Priberam Labs at the NTCIR-15 SHINRA2020-ML: Classification Task

Rúben Cardoso
Priberam Labs, Portugal
ruben.cardoso@priberam.pt

Afonso Mendes
Priberam Labs, Portugal
amm@priberam.pt

Andre Lamurias
Priberam Labs, Portugal
andre.lamurias@priberam.pt

ABSTRACT

Wikipedia is an online encyclopedia available in 285 languages. It composes an extremely relevant Knowledge Base (KB), which could be leveraged by automatic systems for several purposes. However, the structure and organisation of such information are not prone to automatic parsing and understanding and it is, therefore, necessary to structure this knowledge. The goal of the current SHINRA2020-ML task is to leverage Wikipedia pages in order to categorise their corresponding entities across 268 hierarchical categories, belonging to the Extended Named Entity (ENE) ontology.

In this work, we propose three distinct models based on the contextualised embeddings yielded by Multilingual BERT. We explore the performances of a linear layer with and without explicit usage of the ontology's hierarchy, and a Gated Recurrent Units (GRU) layer. We also test several pooling strategies to leverage BERT's embeddings and selection criteria based on the labels' scores. We were able to achieve good performance across a large variety of languages, including those not seen during the fine-tuning process (zero-shot languages).

KEYWORDS

Text Classification, Multilingual, BERT, Wikipedia

TEAM NAME

Priberam Labs - PribL

SUBTASKS

Shinra2020-ML: Classification Task (English, Spanish, French, German, Chinese, Russian, Portuguese, Italian, Arabic, Turkish, Dutch, Polish, Korea, Norwegian, Czech)

1 INTRODUCTION

Wikipedia is a free online encyclopedia. It is an open-access collection of pages in 285 languages which are created, edited and maintained by a large community of volunteers, under a system known as open collaboration. The diverse and far-reaching coverage of topics and languages makes Wikipedia a very complete Knowledge Base (KB).

However, Wikipedia was designed and built as a resource for people and it is not trivial to manipulate its information using Artificial Intelligence systems. Therefore, the SHINRA project aims to structure the information contained in Wikipedia in different languages, with the purpose of better leveraging such information with automatic systems.

The SHINRA2020-ML task [17] aims to categorise Wikipedia entities in 30 languages, based on the Extended Named Entity(ENE) definitions including 200+ categories [15, 17]. These definitions compose a taxonomy with 4 increasingly specific hierarchical levels which enable a fine-grained typing of entities. For example, "New York" should be classified as "1.5.1.1", where the 4 layers of hierarchy are Name (1) - Location (1.5) - Geological and Political Entity (1.5.1) - City (1.5.1.1). Note that since the present ontology allows for multi-label classification, an entity can be assigned more than one type.

The training data consists of hand-categorised entities from the Japanese Wikipedia annotated by experts, and language-links associating these Japanese pages to their corresponding Wikipedia page in each one of the 30 target languages. If the corresponding page does not exist in one of these languages, the page is not considered for that language. A simple step of preprocessing done by SHINRA's organisation leveraged such data to yield, for each language, all the hand-categorised page IDs and their corresponding gold labels.

In this task, we tackle the problem of multilingual multi-label classification. The first challenge is related to the multilingual component of the task: the desired system should be able to achieve good performance in a large set of languages, with variable amount of training data available. This way, the models should be trained with multilingual data and, preferably, should be able to maintain good performance on zero-shot languages. The second challenge arises because a single Wikipedia page can be classified with several categories. Even though ≈ 98 % of the considered pages are assigned with only one category, the classification of the remaining pages poses an additional challenge.

This paper describes our participation at the NTCIR-15 task SHINRA2020-ML. We explore the performance of several systems based on the contextualised embeddings generated by multilingual BERT (mBERT) combined with different pooling strategies and classifiers. The first classifier was a simple linear layer projecting a pooled representation of mBERT's embeddings onto the decision space. This same model was also trained with a small variation in the training data to better leverage the label's hierarchical structure. The second classifier was a Gated Recurrent Units (GRU) layer that sequentially predicted the 4 hierarchical levels of the ontology. We propose the following contributions:

- A study of the performance yielded by different pooling strategies for the embeddings generated by mBERT, using a linear layer as classifier.
- A model trained with an extended version of the gold labels which includes the hierarchical parents of leaf categories, with the purpose of better leveraging the ontology's structure.
- A model combining mBERT's embeddings with a GRU layer that sequentially predicted the 4 hierarchical levels of the ontology. Tests were conducted with two different pooling strategies.

- These models have proved capable of achieving good performance on multilingual classification across very different languages, including zero-shot languages.

