Leveraging Events Sub-Categories for Violent-Events Detection in Social Media

Daniel Vallejo-Aldana[1,2,*], Adrián Pastor López-Monroy[2] and Esaú Villatoro-Tello[3,4]

[1]Department of Mathematics, University of Guanajuato, Guanajuato, Mexico
[2]Department of Computer Science, Mathematics Research Center (CIMAT), Guanajuato, Mexico
[3]Universidad Autónoma Metropolitana, Unidad Cuajimalpa, Mexico City, Mexico
[4]Idiap Research Institute, Martigny, Switzerland

Abstract

This paper describes our participation in the shared evaluation campaign of DA-VINCIS@IberLEF 2022. In this work, we addressed the Violent Event Identification (VEI) task by exploiting Bidirectional Encoder Representations from Transformers (BERT) in combination with Multi-Task learning approaches. Our results indicate that the proposed architecture is able to leverage information about the crime categories for effectively detect the mention of a violent act within a tweet. Our approach obtained the best performance ($F1 = 0.7758$) among 11 different teams and a total of 32 different submissions.

Keywords

Violence Detection, Social Media, Contextual Embeddings, Multi-Task Learning, Natural Language Processing

1. Introduction

Nowadays, social networking has become a major part of humans' lives in all the aspects including politics, education, health, religion, leisure, decision making etc. Such tools allow users to express their thoughts freely and to share information about a variety of topics [1, 2, 3].

Twitter, ranked as the 15th most popular social network[1], has become an extremely important source of real-time information about a massive number of topics, ranging from a trivial note (where someone went last night for a beer) to more overwhelming news (Russian invasion of Ukraine). Thus, detecting events of interest represents an important step to impact the economics and the security of the communities sharing information on this type of social network.

The DA-VINCIS@IberLEF 2022[2] track poses the task of using social media message streams (in Spanish) for violent events detection and categorization [4]. Such task poses a number of

IberLEF 2022, September 2022, A Coruña, Spain.

*Corresponding author.
daniel.vallejo@cimat.mx (D. Vallejo-Aldana); pastor.lopez@cimat.mx (A. P. López-Monroy); esau.villatoro@idiap.ch (E. Villatoro-Tello)
https://github.com/danielvallejo237/ (D. Vallejo-Aldana)

[1]https://buffer.com/library/social-media-sites/
[2]https://sites.google.com/view/davincis-iberlef/home?authuser=0

opportunities and challenges as these streams are frequently: high in volume, contain either duplicated, incomplete, imprecise or incorrect information, are written in informal style, and might have unclear boundaries between different categories. For example, one tweet referring to the same event (e.g., a car accident) might be categorized with different labels, "Accident", "Theft" or "Kidnapping", as perhaps some users are reporting secondary events associated to the car accident.

In this paper, we describe our methodology to approach the Violent Event Identification (VEI) and the Violent Event Categorization (VEC) shared tasks in Spanish Tweets. We addressed the VEI and VEC tasks by means of exploiting Bidirectional Encoder Representations from Transformers (BERT) in combination with a Multi-Task Learning strategy in order to improve the performance of the model when there are highly imbalanced categories. Our proposed approach obtained the best performance in the VEI task ($F1 = 0.7758$) among 11 different teams.

The rest of the paper is organized as follows. Section 2 describes the purpose of the shared tasks as well as provides some statistics from the provided dataset. Section 3 explains the main components of our proposed solution. Section 4 details the data preparation steps, as well as how the main parameters of the proposed architecture were defined. In Section 5 we discuss the obtained results, and finally, in Section 6 we depict our main conclusions and future work directions.

2. Task Description and Data

The DA-VINCIS@IberLEF 2022 shared task is composed of two subtasks:(1) violent event detection, i.e., determine whether a given tweet is associated with a violent incident or not (a binary classification problem); and (2) violent event categorization, i.e., recognize the crime sub-type (*accident, homicide, non-violent, robbery, kidnapping*) to which a given tweet belongs (a multi-class classification problem)[4].

The provided dataset consists of 3412 labeled tweets. From these, 3362 belong to the training partition, while only 50 tweets to the validation set. In order to test and evaluate our model with more data, during experimentation we merged all the provided tweets and create our own *train-dev* partitions. In concrete, we did a stratified partition preserving 20% of the data for validation purposes, and the 80% for training. During the test/submission stage, the training of our final model was done using the original training data partition as provided by the task organizers.

