Automatic Truecasing of Video Subtitles Using BERT: A Multilingual Adaptable Approach

Ricardo Rei[1], Nuno Miguel Guerreiro[1], and Fernando Batista[2(✉)]

[1] Unbabel, Lisbon, Portugal
{ricardo.rei,nuno.guerreiro}@unbabel.com
[2] INESC-ID Lisboa & ISCTE - Instituto Universitário de Lisboa, Lisbon, Portugal
fernando.batista@iscte-iul.pt

Abstract. This paper describes an approach for automatic capitalization of text without case information, such as spoken transcripts of video subtitles, produced by automatic speech recognition systems. Our approach is based on pre-trained contextualized word embeddings, requires only a small portion of data for training when compared with traditional approaches, and is able to achieve state-of-the-art results. The paper reports experiments both on general written data from the European Parliament, and on video subtitles, revealing that the proposed approach is suitable for performing capitalization, not only in each one of the domains, but also in a cross-domain scenario. We have also created a versatile multilingual model, and the conducted experiments show that good results can be achieved both for monolingual and multilingual data. Finally, we applied domain adaptation by finetuning models, initially trained on general written data, on video subtitles, revealing gains over other approaches not only in performance but also in terms of computational cost.

Keywords: Automatic capitalization · Automatic truecasing · BERT · Contextualized embeddings · Domain adaptation

1 Introduction

Automatic Speech Recognition (ASR) systems are now being massively used to produce video subtitles, not only suitable for human readability, but also for automatic indexing, cataloging, and searching. Nonetheless, a standard ASR system usually produces text without punctuation and case information, which makes this representation format hard to read [12], and poses problems to further

This work was supported by national funds through FCT, Fundação para a Ciência e a Tecnologia, under project UIDB/50021/2020 and by PT2020 funds, under the project "Unbabel Scribe: AI-Powered Video Transcription and Subtitle" with the contract number: 038510. The authors have contributed equally to this work.

M.-J. Lesot et al. (Eds.): IPMU 2020, CCIS 1237, pp. 708–721, 2020.
https://doi.org/10.1007/978-3-030-50146-4_52

automatic processing. The capitalization task, also known as truecasing [13,18], consists of rewriting each word of an input text with its proper case information given its context. Many languages distinguish between uppercase and lowercase letters, and proper capitalization can be found in many information sources, such as newspaper articles, books, and most of the web pages. Besides improving the readability of texts, capitalization provides important semantic clues for further text processing tasks. Different practical applications benefit from automatic capitalization as a preprocessing step, and in what concerns speech recognition output, automatic capitalization may provide relevant information for automatic content extraction, and Machine Translation (MT).

Unbabel combines the speed and scale of automatic machine translation with the authenticity that comes from humans, and is now dealing with an increasing demand for producing video subtitles in multiple languages. The video processing pipeline consists of a) processing each video with an ASR system adapted to the source language, b) manual post-edition of the ASR output by human editors, and c) perform the translation for other languages, first by using a customized MT system, and then by using humans to improve the resulting translations. Recovering the correct capitalization of the words coming from the speech transcripts constitutes an important step in our pipeline due to its impact on the post-edition time, performed by human editors, and on the MT task output. Automatic Video subtitles may contain speech recognition errors and other specific phenomena, including disfluencies originated by the spontaneous nature of the speech and other metadata events, that represent interesting practical challenges to the capitalization task.

This paper describes our approach for automatically recovering capitalization from video subtitles, produced by speech recognition systems, using the BERT model [8]. Experiments are performed using both general written data and video subtitles, allowing for assessment of the impact of the specific inner structural style of video subtitles in the capitalization task.

The paper is organized as follows: Sect. 2 presents the literature review. Section 3 describes the corpora and pre-processing steps used for our experiments. Section 4 presents our approach and the corresponding architecture, as well as the evaluation metrics. Section 5 presents the results achieved, both on a generic domain (monolingual and multilingual) and in the specific domain of video subtitles. Finally, Sect. 6 presents the most relevant conclusions and pinpoints a number of future directions.

2 Related Work

Capitalization can be viewed as a lexical ambiguity resolution problem, where each word has different graphical forms [10,30], by considering different capitalization forms as spelling variations. Capitalization can also be viewed as a sequence tagging problem, where each lowercase word is associated with a tag that describes its capitalization form [6,14,15,18].

A common approach for capitalization relies on n-gram language models estimated from a corpus with case information [10,15,18]. Common classification approaches include Conditional Random Fields (CRFs) [27] and Maximum Entropy Markov Models (MEMM) [6]. A study comparing generative and discriminative approaches can be found in [2]. The impact of using increasing amounts of training data as well as a small amount of adaptation is studied in [6]. Experiments on huge corpora sets, from 58 million to 55 billion tokens, using different n-gram orders are performed in [10], concluding that using larger training data sets leads to increasing improvements in performance, but the same tendency is not achieved by using higher n-gram order language models. Other related work, in the context of MT systems, exploit case information both from source and target sentences of the MT system [1,23,27].