2 RELATED WORK

The simplest approach to classify texts in multiple languages follows a naive approach which considers the problem as multiple independent problems of monolingual text classification [8]. This means that a model is trained for each language with a corpus composed only by texts on that specific language. To classify a given target document, the suitable classifier is selected and then used to predict the appropriate categories. However, this naive strategy fails to take advantage of the corpus' multiliguality and prevents knowledge transfer between different languages, i.e., the model is inherently incapable of leveraging relations learnt from one language and applying them to another.

A possible solution is the use of multilingual word embeddings, capable of mapping words in different languages onto the same vector space and resulting in a language-agnostic model. Extensive literature has been published on this topic [1, 3, 16]. An example of such strategy combines multilingual word embeddings with character n-gram features and uses a Support Vector Machine (SVM) as classifier [14]. On this work, monolingual word embeddings are trained for each one of the considered languages and, then, linear mappings between the different monolingual embeddings map them to the same vector space.

More complex classification models can involve convolutional neural networks (CNN), which are able to extract the text's most relevant features and leverage them to predict its categories [10, 20]. These convolutional layers can also be combined with additional resources, such as auto-associative memory relationships or bidirectional long short-term memory units (LSTMs), and applied to multilingual problems [11, 19].

Regarding the use of recurrent neural networks (RNN), MultiFit should be highlighted [6]. This model's architecture consists of 4 quasi-recurrent neural network (QRNN) layers followed by an aggregation layer and 2 linear layers. First, a Language-Agnostic Sentence Representation (LASER) model [2] is trained on the text classification task with multilingual data, while a second QRNN based model is solely pretrained on the target language. Then, the label predictions from the LASER multilingual model are used to fine-tune the pretrained monolingual model on the text classification task. This final model shows very good zero-shot performance, even for very low resource languages. Nevertheless, this approach requires an individual model pretrained on each language, which is computationally expensive and impractical.

Recently, the use of transformer-based language models pretrained on very large corpora has advanced the state of the art in several Natural Language Processing tasks. The most widely used of these models is BERT, Bidirectional Encoder Representations from Transformers [5], capable of yielding contextualised embeddings for the input tokens and an embedding for the whole input sequence. A pooling strategy can then be applied to combine these embeddings and, typically, a final linear layer acts as classifier for the specific downstream task. This strategy results in great flexibility and performance. For text classification, usually, only the embedding representing the whole input sequence is used. Furthermore, the capabilities of such approach have been extended to multilingual tasks by Multilingual BERT (mBERT). This model presents the same architecture as the original BERT but it is trained on a large multilingual corpus containing 104 languages. The resulting model is capable of achieving impressive performance on multilingual tasks, including under a zero-shot paradigm [13]. Performance comparisons between the recent BERT models and the more classical approaches using RNNs, such as LSTMs, show that, for small training corpora and low resource languages, models based on RNNs tend to outperform BERT based models on text classification tasks [6, 7]. Despite this performance gain, these models are monolingual and, therefore, require specific training for the target language. In the present work, we propose several models based on Multilingual BERT embeddings. This strategy enables the fine-tuning of a single model with multilingual data, capable of classifying texts not only on languages contained within the training set but also for zero-shot languages.

3 MODELS

We propose 3 distinct types of models based on Multilingual BERT's embeddings. For all these models, the first 511 tokens of each Wikipedia article are concatenated with a preceding [*CLS*] token, resulting in an input structure for BERT with the form ([*CLS*], $token_0$, ..., $token_{510}$). BERT yields the contextualised embeddings, with 768 dimensions, corresponding to each one of the 511 tokens and the [*CLS*] token embedding representing the whole input sequence. This encoding structure is represented on figure 1. The following proposed models differ in the way they leverage these embeddings and the approach used to handle the ontology's hierarchical structure.

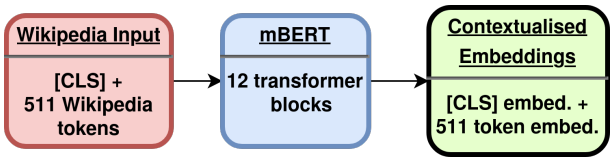

Figure 1: Multilingual BERT is used to obtain the contextualised embeddings corresponding to the first 511 tokens of each Wikipedia page.

3.1 Linear Classification

Our first and simplest approach to the current task consists of using a linear layer as classifier. This layer receives a pooled representation of mBERT's output and projects it onto the decision space. The hierarchical structure of the labels was not explicitly leveraged and, therefore, the decision space is composed only by the 193 leaf labels, i.e., those which correspond to a terminal hierarchical node. To understand the impact of different combinations of token embeddings, three pooling strategies were tested:

- mBERT+CLS: only the [*CLS*] token embedding is used. This embedding passes through a dimension-preserving linear

layer (768x768 dimensions) with hyperbolic tangent activation and a dropout layer with $p = 0.1$. The resulting representation, denominated *pooled CLS embedding*, is used as input to the final linear layer (classifier). This approach is represented on figure 2. This model is hereafter denominated LINEAR+CLS.