Figure 1 shows the data distribution of the different categories present in the dataset. Accordingly, numbers *1...6* in the x-axis represent categories: *violent-incident, accident, homicide, non-violent, robbery,* and *kidnapping* respectively. From this figure we can clearly notice the imbalance present for most of the distinct violent event categories.

3. Methodology

Inspired by the work described in [5] and [6], our participation in the DA-VINCIS shared task consisted in an adaptation of the methods described in the former papers. We incorporate a residual connection and layer normalization blocks, which as stated in [7], contribute to

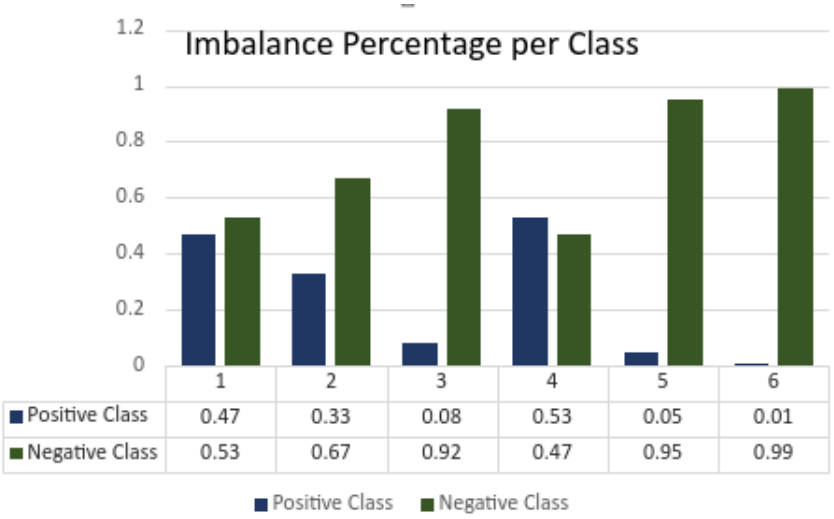

Figure 1: Classes' distribution: 1-violent-incident, 2–accident, 3–homicide, 4–non-violent, 5–robbery, 6–kidnapping

the internal representations of the transformer-based attention mechanism. Additionally, our proposed solution consists in tackling the learning task by means of a Multi-Task Learning (MTL) approach. The goal of the MTL is to generalize better on our main task by sharing representations between related tasks [8].

Figure 2 depicts the overall configuration of the main component of our proposed solution. Notice that the first step of the architecture is the encoding of the tweets by means of a BERT-based approach [9]. Particularly, we used a pre-trained encoder model from the Huggingface website, which is explained into more detail in [5]. For our performed experiments we use the uncased version of the encoder to not constraint it to a specific task. As known, the BERT-based encoder returns the encoded tweet through the [CLS] token (a 768 dimensionality vector) as well as the attention mask, which are then fed into the next block. The [CLS] vector is combined (i.e, summed) with a residual output. The residual block for this stage consists of three linear layers with activation function ReLU and Batch normalization (See Figure 3). A detailed explanation of the advantages of adding residuals in language models can be found in [7].

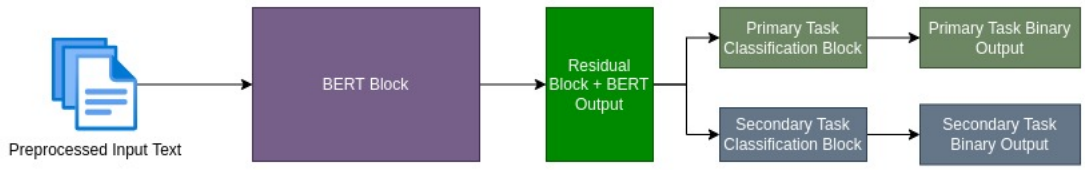

Figure 2: Multi-task learning architecture used for our performed experiments in the VEI and VEC shared tasks.

Once the residuals are obtained, this vector (also dimensionality 768) serves as the input to each one of the multi-task learning heads. Specifically, we used the the hard parameter sharing

strategy for the multi-task learning blocks, as proposed in [8]. Examples where such learning strategy has been successfully used in Natural Language Processing tasks can be found in [10, 11]. Intuitively speaking, this approach it is generally applied by sharing the hidden layers between all tasks, while keeping several task-specific output layers. As mention in [12], hard parameter sharing greatly reduces the risk of overfitting among classes. Thus, in our proposed architecture, each head for classification consist of three linear layers with ReLU functions in between as activation functions. At the end of the classification heads, we add a softmax layer for the logits estimation.

