect text spans mentioning tumour morphologies and to assign valid ICD-O-3 codes to them. Since the NER track is contained within the NORM track, solving NORM implies solving NER. In addition, we use the ICD-O-3 codes obtained from the NORM track as candidates for the CODING track, after some post-processing. In summary, we address the three CANTEMIST tracks with the same models, which we describe in the following sections.

2.1. Data representation

The CANTEMIST datasets come in BRAT format. This format consists of plain text files paired with annotation files that indicate the character spans of each tumour morphology mention and their corresponding gold ICD-O-3 codes. In what follows, we explain how we transform these datasets to solve the proposed problem.

2.1.1. Document segmentation

The CANTEMIST corpus contains documents longer than 512 tokens, the maximum allowed by BERT. A common fix to perform sequence-labelling tasks on long documents is to define a more granular processing unit, such as sentences. A sentence is likely to fit within 512 tokens, so the task can be performed without cropping any potentially relevant part of an input document. Yet this approach poses several risks: *a)* sentence splitters may introduce errors, *b)* isolated sentences may lack relevant information for the target task, and *c)* unbalanced sentence lengths may lead to an inefficient use of the computational resources.

In an attempt to overcome these problems, we have opted for a sliding-windows approach, depicted broadly in Figure 1: After each document is tokenised with a pre-trained BERT tokeniser, the sequences of subwords are split into windows of a fixed length W. Then, surrounding contexts of size C are appended and prepended, padding as necessary in order to obtain subsequences of size $C + W + C$. Finally, BERT's `[CLS]` and `[SEP]` tokens are added to each subsequence. A mask indicates which sequence positions are part of the window and which ones form the context. Both contexts and window positions are attended to build the BERT contextual embeddings, but the loss function is only calculated for the positions inside the window. We have chosen $W = 300$ and $C = 100$, resulting in sequences of 502 tokens.

2.1.2. Classification objectives

ICD-O-3 morphology codes have a very specific structure [4] (see Figure 2): they consist of at least 5 digits, where the first four digits indicate the tumour or cell type and the fifth digit indicates the

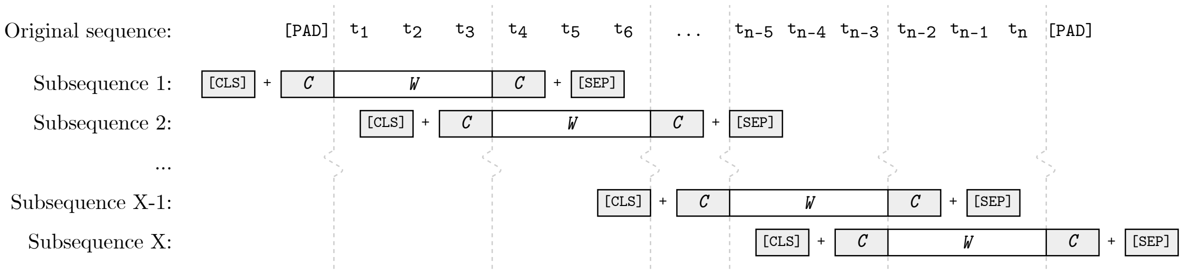

Figure 1: Segmentation of documents into subsequences for BERT with the sliding windows technique

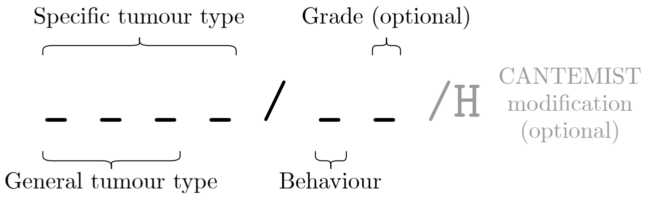

Figure 2: ICD-O-3 code structure

Table 1
Number of different values each code position can take and examples

	Values	Example	
3 digits	189	868 to 871	Paragangliomas and glomus tumours
4^{th} digit	10	8711	Glomus tumour
Behaviour	6	8711/3	Malignant glomus tumour
Grade	9	8711/31	Malignant glomus tumour, differentiated
/H	1	8711/3/H	Malignant glomus tumour, with uncoded modifier

behaviour of the tumour; an optional digit codes histologic grading o differentiation. In addition, CANTEMIST annotators introduced a task-specific code extension: /H. It is used when ICD-O-3 does not offer a code specific enough for the tumour morphology mention being coded.