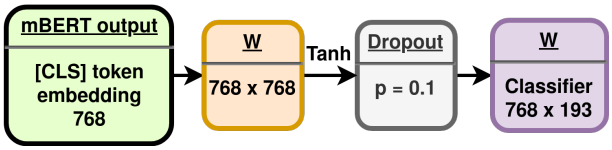

Figure 2: mBERT+CLS pooling strategy and final classifier.

- mBERT+MEAN: leverages all the 511 contextualised token embeddings and [*CLS*] embedding yielded by mBERT through a simple average operation, as shown in figure 3. This model is hereafter denominated LINEAR+MEAN.

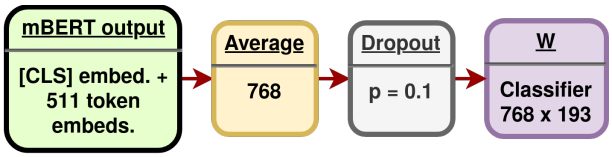

Figure 3: mBERT+MEAN pooling strategy and final classifier.

- mBERT+CONCAT: this pooling strategy combines concatenation and averaging of different token embeddings. First, the *pooled CLS embedding* is concatenated with the token embeddings corresponding to the first 200 tokens of each Wikipedia article. This results in a hidden size of $768 + 200 \times 768 = 154368$. Then, the average of the remaining 311 token embeddings is computed and its result is concatenated to the previous representation, resulting in a final pooled representation with 155136 dimensions, as represented in figure 4. This model is hereafter denominated LINEAR+CONCAT.

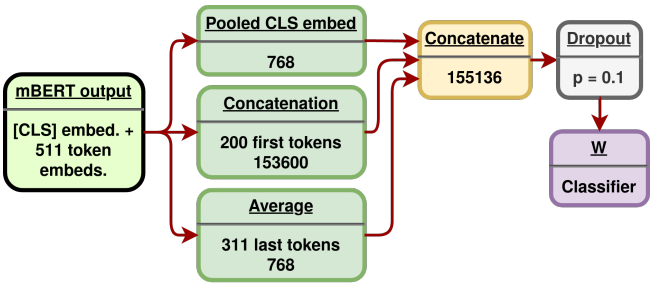

Figure 4: mBERT+CONCAT pooling strategy and final classifier.

Finally, the linear layer yields a score for each label. To decide whether each one of these labels should be considered, three approaches were tested:

- Threshold: a global threshold is fine-tuned on the development set, labels whose scores are above this threshold are considered.
- Max Score: given that only about 2 to 3 % of the samples have multiple gold labels, the current task can be approximated as a single-label classification problem without a necessary decrease of performance. This strategy selects for each Wikipedia page the leaf label with maximum score.
- Threshold with Max Score: the Threshold approach is applied as before, however, since for some Wikipedia articles all their corresponding label scores may be below the global threshold, this approach leaves some samples without any labels. For these cases, the current strategy performs an additional step which assigns them their maximum scored label.

3.2 Multi-level Hierarchical Classification

This model presents the same architecture and pooling strategy as the previous LINEAR+CONCAT model, represented in figure 4. However, the gold labels used during the training process differ from the ones used in the previous classifiers.

To leverage the hierarchical structure of the Extended Named Entity ontology, the gold labels were decomposed into their hierarchical ancestors and the resulting set of labels became the new gold label set. The problem remains a multi-label classification problem, however, the number of labels per Wikipedia article increases considerably. For example, if one of the labels assigned to a sample is "1.10.4.1" (Fungus), the new set of labels used for training is ["1" (Name), "1.10" (Natural_Object), "1.10.4" (Living_Thing), "1.10.4.1" (Fungus)]. This strategy allows the model to learn not only the leaf labels but also the hierarchical steps that lead up to such labels. Note that, with this approach, the decision space includes all the 268 topology labels.

As for the previous models, during test time a score is yielded by the linear layer and a global score threshold is fine-tuned to decide whether or not a label should be included. However, for this model, an additional step is required: if one of the predicted labels is not a leaf label, its hierarchical descendent with the highest score is selected as part of the predicted label set. This process is repeated until all the labels in the predicted set correspond to leaf labels. The present model is hereafter denominated MULTI-LEVEL HIERARCHICAL.