Figure 3: Graphical representation of the residual block module, containing three linear layers with activation function ReLU and batch normalization.

3.1. Weighted Loss Function

To handle the imbalance problem present in the dataset (see Figure 1), we used the weighted cross entropy loss with mean reduction as our loss function. The formula of the corresponding loss function is described below.

$$L(x, y) = \sum_{i=1}^{N} \frac{1}{\sum_{i=1}^{N} w_{y_i} 1\{y_i \neq ignoreIndex\}} l_i$$

where l_i corresponds to

$$l_i = -w_{y_i} \log \frac{\exp(x_{i,y_i})}{\sum_{c=1}^{C} \exp(x_{n,c})} 1\{y_i \neq ignoreIndex\}$$

and w_{y_i} corresponds to the weight associated with the label y_i.

To estimate the weights w_{y_i} for the positive and the negative class, we considered the imbalance percentage of each class into account (see Figure 1). Thus, we proposed the following expression to estimate the corresponding weights. Let S be the space of text inputs and let W^C_{pos} and W^C_{neg} two disjoint sets such that $W^C_{pos} \cup W^C_{neg} = Y^C$, where Y^C correspond to the set consisting of all labels associated to S for the task C. Let p^C be defined as:

$$p^C = \frac{|W^C_{pos}|}{|W^C_{neg}|}$$

Then the weight associated to a label y^C_i called $w^C y_i$ is defined as follows.

$$w^C y_i = \begin{cases} 1 - p^C \text{ if } y^C_i = 1 \\ p^C \text{ if } y^C_i = 0 \end{cases}$$

After applying the previous formulation, we obtained the weights shown in Table 1. These represent the final weights used for the positive and negative class for each of the subtasks (i.e., violent events categories) present in the posed task.

Task	$w_{negative}$	$w_{positive}$
Violent Event Detection	0.5	0.51
Accident	0.34	0.66
Homicide	0.07	0.93
Non-Violent-Incident	0.53	0.47
Robbery	0.05	0.95
Kidnapping	0.01	0.99

Table 1
Weights used for each loss function to handle with imbalanced classes in the dataset

4. Experimental Setup

In this section we provide further details about the pre-processing steps applied to the data, optimization functions, learning rates, etc.

4.1. Preprocessing Steps

As main pre-rpocessing steps we follow some of the strategies described in [5]. First, all tweets are lower-cased. This reduces the size of the vocabulary present in the data, i.e., avoids repetitions that can lead to unnecessary confusion in the encoding process.

All the emojis, URL's, user tagging (@'s) and hashtags (#'s) are considered as special cases as they might contain relevant information for detecting important events in the tweets. URL's are replaced by the `<url>` label. Similarly, user tagging (@'s) are replaced by the label `@user`. Hashtags (#'s) remain unchanged.

For processing the emojis, we use the python library `emoji`[3]. Thus, we convert every emojis into its corresponding word representation. We then change the character `:`, that surrounds the word meaning of the emoji, for the actual word `emoji`.

Finally, all the special characters are removed. Examples of such characters are exclamation marks, question marks and underscores. Every sequence of consecutive spaces are reduced to one. This prevents the encoder from splitting the sentence hastily.

4.2. Training Parameters

As mentioned, we followed a multi-task learning strategy to leverage the performance of model classification in highly imbalanced classes. However, our experiments indicate that this strategy also leverages the performance of the model in balanced classes, which is the case of the Violent-Event category.

For our performed experiments we used the AdamW optimizer [13] with a weight decay of 0.24. We set a starting learning rate of 1×10^{-5} and an ending learning rate of 3.5×10^{-6}. The latter corresponds to the learning rate suggested for fine-tuning a BERT-based model [9]. The learning rate decay uses a cosine weight decay with hard resets to simulate a local search for parameter tuning. The weights used in the loss function (see Section 3.1) during the multi-tasking learning was empirically determined. The final weights are shown in Table 2

Loss Weights			
Task 1	**Task 2**	w_{Task1}	w_{Task2}
Violent-Event-Detection	Accident	0.7	0.3
	Homicide	0.9	0.1
	Non-Violent-Incident	0.6	0.4
	Robbery	0.9	0.1
	Kidnapping	0.9	0.1

Table 2
Weights of each of the two losses used for the multitasking stage

We trained our proposed architecture for 20 epochs, taking an estimate of five minutes, on an RTX 2080 Ti ©NVIDIA graphic card.