Recent work on capitalization has been reported by [21,25]. [25] proposes a method for recovering capitalization for long-speech ASR transcriptions using Transformer models and chunk merging, and [21] extends the previous model to deal with both punctuation and capitalization. Other recent advances are reported by [29] for Named Entity Recognition (NER), a problem that can be tackled with similar approaches.

Pre-trained transformer models such as BERT [8] have outperformed previous state-of-the-art solutions in a wide variety of NLP tasks [7,8,19]. For most of these models, the primary task is to reconstruct masked tokens by uncovering the relation between those tokens and the tokens surrounding. This pre-train objective proved to be highly effective for token-level tasks such as NER. Bearing this in mind, in this paper, we will follow the approach proposed in [3–5] and address the capitalization task as a sequence tagging problem similar to NER and show that, as in that task, BERT can also achieve state-of-the-art results for capitalization.

3 Corpora

Constructing an automatic translation solution focused on video content is a complex project that can be subdivided into several tasks. In this work, we are focusing on enriching the transcription that comes from the ASR system, by training a model prepared to solve the truecasing problem. This process is of paramount importance for satisfactory machine translation task output and would ultimately alleviate the post-edition time performed by human editors.

3.1 Datasets

Experiments performed in the scope of this paper use internal data (hereinafter referred as *domain* dataset) produced by the ASR system and subsequently post-edited by humans in order to correct bad word transcripts, introduce capitalization and punctuation, and properly segment the transcripts to be used for video subtitling.

Table 1. Source sentence and target tags construction for a given sentence. Note that apart from lowercasing all tokens from the target, punctuation was also stripped to create the source sentence.

Target	Automatic Truecasing of Video Subtitles using BERT: A multilingual adaptable approach
Source	Automatic truecasing of video subtitles using bert a multilingual adaptable approach
Target tags	T T L T T L U U L L L

In order to establish a comparison with a structured out-of-domain training corpus, we use the Europarl V8 corpus. This corpus is composed of parallel sentences which allows for coherent studies in terms of complexity across different languages. As one of the main objectives is that of building a single model that can be used for several languages, we also constructed a dataset composed by sentences in four different languages (English, Spanish, French and Portuguese) in such a way that there are no parallel sentences across different languages.

The dataset composed by English-only sentences will be hereinafter referred as *monolingual* dataset whereas the one composed by sentences in different languages will be referred as *multilingual* dataset.

3.2 Pre-processing

Considering that we want to build a model prepared to receive the outputs from the ASR system and automatically solve the truecasing problem, we removed all punctuation but apostrophes and hyphens which are extensively used in the languages considered for this study. This is an important step towards building a reliable model, since the ASR outputs' punctuation is not consistently trustworthy. For fair comparisons with the generic dataset, punctuation was also removed from its data. Moreover, metadata events such as sound representations (e.g: "laughing") are removed from the domain dataset.

The problem of truecasing is approached as a sequence tagging problem [6, 14,15,18]. Thus, the source sentences for both datasets are solely composed by lowercased tokens, whereas the target sequences for both datasets are composed by the ground truth tags. A tag "U" is attributed to an uppercase token, a tag "T" is attributed to a *title* token (only the first letter is uppercase) and a tag "L" to all the remaining tokens. An example of this procedure can be seen in Table 1. We observed that for the monolingual and multilingual datasets, as the first token tag corresponds to "T" in the vast majority of their sentences, the model would capitalize the first token just for its position. As we do not want to rely on positional information to solve the truecasing problem, if a sentence starts with a *title* token, we do not consider that token during training/testing. Statistics on the size of the train and test set for each dataset (domain, monolingual and multilingual), absolute frequency of each tag and the ratio of not-lowercased tags

Table 2. Size of the train and test set for each dataset.

Dataset		Number of sentences	"L" tags	"U" tags	"T" tags	Not–"L" ratio (%)
Domain	Train	127658	904288	38917	94767	14.78
	Test	10046	76200	3420	8179	15.22
Generic	Train	1908970	42936838	861199	4953879	13.54
	Test	99992	2246120	43236	267972	13.86
Multilingual	Train	1917932	46420467	624757	4532864	11.11
	Test	99985	2399968	29564	240390	11.25

for each dataset is displayed in Table 2. The not–"L" ratio is relevant since the datasets are unbalanced as "L" tags are much more frequent.