3.3 Hierarchical Sequential Classification

The present classification approach explicitly leverages the ontology's hierarchical structure. A Gated Recurrent Units (GRU) layer with a hidden size of 768 dimensions is used to sequentially predict the labels corresponding to the 4 hierarchical levels. This approach also approximates the task as a single-label classification problem.

In more detail, a loop executes 4 steps of the GRU layer. Starting from the more general first hierarchical label, at each consecutive step, an additional more fine-grained label is predicted. A masking system enforces the hierarchical structure by reducing the set of possible labels at each step to those corresponding to direct descendants of the label selected in the previous hierarchical level.

Regarding this model's architecture, it presents a general structure similar to the previous models with the difference that before the final linear layer a Gated Recurrent Units (GRU) layer is added. A scheme of such structure is shown in figure 5.

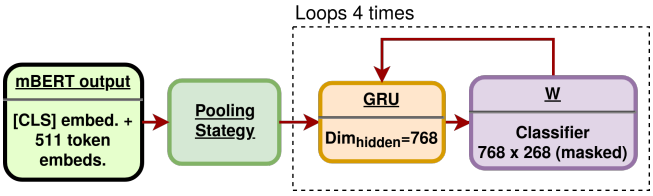

Figure 5: Hierarchical sequential classifier architecture. The GRU's input dimension depends on the pooling strategy used.

The key part of this model is the GRU/Classifier loop: the GRU's input is the concatenation of the pooling output with the embedding of the label predicted in the previous step. This embedding is the line of the classifier's matrix corresponding to such a label. For the first loop step, a previously predicted label is not available and, therefore, a trainable initial embedding is used.

Concerning the pooling strategies, two options were tested: mBERT+CLS and mBERT+CONCAT, as described in section 3.1. These options result in the models hereafter denominated GRU+CLS and GRU+Concat, respectively.

4 EXPERIMENTAL SETUP

In this section, we describe how the data provided by SHINRA's organisation was used and also report several details regarding the model's implementation and training.

4.1 Data

Out of the complete set of 31 languages with available annotations, we selected the following 13 as training data for all our models: English (EN), German (DE), Spanish (ES), French (FR), Italian (IT), Portuguese (PT), Russian (RU), Turkish (TR), Arabic (AR), Chinese (ZH), Polish (PL), Dutch (NL), and Korean (KO). This selection takes into account not only the total number of annotated pages per language but also the intention of leveraging the most spoken languages in the world and having a considerable variability of writing systems. During the training process, the articles corresponding to each language were split into 10 equal slices, each containing 10% of the articles in that language. The model is trained with the first slices of all languages, then the second slices, and this process is repeated until all the slices have been used.

The 13 selected languages contain a total of $21M$ Wikipedia pages, out of which $3.1M$ are annotated. These annotated pages were randomly split into two sets: 95% as training set and the remaining 5% as a development set used to evaluate the model's performance.

An additional leaderboard set with 2000 samples per language was released. This set does not contain the gold annotations, however, it enables the submission of its corresponding predictions on a public leaderboard[1], which yields scores for the micro precision, recall and F1 metrics. The leaderboard set is not the official test set used for the task, it simply allows public comparison of model performances during the development period.

For some models, we evaluated the performance on this test set in a zero-shot paradigm, i.e., on languages which we did not include in our training set. These zero-shot languages are Norwegian (NO), Danish (DA), Czech (CS), Ukrainian (UK), Vietnamese (VI), and Hindi (HI). Given that Shinra2020-ML is a shared-task with the purpose of classifying all the Wikipedia articles for several languages, the last relevant set for this task is the complete Wikipedia dump for the languages to be submitted. These languages consist of the 13 selected training languages and, for one model, also Czech and Norwegian. From this Wikipedia dump set, a subset of annotated articles for each language composes the official test set used to rank the final model's performance.

4.2 Training and Hyper-parameters

All models were implemented using the Python packages Transformers and PyTorch [12, 18]. The contextualised embeddings were obtained from the pretrained model BERT-base multilingual cased.

The training was performed with a maximum sequence length of 512, a batch size of 32, and a maximum learning rate of 2×10^{-5} following a linear warm-up strategy with 10000 warm-up steps. The models were trained on a GPU NVIDIA Quadro RTX 8000, with an epoch taking $\approx$ 1 day.

5 RESULTS

It is important to understand the several metrics presented and how the different datasets used for evaluation affect these metrics values. We show scores corresponding to both micro and macro averages. For the micro average, metrics are calculated globally by counting the total true positives, false negatives and false positives. This average is affected by the dataset's distribution of labels, inherently assigning more weight to those with larger frequency. For the macro average, metrics are calculated for each label, and then their unweighted mean is computed. This results in scores which are independent of the set's label distribution.