The current ICD-O-3 version describes 4,205 codes. However, because of its multi-axial nature, new well-formed codes can be composed if necessary following the aforementioned convention. In CANTEMIST, a total number of 58,062 codes are considered valid. Table 1 shows how many different values each code position can take and provides some examples.

Based on these facts, we have approached the task as a multitask sequence-labelling problem. The ICD-O-3 codes have been split into several pieces, each piece comprising a classification objective. After preliminary experimentation, the selected classification objectives are:

a) the first 3 digits of the code,

b) the fourth digit, and

c) the Behaviour and Grade digits, and the H indicator, as a single variable.

If a token is not part of a tumour morphology mention, the label O (from “Out”) should be predicted by the three classifiers. We henceforth refer to them as 3DS, 4D and BGH.

An additional classification objective—since this is, in essence, a sequence-labelling task—is:

d) the BIO tag.

The BIO tag [5] indicates whether a token is the first element of a tumour morphology mention (B-, “Begin”), whether it is inside a mention (I-, “In”), or it is not part of a mention at all (O, “Out”). Although it does not convey ICD-O-3-related information, the BIO tag is an additional signal of whether a token is part of a mention or not, and it helps discern between contiguous mentions.

2.2. Architecture

The submitted systems are built on the Transformers [6] architecture, specifically BERT [3]. In few words, they consist of pre-trained BERT models with several classification layers on top.

We have tested two approaches, one of which is the continuation of the other. We henceforth refer to them as the *baseline* approach and the *two-experts* approach. The latter is an experiment to assess whether two sources of knowledge can be fused to collaborate and improve the results they would obtain on their own. There are different ways of combining two models into a bigger model; in this work, we have chained one after another.

A high-level diagram of the baseline and two-experts approach is shown in Figure 3: Both approaches start by passing the prepared tokens to a BERT model. In the two-experts approach, the output of the last layer is fed to a second BERT model as pre-computed embeddings. The output of the second model’s last layer is then processed by a dropout layer. In the baseline approach, the output of the first model is directly passed to the dropout layer. After dropout, the token representations are passed to 4 independent linear transformation layers, which output the logits for the 4 output variables described earlier (see Section 2.1.2). That is, all the objectives are trained jointly in a single model that has several classification heads. All of them rely on the same per-token contextual embedding obtained from a pre-trained BERT model.

In training, the back-propagated error is the sum of the cross-entropy losses of the 4 outputs. BERT’s special tokens and context tokens do not participate in the computation of the loss. That is, while the BERT models do attend to all positions, they only learn from the gold labels in each sequence’s window, not from its context (see Section 2.1.1). This helps avoid an “edge bias” near the arbitrary start/end of the input.

For inference, the label with the maximum probability is chosen for each token and variable after applying the softmax function to the logits. Then, the outputs of the sliding windows are concatenated, and BERT’s special tokens and context tokens ruled out, in order to obtain the original sequence of tokens and their corresponding predictions. In the case of tokens split into subwords by the tokeniser, the predictions corresponding to the first subword are used as predictions for the whole token.

2.3. Output interpretation

The implemented systems output 4 predictions per token, which correspond to the BIO tag, the first 3 digits of an ICD-O-3 code, the fourth digit, and the Behaviour, Grade and /H positions. This output must be interpreted and transformed to BRAT’s span-based format, where each tumour morphology mention detected, whether a single token or multiple, is associated with a valid ICD-O-3 code. The post-process consists of two main steps:

First, if any of the classifiers 3Ds, 4D or BGH predicted the tag O, O is assigned to the token; it is not part of a tumour morphology mention. Otherwise, an ICD-O-3 code is composed from the predictions, prefixed with the corresponding BIO tag (see examples on the right-hand side of Figure 3). A probability is assigned to the newly created code, defined as the product of the probabilities emitted by the classifiers 3Ds, 4D and BGH.