[3] https://pypi.org/project/emoji/

4.3. Ensemble Model

Motivated by the findings described in [6], we evaluate the performance of an assemble approach. Thus, we use three different models trained using a multi-task learning approach (as described in Section 3), with a different secondary task for each. We then make predictions of the *Violent-Event* detection task via a binary voting scheme. The final prediction is the one that is submitted for evaluation. Figure 4 depicts the configuration of the assemble approach. As explained in Section 5, selection of the secondary task for each of the individual models was done considering their performance independently.

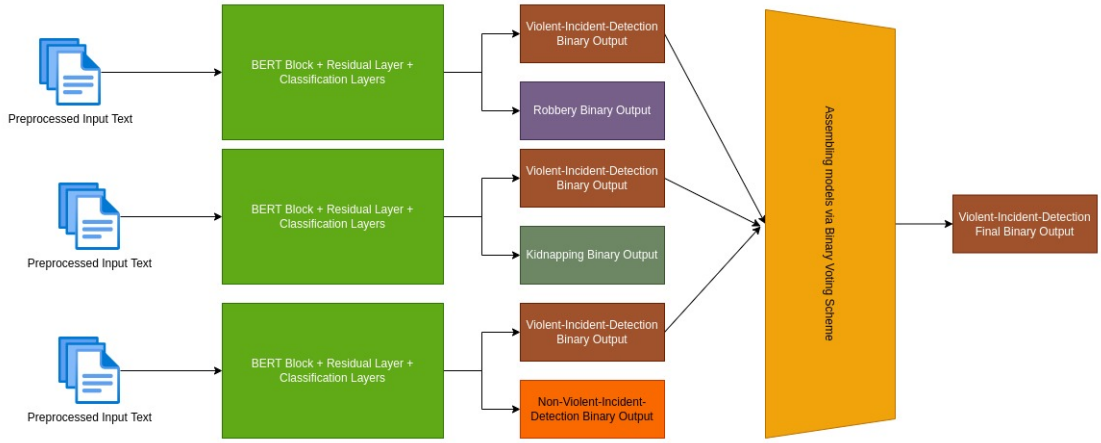

Figure 4: Final architecture of the strategy followed to win the IberLEF 2022 DaVinci's competition

5. Results

In this section, we describe our results obtained during our experimental stage as well as the final (official) submissions made to the DA-VINCIS challenge. Section 5.1 and Section 5.3 describe the experimental results obtained in our stratified version of the data for the VEI and VEC shared tasks respectively. Section 5.2 and Section 5.4 explain the official submissions made to the DA-VICIS VEI and VEC shared tasks respectively. Finally, Section 5.5 discuss a few examples of the errors made by our model.

5.1. VEI Experimental Results

Overall, we performed seven different experiments. For **EXP1** we did not considered a boosting task, i.e., we evaluate the model depicted in Figure 2 without the "secondary task" blocks. From **EXP2** to **EXP6** we evaluate the model shown in Figure 2 considering as the secondary task the categories *accident, homicide, non-violent-event, robbery* and *kidnapping* respectively. Finally, for **EXP7** we evaluate the performance of the assembling model (Figure 4) where the considered secondary task for each sub-model was chosen based on the performance of the individual MTL blocks, specifically *robbery, kidnapping and non-violent-event* categories.

As explained in Section 2, for performing our experiments we did a stratified partition of the original dataset (80%- *train*, 20%-*val*). The metric used for evaluation is the $F1-$score of the positive class. Table 3 shows the obtained results in our validation set.

Results over the stratified partition			
Experiment	**Main Task**	**Boosting Task**	**F1-Score**
EXP1	Violent-Event-Detection	—	0.7591
EXP2		Accident	0.7643
EXP3		Homicide	0.7632
EXP4		Non-Violent-Incident	**0.7731**
EXP5		Robbery	**0.7754**
EXP6		Kidnapping	**0.7798**
EXP7		Assembly Model	**0.7878**

Table 3
Table showing the performance of the F1-Score

We can notice from Table 3, that the best results in terms of $F1$-score were obtained using the classes **Non-Violent-Event**, **Robbery** and **Kidnapping** as boosting classes. Considering these results as a good indicator, we selected these models to generate the assembly model (EXP7). As can be seen, the assemble model was able to outperform all the proposed configurations, reaching an $F1 = 0.7878$.

5.2. Official submissions to the VEI shared task (Subtask 1)

We did two official submissions for the sub-task *violent event identification* (VEI): **RUN1** using the assembly model as described in the previous section, and , **RUN2** the best individual MLT model, in this case the one with the *kidnapping* category as secondary task. As mentioned before, for the training of the final models, we used the entire dataset. Table 4 shows the official results obtained by our two different submissions.