Table 1 shows the values obtained for different metrics computed for both development and leaderboard sets. These results correspond to the best performances yielded by the Linear+Concat model for the English (EN), Portuguese (PT) and Korean (KO) languages.

These results show that, independently of the language, the micro scores computed for the leaderboard set are considerably smaller than those corresponding to the development set. Given that this behaviour is verified for all languages and for several samplings of the development set, we can conclude that these two sets have different distributions of labels and, very likely, labels for which the model shows worse performance are much more represented on the leaderboard set. This way, to ensure that our final models perform well for the majority of the labels and, consequently, across datasets with different label distributions, we train the models with the goal of maximising the macro F1 score on the development set.

[1]Shinra2020-ML Leaderboard: https://www.nlp.ecei.tohoku.ac.jp/projects/AIP-LB/task/shinra2020-ml

Set	Metric	EN	PT	KO
Dev	MaF1	0.5701	0.5355	0.5415
	MaR	0.5641	0.5373	0.5486
	MaP	0.5836	0.5382	0.5444
	uF1	0.9586	0.9601	0.9494
	uR	0.9546	0.9474	0.9440
	uP	0.9626	0.9732	0.9549
LeadB	uF1	0.733	0.696	0.720
	uR	0.745	0.726	0.738
	uP	0.726	0.681	0.711

Table 1: LINEAR+CONCAT model scores corresponding to the number of training steps and score threshold with best performance for the development and leaderboard datasets.

The multilinguality of the current task, arising from the need to classify Wikipedia pages in different languages, implies that the models' performance must be evaluated individually for each language. Different languages can have different best-performing models corresponding to different score thresholds and number of training steps. To properly understand the evolution of the models' performance scores for each language throughout the training process, plots such as those shown in figures 6 and 7 were generated for all the 13 selected training languages. Each model was, in general, trained for 2 to 4 epochs and the plots show that quite possibly these could be further trained with additional performance gains. It is also clear that generally the macro F1 scores increase at a similar rate during the training process, maintaining their relative performances. Due to computational limitations, the models were stopped early or were trained for less epochs to allow the training of more promising models.

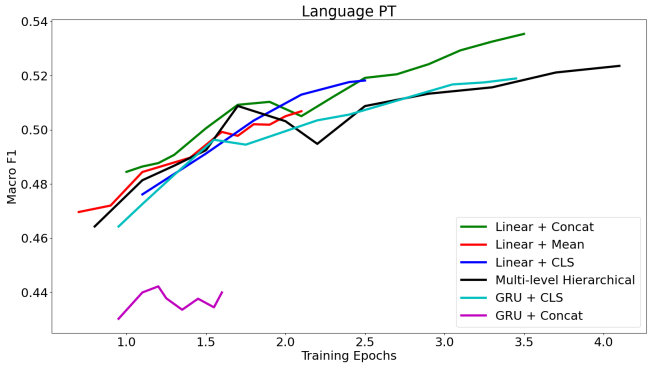

Figure 6: Macro F1 score evolution throughout training for Portuguese.

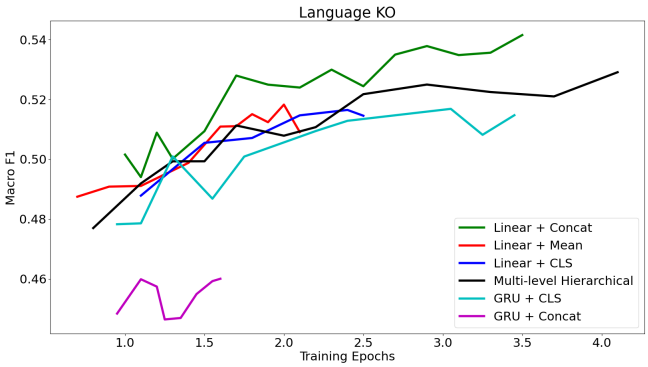

Figure 7: Macro F1 score evolution throughout training for Korean.

Table 2 shows the macro F1 scores evaluated on the development set. The values shown correspond to the best-performing number of training steps for each model, and for the Linear and Multi-level Hierarchical models also the best-performing score threshold. Such parameters can be found in table 7 under appendix A. Table 2 shows that the best pooling strategy when using a linear layer as classifier is CONCAT. These results were expected given that this model not only leverages the *pooled CLS embedding* but also the individual embeddings of the first, and therefore most relevant, tokens.

The model LINEAR+CONCAT is also the best-performing model for most of the languages, only for Chinese and Italian is the MULTI-LEVEL HIERARCHICAL model capable of very slightly outperforming it.