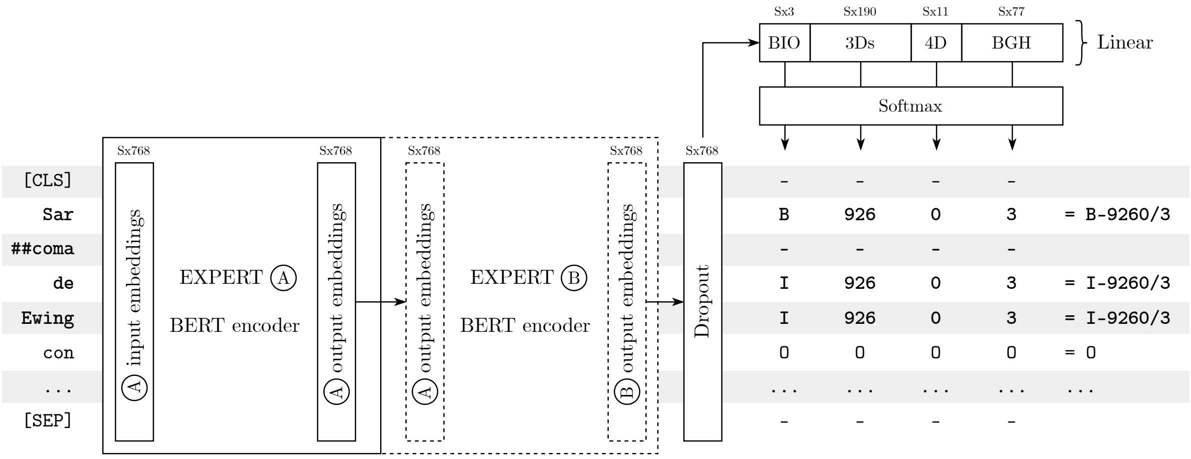

Figure 3: High-level architecture diagram

Then, the token-based annotations are translated to span-based annotations with the help of the BIO tags. In the case of single-word mentions, the ICD-O-3 code assigned to the mention is simply the ICD-O-3 code of the corresponding token. In the case of multi-word mentions, the ICD-O-3 code chosen is the code with maximum average probability among the codes of the tokens that participate in the mention.

As a result, we obtain outputs for the NER and NORM tracks. To produce outputs for the CODING track, where a ranked list of ICD-O-3 codes is expected per document, we simply take the set of predicted codes and order them by their assigned probability.

2.4. Training setup and submitted systems

In the earlier phases of this work, we experimented with several publicly available pre-trained BERT models, namely, BERT-base Multilingual Cased[3], BETO [7], SciBERT [8], Clinical BERT [9], and BioBERT [10]. The latter three have been pre-trained with English text of the health and biomedical domains. The best results on the official development sets were achieved by BETO and SciBERT. Thus, the submitted systems use BETO, SciBERT, or both.

Early experimentation also showed considerable differences in performance between the two development datasets. In order to leverage all the available data, we have trained several systems with different data splits and combined their predictions in voting ensembles.

2.4.1. Voting ensembles

Let D_1 and D_2 be the two official development sets provided by the task organisers. Let D_3 be a third development set randomly sampled from the official training set T, and T_d the remaining data of the training set, so $T = T_d \cup D_3$. We have trained 3 versions of each model, setting aside one development set each time, so for each rotation the training data split is $T_{rotation} = \{T_d \cup D_i \cup D_j\}$ and the development set is $D_{rotation} = \{D_k\}$, with $i, j, k \in \{1, 2, 3\}$ and $i \neq j \neq k$.

The model ensembles are obtained via token-wise soft voting, prior to transforming the standalone predictions to BRAT's character-span-based format: the full ICD-O-3 code for each token and voting system is built from its predicted components as explained in Section 2.3; after, the vote of each system

[3] https://github.com/google-research/bert/blob/master/multilingual.md

is weighted by the probability of the codes, the probability being the product of the probabilities given by the classifiers 3Ds, 4D, and BGH. Finally, the BRAT files and the CODING track outputs are generated as if the predictions came from a single system.