DaVinci's results				
Run	**Model**	**F1-Score**	**Recall**	**Precision**
RUN1	Assembly model	**0.7759**	**0.7503**	**0.8032**
RUN2	Kidnapping class as boosting class	0.7658	0.7318	**0.8032**
-	*2nd best team*	0.7732	0.7373	0.8128
-	*3rd best team*	0.7651	0.7515	0.7792
-	*Average performance*	0.7496	0.7385	0.7639
-	*Baseline*	0.7633	0.78	0.75

Table 4
Official results reported by the DaVinci's challenge organizers.

As can be observed, our RUN1 (i.e., assembly model) obtained the best performance among the 11 different teams (32 different submissions). On the other hand, out RUN 2 was ranked 6th place, nevertheless, this obtained a better performance than the average performance of the 32 submissions, and obtained a better performance that the 3rd best team. The baseline model corresponds to a single-task fine-tuned BETO [14] trained for 20 epochs.

5.3. VEC Experimental Results

Fo the *violent event categorization* (VEC) task, we performed 5 different experiments. Similarly as for the VEI experiments described in Section 5.1, we evaluate the model as shown in Figure 2 where the secondary task is one of the different categories, i.e., *accident, homicide, non-violent, robbery* and *kidnapping*. However, contrary to the VEI experiments, here we evaluate the performance of the model over the secondary task. Obtained results are shown in Table 5.

Results over Violent Incident Classification			
Secondary Task	**F1-Score**	**Recall**	**Precision**
Accident	0.79	0.83	0.76
Homicide	0.33	0.78	0.21
Non-Violent	0.79	0.83	0.75
Robbery	0.17	0.60	0.10
Kidnapping	0.09	0.82	0.05

Table 5
Experimental results of the proposed model for the VEC task.

Notice that contrary to the VEI task, for the VEC classification problem, the best performance is obtained for the less unbalanced categories, i.e., *accident, non-violent* and *homicide*. Accordingly, the *robbery* and *kidnapping* categories are the most difficult type of events to classify.

5.4. Official submissions to the VEC shared task (Subtask 2)

We only submitted one experiment for the VEC (substask 2). Generally speaking, for generating the final predictions of the test set, we combine the predictions of the different models trained for each of the different event categories. The official results are shown in Table 6

DaVinci's results				
Run	**Model**	**F1-Score**	**Recall**	**Precision**
RUN1	Submitted Model	0.4733	0.47	0.47
-	*1st best team*	**0.5543**	**0.55**	**0.55**
-	*2nd best team*	0.5286	0.53	0.53
-	*Average performance*	0.464	0.4765	0.4817
-	*Baseline*	0.4981	0.46	0.57

Table 6
Official results reported by the DaVinci's challenge organizers.

Notice that our model obtained a lower performance compared to the baseline. However, a similar performance compared to the average result reported by all the participants. It is worth mentioning that for these experiments, the model was not tuned to solve the secondary task, instead, we only took the predictions of the secondary task. We hypothesize that the performance will improve if we modify the primary objective of the model.

5.5. Error Analysis

In this section, we present examples miss-classified by our model. The example below corresponds to a false positive. Our model determines that this tweet contains a violent incident when according to the official labeling, it does not.

SigAlert en Lake Elsinore. En la I-15 norte cerca de Lake St. Los carriles # 2 y # 3 están cerrados por una duración desconocida debido a un accidente de tráfico. https://t.co/i8muQMroSq

Table 7
False positive example: Tweet classified as containing a violent event.

We argue that this miss-classification is due to the presence of the sequence "*accidente de tráfico* (car accident)", which our model interprets as a violent incident. However, in reality, the tweet is talking about some lanes being closed due to some car accident, but not really talking about the accident itself.

Following examples correspond to false negatives cases:

#2Septiembre | Policia Nacional de #Nicaragua presenta a 16 sujetos que fueron capturados en el Dpto de Chinandega por cometer delitos como;robo con intimidación,robo con fuerza, tráfico de drogas y delitos contra la Mujer #JuntoALaComunidad #2021SoberaniaEnMiTierra @vppolicial https://t.co/i9OikTnN5b
Acusan a mayor de la Policía por muerte de joven de 19 años en el paro. Audiencia preparatoria por homicidio de Santiago Murillo seguirá en noviembre. Le contamos los detalles https://t.co/12uXGvoNahttps://t.co/DyLLxTO3EZ

Table 8
False negative examples: Tweets classified as non containing violent events

In this case, both tweets are talking about some events that occurred in the past. Although further analysis is required, we argue that this type of case (i.e., past events descriptions) might not be as frequent as "in the moment events" (real-time) events. This could be one explanation for our model to confuse these types of cases.