4 Approach Description and Evaluation Metrics

As pre-trained text encoders have been consistently improving the state of the art on many NLP tasks, and since we are approaching the problem as a sequence tagging problem, we decided to use the BERT [8] model. The BERT base model is a 12-layer encoder-only bidirectional model based on the Transformer [26] architecture with 768 hidden units that was trained for masked word prediction and on next sentence prediction on a large corpus of plain unlabelled text. We refer to [8] for further details of the model.

4.1 Architecture

Given an input sequence $\boldsymbol{x} = [x_0, x_1, \ldots, x_n]$, the BERT encoder will produce an embedding $\boldsymbol{e}_{x_j}^{(\ell)}$ for each token x_j and each layer ℓ.

In [24], it is revealed that the BERT model captures, within the network, linguistic information that is relevant for downstream tasks. Thus, it is beneficial to combine information from several layers instead of solely using the output of the last layer. To do so, we used the approach in [17,22] to encapsulate information in the BERT layers into a single embedding for each token, $\boldsymbol{e}_{x_j}$, whose size is the same as the hidden size of the model. This embedding will be computed as a weighted sum of all layer representations:

$$\boldsymbol{e}_{x_j} = \gamma \sum_{\ell=0}^{12} \boldsymbol{e}_{x_j}^{(\ell)} \cdot \text{softmax}\left(\boldsymbol{\alpha}\right)^{(\ell)} \tag{1}$$

where γ is a trainable scaling factor and $\boldsymbol{\alpha} = \left[\alpha^{(1)}, \alpha^{(2)}, \ldots, \alpha^{(12)}\right]$ are the layer scalar trainable weights which are kept constant for every token. Note that this computation can be interpreted as **layer-wise attention mechanism**. So, intuitively, higher $\alpha^{(\ell)}$ values are assigned to layers that hold more relevant

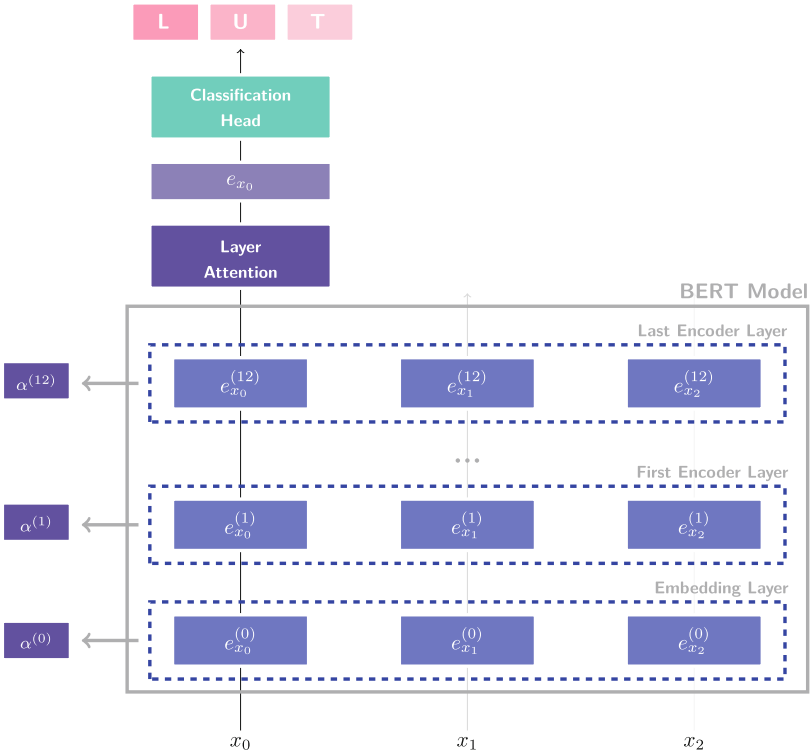

Fig. 1. The architecture of our solution. The output of the BERT model is computed using the Layer Attention block respective to (1). The scalar weights respective to each layer are trained simultaneously with the rest of the model.

information to solve the task. In order to redistribute the importance through all the model layers, we used **layer dropout**, devised in [17], in which each weight $\alpha^{(\ell)}$ is set to $-\infty$ with probability 0.1. This will also prevent overfitting of the model to the information captured in any single layer.

Finally, the embeddings are fed to a classification head composed by a feed-forward neural network which will down-project the size-768 token embedding $\boldsymbol{e}_{x_j}$ to a size-3 logits vector. This vector will then be fed to a softmax layer to produce a probability distribution over the possible tags and the position index of the maximum value of the vector will be considered as the predicted tag (Fig. 1).

4.2 Evaluation Metrics

All the evaluation presented in this paper uses the performance metrics: F1-score and Slot Error Rate (SER) [20]. Only capitalized words (not lowercase) are considered as slots and used by these metrics. Hence, the capitalization SER is computed by dividing the number of capitalization errors (misses and false alarms) by the number of capitalized words in the reference.