The final submitted systems are the following:

- S1: an ensemble of 3 BETO-based baseline models
- S2: an ensemble of 3 SciBERT-based baseline models
- S3: an ensemble of 3 Two Experts models, with BETO as the first expert and SciBERT the second
- S4: an ensemble of the prior 9 models, henceforth the *Flat* ensemble
- S5: an ensemble of S1, S2 and S3, henceforth the *2-step* ensemble

Both ensembles S4 and S5 take advantage of all the 9 trained standalone models, but the former performs the voting with the 9 outcomes at the same time, while the latter calculates the votes in 2 consecutive rounds: it first calculates S1, S2 and S3, then uses their results to vote a second time.

2.4.2. Hyperparameters and other implementation details

The implementation of the models and all the auxiliary modules, helpers and functions are mainly written in Python 3.7 and the HuggingFace's Transformers library [11].

During training, the base learning rate was 2E-5 with a linear warm-up scheduling that reaches its maximum during the first 5,000 iterations. The training of all models was limited to a maximum of 200 epochs with an early-stopping patience of 50 epochs (i.e., the training was stopped after 50 consecutive epochs without improvement). In most of the cases, the early-stopping was triggered before reaching the maximum allowed epochs. The dropout rate was the same used in the pre-trained BERT-base models: 0.1. The batch size for the baseline models was set to 6, while for the two-experts variants it was set to 4, in both cases because it was the largest possible batch that fit in memory on a single GPU. The training has been run on a single Nvidia RTX 2080 GPU with 11GB of RAM. Training times vary depending on when the early-stopping condition is met, but all of them have fallen within a range of a few hours.

For inference, we have used a much larger batch size of 128, because the memory requirements are lower due to the lack of gradients calculation. The context and window sizes for the sliding-windows have been kept the same: $W = 300$ and $C = 100$. The inference speed in GPU exceeds 8,000 tokens/second[4], which for this task is equivalent to processing about 10 documents per second. With these settings, the 5,323 background documents of the competition have been processed in 7-8 minutes.

3. Results

Table 2 shows the results obtained by the submitted systems for all the tracks. It includes the results we have calculated on the different development sets described in Section 2.4.1. The results for the test set are the official results reported by the task organisers. A comprehensive comparison and ranking of the results from all the shared task participants can be found in [2].

[4] Although training is impractical without a GPU, the inference can be performed on CPU achieving a throughput of about 800 tokens/second.

Table 2
Results per system, track and dataset

		NER			NORM			COD
		P	R	F1	P	R	F1	MAP
Development set 1								
D1.1	BETO	84.32	**83.54**	**83.93**	78.10	**77.37**	**77.73**	81.23
D1.2	SciBERT	**85.38**	82.13	83.72	**79.25**	76.23	77.71	**82.24**
D1.3	Two Experts	83.79	83.42	83.61	75.23	74.89	75.06	78.17
Development set 2								
D2.1	BETO	85.81	**85.45**	**85.63**	**77.77**	**77.44**	**77.60**	78.48
D2.2	SciBERT	84.24	85.00	84.62	76.49	77.18	76.83	**79.47**
D2.3	Two Experts	**86.48**	83.91	85.17	77.14	74.85	75.98	76.77
Development set 3								
D3.1	BETO	86.49	**85.39**	**85.94**	**79.24**	78.24	**78.74**	80.59
D3.2	SciBERT	84.35	85.30	84.82	77.94	**78.81**	78.37	**81.34**
D3.3	Two Experts	**86.58**	84.87	85.72	78.57	77.02	77.79	78.66
Test set								
S1	BETO ensemble (D1.1 + D2.1 + D3.1)	86.29	86.62	86.46	80.74	80.76	80.75	82.91
S2	SciBERT ensemble (D1.2 + D2.2 + D3.2)	85.45	86.65	86.05	80.13	81.15	80.63	83.84
S3	Two Experts ensemble (D1.3 + D2.3 + D3.3)	86.29	86.13	86.21	79.81	79.55	79.68	81.49
S4	Flat ensemble (all 9 D*X*.*X*)	**86.92**	86.54	86.73	82.16	81.92	82.04	84.21
S5	2-step ensemble (S1 + S2 + S3)	86.83	**87.12**	**86.97**	**82.19**	**82.08**	**82.14**	**84.68**