Regarding the HIERARCHICAL SEQUENTIAL models, which sequentially predict the 4 hierarchical label levels using a GRU, it is interesting to note that the CLS pooling can achieve performances very similar to the remaining models while the CONCAT pooling performs considerably worse. Given that the only difference is the size of the GRU's input (from 1536 dimensions for CLS to 155904 for CONCAT), it is possible that the GRU was not able to leverage such a large input.

	LINEAR+ CONCAT	LINEAR+ MEAN	LINEAR+ CLS	MULTI-LEVEL HIERARCHICAL	GRU+ CLS	GRU+ CONCAT
EN	**0.5701**	0.5494	0.5528	0.5669	0.5572	0.4790
ES	**0.5510**	0.5198	0.5271	0.5408	0.5282	0.4557
FR	**0.5523**	0.5174	0.5176	0.5433	0.5190	0.4597
DE	**0.5397**	0.5145	0.5185	0.5378	0.5339	0.4616
ZH	0.5499	0.5225	0.5317	**0.5505**	0.5360	0.4693
RU	**0.5246**	0.5061	0.5030	0.5200	0.5029	0.4433
PT	**0.5355**	0.5069	0.5182	0.5236	0.5190	0.4422
IT	0.5397	0.5164	0.5238	**0.5398**	0.5262	0.4609
AR	**0.4392**	0.4320	0.4346	0.4347	0.4153	0.3649
TR	**0.4892**	0.4756	0.4779	0.4848	0.4734	0.4067
NL	**0.5554**	0.5226	0.5307	0.5428	0.5347	0.4602
PL	**0.5248**	0.5048	0.5048	0.5159	0.5106	0.4412
KO	**0.5415**	0.5183	0.5165	0.5291	0.5168	0.4601

Table 2: Best macro F1 scores for the development set.

Table 3 shows the micro F1 scores evaluated on the leaderboard set. The models, number of training steps, and score thresholds are the same as those used for table 2. On this leaderboard set, we have only experimented with the models that achieved best performances on the development set: Linear+Concat, Multi-level Hierarchical and GRU+CLS. Concerning the Linear+Concat model, we explored the performance of the 3 possible approaches to decide whether or not a label should be considered. Despite their similar performance, the Threshold with Max Score strategy tends to outperform or at least match the other strategies.

On this leaderboard aet, the performances of the Linear+Concat and the GRU+CLS models are more similar, which causes the best-performing model to vary with the considered language. We have additionally evaluated the zero-shot performance of the models that do not require tuning of score thresholds: Linear+Concat Max Score and GRU+CLS. These performances were evaluated for Czech (CS), Ukrainian (UK), Hindi (HI), Vietnamese (VI), Danish (DA), and Norwegian (NO). For both models, the performance slightly decreased for zero-shot languages: average decrease of 3.7% for Linear+Concat Max Score and 4.8% for GRU+CLS. These zero-shot performances are nonetheless impressive given that these languages present a large variety of writing systems and many of them are considerably different from the languages used during fine-tuning. In general, the Linear+Concat model showed better zero-shot performance than the GRU+CLS.

	Linear+ Concat Threshold	Linear+ Concat Max score	Linear+ Concat Threshold w/ Max Score	Multi-level Hierarchical	GRU+ CLS
EN	0.733	**0.739**	**0.739**	0.713	0.707
ES	0.740	0.739	0.744	0.739	**0.751**
FR	0.700	0.722	0.726	0.696	**0.735**
DE	**0.758**	0.754	**0.758**	0.743	0.720
ZH	0.718	0.735	0.732	0.598	**0.754**
RU	0.737	**0.745**	0.744	0.723	0.730
PT	0.696	0.699	0.699	0.703	**0.710**
IT	0.706	0.711	0.706	0.702	**0.734**
AR	0.683	0.678	0.683	0.678	**0.702**
TR	0.728	0.723	**0.732**	0.711	0.699
NL	0.702	0.724	0.719	**0.738**	0.729
PL	**0.766**	0.757	**0.766**	0.701	0.722
KO	0.720	**0.746**	0.731	0.721	0.738
CS	-	**0.692**	-	-	**0.692**
UK	-	**0.696**	-	-	0.668
HI	-	**0.605**	-	-	0.585
VI	-	**0.722**	-	-	0.699
DA	-	0.717	-	-	**0.722**
NO	-	**0.717**	-	-	0.700

Table 3: Micro F1 scores for the leaderboard set.

From the results shown in table 3, we selected the two best models to official submit to the Shinra2020-ML task: Linear+Concat Threshold with Max Score and GRU+CLS. We submitted results for our 13 selected training languages and, for the Linear+Concat model, we have also submitted results for Czech and Norwegian. The micro F1 scores evaluated on the official test set are shown in table 4.