6. Conclusions

This paper describes our participation at the DA-VINCIS@IberLEF 2022 challenge on the *Violent-Event-Identification* and *Violent-Event-Categorization* subtasks. Our participation aimed at analyzing the performance of recent Multi-Task Learning and NLP technologies for solving the posed tasks. Our performed experiments showed that the proposed solution can leverage the implicit similarity, relationship, and present hierarchy in the data at the moment of learning the classification task.

As future work, we plan to evaluate the impact of hyperparameter tuning, as well as alternative ways to assemble the predictions.

Acknowledgments

Esaú Villatoro-Tello, was supported partially by Idiap Research Institute, SNI-CONACyT, and UAM-Cuajimalpa Mexico during the elaboration of this work. The authors thank CONACYT, INAOE and CIMAT for the computer resources provided through the INAOE Supercomputing Laboratory's Deep Learning Platform for Language Technologies (*Laboratorio de Supercómputo: Plataforma de Aprendizaje Profundo*) with the project *"Identification of Aggressive and Offensive text through specialized BERT's ensembles"* and CIMAT Bajio Supercomputing Laboratory (#300832). Vallejo-Aldana and Lopez-Monroy would like to thank CONACYT for its support through projects *"Ciencia de datos aplicado al análisis de expedientes de personas desaparecidas"*.

References

[1] M. Osborne, S. Moran, R. McCreadie, A. Von Lunen, M. Sykora, E. Cano, N. Ireson, C. Macdonald, I. Ounis, Y. He, et al., Real-time detection, tracking, and monitoring of automatically discovered events in social media, in: Proceedings of 52nd annual meeting of the association for computational linguistics: system demonstrations, 2014, pp. 37–42.

[2] U. Dikwatta, T. Fernando, Violence detection in social media-review, Vidyodaya Journal of Science 22 (2019).

[3] F. A. Pujol, H. Mora, M. L. Pertegal, A soft computing approach to violence detection in social media for smart cities, Soft Computing 24 (2020) 11007–11017.

[4] L. J. Arellano, H. J. Escalante, L. Villaseñor-Pineda, M. Montes-y Gómez, F. Sanchez-Vega, Overview of da-vincis at iberlef 2022: Detection of aggressive and violent incidents from social media in spanish, in: Procesamiento del Lenguaje Natural, volume 69, 2022.

[5] J. M. Pérez, D. A. Furman, L. A. Alemany, F. Luque, Robertuito: a pre-trained language model for social media text in spanish, arXiv preprint arXiv:2111.09453 (2021).

[6] M. Guzman-Silverio, Á. Balderas-Paredes, A. P. López-Monroy, Transformers and data augmentation for aggressiveness detection in mexican spanish., in: IberLEF@ SEPLN, 2020, pp. 293–302.

[7] G. Kobayashi, T. Kuribayashi, S. Yokoi, K. Inui, Incorporating residual and normalization layers into analysis of masked language models, arXiv preprint arXiv:2109.07152 (2021).

[8] R. Caruana, Multitask learning, Machine learning 28 (1997) 41–75.

[9] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, Bert: Pre-training of deep bidirectional transformers for language understanding, arXiv preprint arXiv:1810.04805 (2018).

[10] C. Lee, S. Jung, K. Kim, G. G. Lee, Hybrid approach to robust dialog management using agenda and dialog examples, Computer Speech & Language 24 (2010) 609–631.

[11] R. Collobert, J. Weston, A unified architecture for natural language processing: Deep neural networks with multitask learning, in: Proceedings of the 25th international conference on Machine learning, 2008, pp. 160–167.

[12] J. Baxter, A bayesian/information theoretic model of learning to learn via multiple task sampling, Machine learning 28 (1997) 7–39.

[13] I. Loshchilov, F. Hutter, Decoupled weight decay regularization, arXiv preprint arXiv:1711.05101 (2017).

[14] J. Cañete, G. Chaperon, R. Fuentes, J.-H. Ho, H. Kang, J. Pérez, Spanish pre-trained bert model and evaluation data, in: PML4DC at ICLR 2020, 2020.