The results obtained by our models vary among the development sets, but are quite consistent. The results for the test set are higher, probably due to the effect of the ensembles. Precision and recall are evenly balanced for all the tested systems.

The best performing system, the 2-step ensemble, obtains 86.97 F1-score in NER, 82.14 F1-score in NORM and 84.68 Mean Average Precision (MAP) in CODING. Overall, the systems surpass the scores 83.00, 75.00 and 76.00, respectively, by a large margin. The ensembles of the 9 model variants—3 per model type—work noticeably better than the ensembles of a single model type. The 2-step ensemble works even better than the flat ensemble for all the tracks, in particular for CODING, where the difference between the flat and 2-step ensemble is almost 0.5 MAP points.

With respect to BETO and SciBERT, the former performs marginally better in NER and NORM; however, it obtains consistently better MAP in CODING. The two-experts approach has not resulted in a performance improvement.

A quantitative analysis of the errors committed by the submitted systems is provided in Table 3. Again, we observe similar trends among the systems. SciBERT seems to yield more annotations—spurious and correct—than BETO; Two Experts produces less annotations than BETO or SciBERT alone. Meanwhile, the flat and 2-step ensembles miss less annotations, make less spurious predictions, and produce more exact matches.

In general, when a mention span is matched exactly—which happens ~80% of the times on average—, the code given is likely to be correct with >93% probability. The chances drop to around 35% with overlapping spans. In whichever case, the error is more likely to be found in the first four digits of the code than in the behaviour (B), grading (G) or /H position when an incorrect code is proposed.

Table 3
NORM error analysis on the test set

	S1	S2	S3	S4	S5
Total predictions	3,634	3,679	3,621	3,622	3,628
Exact span matches	3,141	3,142	3,131	3,149	3,150
of which,					
exact code matches	2,933	2,948	2,889	2,975	2,981
3Ds errors	86	77	116	65	62
4D errors	77	67	93	59	59
B errors	47	48	51	37	33
G errors	23	24	23	16	21
H errors	49	48	48	52	49
code not in train, guessed correctly	9	15	5	17	15
code not in train, guessed incorrectly	66	55	67	58	49
Span overlaps	335	348	326	318	327
of which,					
exact code matches	120	126	116	111	115
3Ds errors	97	116	104	98	110
4D errors	65	77	76	64	72
B errors	73	74	72	66	70
G errors	47	52	43	51	48
H errors	28	31	28	24	28
code not in train, guessed correctly	1	2	1	2	4
code not in train, guessed incorrectly	68	83	72	81	82
Spurious predictions	158	189	164	155	151
Missed mentions	218	204	233	219	216
of which,					
code not in train	18	14	17	19	18

While 58,062 codes are considered valid in CANTEMIST, only 746 of them actually occur in the training and development data provided. Our systems are capable, to an extent, of producing ICD-O-3 codes that they have not seen in the training data. This is possible on account of the multi-task approach. Still, our systems fail to generate correct unseen codes much more often than they succeed, even more so when the mention span has not been matched exactly.

The bulk of missed mentions are not mentions pertaining to unseen codes, but mentions that do occur and are even very frequent in the training and development datasets. This phenomenon requires further analysis to be better explained and addressed.

4. Discussion

The systems presented rely mainly on the semantic representation capabilities derived from the BERT architecture and the knowledge captured by their own pre-training. The results are seemingly good—

other participants' results are unknown to us at the time of writing—, but there is still room for improvement. We pose the following open questions as discussion:

1. Our approach does not leverage information associated to ICD-O-3 codes (code descriptions, definitions, and so on) nor any other hand-crafted knowledge source, which could improve the results obtained by helping produce representations for ICD-O-3 codes not seen in the training data.