Once again, these two models achieve similar performances for all the languages, with the best model depending on the considered language. Finally, the zero-shot performance of the GRU+CLS model achieves again scores similar to those of the remaining languages.

	Linear+Concat	GRU+CLS
EN	0.8012	**0.8127 (5th)**
ES	**0.8072 (5th)**	0.8030
FR	**0.7852 (3rd)**	0.7793
DE	0.7983	**0.8024 (5th)**
ZH	**0.7937 (3rd)**	0.7838
RU	**0.8308 (2nd)**	0.8260
PT	0.8188	**0.8236 (2nd)**
IT	0.8189	**0.8192 (4th)**
AR	0.7545	**0.7627 (1st)**
TR	0.8323	**0.8436 (5th)**
NL	**0.8126 (5th)**	0.8095
PL	**0.8346 (5th)**	0.8273
KO	0.8104	**0.8151 (5th)**
CS	-	0.8119 (5th)
NO	-	0.7839 (5th)

Table 4: Micro F1 scores evaluated on the official test set and corresponding system ranking within the Shinra2020-ML task. The corresponding number of training steps and score threshold can be found in table 7 under appendix A.

6 ERROR ANALYSIS

To better understand the results obtained and compare the capabilities of the two submitted models, we analysed the mistakes made by each models for Portuguese (PT) and Korean (KO). The results are shown in table 5. We considered the following types of mistakes:

- Completely incorrect: zero matches between the predicted and gold label sets.
- Over-predicted: predicted set contains at least one correct label, however, additional incorrect labels are also present.
- Under-predicted: predicted set contains at least one correct label, however, at least one gold label is missing.

A sample is only considered as correctly predicted if there is a perfect match between predicted and gold label sets.

From table 5, we can notice that the Linear+Concat model shows a considerable number of over-predictions for both languages. On the other hand, as expected, the GRU+CLS model cannot over-predict labels since it only predicts 1 label per article. However, despite this, the GRU+CLS model shows more incorrect classifications than the Linear+Concat model because the single label chosen by this first model is more often the incorrect one. The number of under-predicted labels is similar across both models and languages.

Model	Linear+Concat		GRU+CLS	
	PT	KO	PT	KO
#correct	10133	8922	10025	8794
#incorrect	546	610	654	738
#completely incorrect	384	451	644	727
#over-predicted	157	155	0	0
#under-predicted	24	25	21	18
#over and under-predicted	19	21	11	7

Table 5: Error analysis for the two submitted models for Portuguese (PT) and Korean (KO) languages.

Table 6 shows the labels with smaller $F1$ scores on the development set for the models and languages under analysis. We can see that in general the models struggle to predict the correct fine-grained types of facilities, products and colours. For the Korean language, we additionally notice difficulties related to expressions of time.

Model	PT	KO
Linear+concat	1.6.1: Facility_Part	1.6.6.0: Line_Other
	1.7.4: Money_Form	1.7.23.0: Title_Other
	1.7.10: Offense	1.9.3.0: Natural_Phenomen.
	1.12.0: Color_Other	1.10.5.0: Living_Thing_Part
	1.12.1: Nature_Color	3.8: School_Age
GRU+CLS	1.6.3.1: Tomb	1.6.6.0: Line_Other
	1.6.6.4: Water_Route	1.7.21.5: Style
	1.7.4: Money_Form	1.11.0: Disease_Other
	1.7.14: ID_Number	2.1.1: Time
	1.12.1: Nature_Color	3.8: School_Age

Table 6: Leaf labels with worse $F1$ performance on the development set.

7 CONCLUSIONS

We can conclude that models based on Multilingual BERT can achieve very good performance across languages with different writing systems and diverse linguistic properties, even under a zero-shot paradigm.

The several tests conducted show that the best pooling strategy for a linear layer classifier involves the concatenation of embeddings corresponding to the first tokens in the text, while for a model with GRU the simple *pooled CLS embedding* results in the best performance.

These two models, linear+concat and gru+cls, yield the best results. Their performances are typically very similar and the best model depends on the considered language. The gru+cls model was additionally capable of maintaining its performance even on zero-shot languages.

8 FUTURE WORK

Many different tests and experiments have been left for future work due to lack of time. Such future work includes further training all the models until complete score stabilisation, specially those which were stopped very early. Other possible improvements could be the development of a Hierarchical Sequential model capable of multi-label classification, a new input structure and pooling strategy to leverage more than the first 511 tokens, and experiments with others multilingual language models, such as XLM and XLM-R [4, 9].