2. Regarding BioBERT, Clinical BERT and SciBERT: we hypothesise that SciBERT has outperformed BioBERT and Clinical BERT in our experiments because it has been trained from scratch with its own vocabulary, better suited to the health domain.

3. In the same line, it may come as a surprise that BETO and SciBERT obtain similar results, when SciBERT has only been trained with texts in English. We hypothesise that because the terminology of the health domain is mainly constructed from Greek and Latin roots and affixes both in English and in Spanish, the WordPiece strategy and the domain-specific vocabulary of SciBERT play to its advantage in this case.

4. The two previous points indicate that a Spanish Clinical BERT may lead to better results still.

5. The combination of BETO and SciBERT, in the manner explained in this paper, does not seem to be beneficial in this task, having obtained slightly worse results than the standalone models. Many other ways exist in which the two models could be combined, so further experimentation in this line might be of interest.

6. While the flat and 2-step ensemble models show performance gains in comparison to the simpler models, it is questionable whether such a system would be viable in a real-world scenario.

5. Conclusions

In this working notes we have described our participation in the CANTEMIST shared task. Our end-to-end deep-learning-based system relies on pre-trained BERT models as the base for semantic representation of the texts. With these semantic representation, the ICD-O-3 codes are calculated for each token in a sequence-labelling fashion, and this information is used to address the three competition tracks (namely, NERC, NORM and CODING) at the same time. We have described how we have preprocessed and represented the information, and how we have performed rotating training runs to leverage all the available data (i.e., the official training set and the two official development sets). We have submitted results of ensemble models trained on different views of the data.

Both our experiments and the official evaluation show robust results in different subsets of data. According to these results, the ensembles do provide a performance advantage, with a two-step ensemble outperforming a flat ensemble. We have also found that BETO and SciBERT obtain comparable results in this particular task, but the proposed combination of both has not resulted in better scores.

As future work, the models may benefit from a mechanism to inject ICD-O-3 codes semantics to enhance their capability to match codes that have not been seen during the training phase. Further experimentation on the combination of several pre-trained models would also be helpful for scenarios where each model brings some useful knowledge to the task, and there is not a single pre-trained model that suits the task better.

Acknowledgments

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

References

[1] 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.

[2] A. Miranda-Escalada, E. Farré, M. Krallinger, Named entity recognition, concept normalization and clinical coding: Overview of the CANTEMIST track for cancer text mining in Spanish, Corpus, Guidelines, Methods and Results, in: Proceedings of the Iberian Languages Evaluation Forum (IberLEF 2020), CEUR Workshop Proceedings, 2020.

[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] Word Health Organization, International Classification of Diseases for Oncology, 3rd Edition (ICD-O-3), 2015. URL: https://www.who.int/classifications/icd/adaptations/oncology/en/, accessed: 24-07-2020.

[5] L. Ramshaw, M. P. Marcus, Text Chunking Using Transformation-based Learning, in: S. Armstrong, K. Church, P. Isabelle, S. Manzi, E. Tzoukermann, D. Yarowsky (Eds.), Natural Language Processing Using Very Large Corpora, Springer Netherlands, 1999, pp. 157–176.

[6] 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.

[7] 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.

[8] I. Beltagy, K. Lo, A. Cohan, SciBERT: A Pretrained Language Model for Scientific Text, in: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), 2019, pp. 3615–3620.

[9] E. Alsentzer, J. Murphy, W. Boag, W.-H. Weng, D. Jin, T. Naumann, M. McDermott, Publicly Available Clinical BERT Embeddings, in: Proceedings of the 2nd Clinical Natural Language Processing Workshop (ClinicalNLP 2019), 2019, pp. 72–78.

[10] J. Lee, W. Yoon, S. Kim, D. Kim, S. Kim, C. H. So, J. Kang, BioBERT: a pre-trained biomedical language representation model for biomedical text mining, Bioinformatics 36 (2019) 1234–1240.

[11] 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.