9 ACKNOWLEDGEMENTS

This work is supported by the Lisbon Regional Operational Programme (Lisboa 2020), under the Portugal 2020 Partnership Agreement, through the European Regional Development Fund (ERDF), within project TRAINER (Nº 045347).

REFERENCES

[1] Rami Al-Rfou, Bryan Perozzi, and Steven Skiena. 2013. Polyglot: Distributed Word Representations for Multilingual NLP. In *Proceedings of the Seventeenth Conference on Computational Natural Language Learning*. 183–192.

[2] Mikel Artetxe and Holger Schwenk. 2019. Massively multilingual sentence embeddings for zero-shot cross-lingual transfer and beyond. *Transactions of the Association for Computational Linguistics* 7 (2019), 597–610.

[3] Xilun Chen and Claire Cardie. 2018. Unsupervised Multilingual Word Embeddings. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. 261–270.

[4] Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Unsupervised cross-lingual representation learning at scale. *arXiv preprint arXiv:1911.02116* (2019).

[5] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. 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)*. 4171–4186.

[6] Julian Eisenschlos, Sebastian Ruder, Piotr Czapla, Marcin Kadras, Sylvain Gugger, and Jeremy Howard. 2019. MultiFiT: Efficient Multi-lingual Language Model Fine-tuning. 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)*. 5706–5711.

[7] Aysu Ezen-Can. 2020. A Comparison of LSTM and BERT for Small Corpus. *arXiv preprint arXiv:2009.05451* (2020).

[8] Teresa Gonalves and Paulo Quaresma. 2010. Multilingual text classification through combination of monolingual classifiers. Citeseer.

[9] G Lample and A Conneau. [n.d.]. Cross-lingual language model pretraining. arXiv 2019. *arXiv preprint arXiv:1901.07291* ([n. d.]).

[10] Jingzhou Liu, Wei-Cheng Chang, Yuexin Wu, and Yiming Yang. 2017. Deep learning for extreme multi-label text classification. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 115–124.

[11] Jiao Liu, Rongyi Cui, and Yahui Zhao. 2018. *Multilingual Short Text Classification via Convolutional Neural Network: 15th International Conference, WISA 2018, Taiyuan, China, September 14–15, 2018, Proceedings*. 27–38. https://doi.org/10.1007/978-3-030-02934-0_3

[12] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In *Advances in Neural Information Processing Systems 32*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (Eds.). Curran Associates, Inc., 8024–8035. http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf

[13] Telmo Pires, Eva Schlinger, and Dan Garrette. 2019. How Multilingual is Multilingual BERT?. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. 4996–5001.

[14] Barbara Plank. 2017. All-in-1 at ijcnlp-2017 task 4: Short text classification with one model for all languages. In *Proceedings of the IJCNLP 2017, Shared Tasks*. 143–148.

[15] ENE Project. 2020. Extended Named Entity. http://ene-project.info

[16] Sebastian Ruder, Ivan Vulić, and Anders Søgaard. 2019. A survey of cross-lingual word embedding models. *Journal of Artificial Intelligence Research* 65 (2019), 569–631.

[17] Satoshi Sekine, Masako Nomoto, Kouta Nakayama, Asuka Sumida, Koji Matsuda, and Maya Ando. 2020. Overview of SHINRA2020-ML Task. In *Proceedings of the*

NTCIR-15 Conference.

[18] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2019. HuggingFace's Transformers: State-of-the-art Natural Language Processing. *ArXiv* abs/1910.03771 (2019).

[19] Meng Xian-yan, Cui Rong-yi, Zhao Ya-hui, and Zhang Zhenguo. 2019. Multilingual short text classification based on LDA and BiLSTM-CNN neural network. In *International Conference on Web Information Systems and Applications*. Springer, 319–323.

[20] Ritu Yadav. 2020. Light-Weighted CNN for Text Classification. *arXiv* (2020), arXiv–2004.

A SUBMITTED MODELS: TRAINING STEPS AND SCORE THRESHOLDS

	Linear+Concat		GRU+CLS
	Training steps	Score Threshold	Training steps
EN	321055	0.07	316503
ES	284363	-0.84	316503
FR	302709	0.26	316503
DE	266017	-1.57	316503
ZH	284363	-0.10	279807
RU	302709	-0.28	316503
PT	321055	-0.28	316503
IT	302709	0.07	316503
AR	266017	-0.84	316503
TR	302709	-0.28	316503
NL	302709	0.81	298155
PL	284363	-0.84	316503
KO	321055	-0.28	279807
CS	302709	-	-
NO	302709	-	-

Table 7: Number of training steps and score threshold for the submitted Linear+Concat and GRU+CLS models.