Experiments reported here do not consider the first word of each sentence whenever the corresponding case information may be due to its position in the sentence. So, every titled word appearing at the beginning of a sentence will be excluded both at the training and testing stages.

5 Results

In this section, we compare the results of experiments ran on the Europarl V8 corpus and on domain data for both monolingual and multilingual models. After loading the pre-trained model and initializing both the layer-wise attention and the feed-forward projection on top, we split the network parameters into two groups; encoder parameters, composed by the layer-wise attention and the pre-trained transformer architecture, and classification-head parameters, composed by the final linear projection used to compute the logits for each tag. Following the approach in [11,17] we apply discriminative learning rates for the two different groups of parameters. For the classification-head parameters we used a learning rate of 3×10^{-5} with a dropout probability of 0.1. We froze the encoder parameters during the first epoch, and trained them on the subsequent epochs using a 1×10^{-5} learning rate. The optimizer used in both groups was Adam [16]. We use a batch size of 8 for the models trained on the generic and domain datasets, and a batch size of 16 for the models trained on the multilingual dataset. At test time, we select the model with the best validation SER.

In order to evaluate if the models trained on written text data are able to transfer capabilities to in-domain data, we perform domain adaptation by fine-tuning the monolingual models on in-domain data.

We implemented all the models using either the `bert-base-uncased` (for the models trained on monolingual English data) or `bert-base-multilingual-uncased` (for the models trained on multilingual data) text encoders from the Huggingface library [28] as the pre-trained text models and we ran all experiments making use of the Pytorch Lightning wrapper [9].

5.1 Experiments on Europarl Data

For both generic and multilingual datasets, we train models under four settings: *+1.9M* (correspondent to the datasets in Table 2), *200K* (200,000 training sentences), *100K* (100,000 training sentences) and *50K* (50,000 training sentences). We will be referring to the models trained on monolingual data as **monoligual models**, and the models trained on multilingual data as **multilingual models**. Moreover, we trained a Bidirectional Long Short-Term Memory (BiLSTM) with a CRFs model on the entire monolingual dataset (+1.9M setting), which will be referred to as **baseline model** since we used its evaluation results as the baseline.

Monolingual Setting. Results are shown in Table 3. We observe that the monolingual model performs better than the baseline model for all training settings. This is evidence that our approach using pre-trained contextual embed-

Table 3. Results for the monolingual models evaluated on the **generic** test set.

Model architecture	Training setting	SER	F1-score
Baseline (BiLSTM + CRF)	+1.9M	0.1480	0.9200
Monolingual	+1.9M	**0.0716**	**0.9753**
	200K	0.0775	0.9717
	100K	0.0800	0.9701
	50K	0.0850	0.9682

Table 4. Evaluation on the **multilingual** test set.

Model	Training setting	SER	F1-score
Multilingual	+1.9M	**0.1040**	**0.9579**
	200K	0.1206	0.9472
	100K	0.1240	0.9447
	50K	0.1312	0.9379

Table 5. Evaluation on the **monolingual** test set.

Model	Training setting	SER	F1-score
Monolingual	+1.9M	**0.0716**	**0.9753**
Multilingual		0.0761	0.9690
Baseline		0.1480	0.9200

dings is not only able to achieve better results, but it also manages to do so using only a small portion of the data when compared to the baseline model.

Multilingual Setting. Results are shown in Tables 4 and 5. As expected, results for the monolingual model are better than the ones obtained by the multilingual model. Nevertheless, the multilingual model trained on its +1.9M setting outperforms all the models trained on monolingual data under all settings but the +1.9M setting, although this could be happening because the multilingual train dataset has more English individual sentences than the monolingual 200 K setting dataset. The results are evidence that a multilingual model which holds information on several languages is able to achieve similar results to a monolingual model and outperforms previous state-of-the-art solutions trained and tested in an equivalent monolingual setting.

Comparison with the Baseline Model. Results show that both the monolingual and multilingual models outperform the results obtained using the baseline model even when training on a small portion of the available data. Thus, further experiments will be solely evaluated on the models based on our architecture.

5.2 Experiments on Domain Data

All the experiments reported in this section make use of the domain datasets described in Table 2. First, we trained a model using the pre-trained contextual embeddings from `bert-base-uncased` and another using those from `bert-base-multilingual-uncased` on the domain training dataset. We will be referring to these models as **in-domain models**. Then, we perform domain

Table 6. Evaluation on the **domain** test set.

Model	Training setting	SER	F1-score
Domain		**0.2478**	**0.8489**
Monolingual	+1.9M	0.3128	0.7855
	200K	0.3136	0.7827
	100K	0.3071	0.7927
	50K	0.3193	0.7853
Multilingual	+1.9M	0.3285	0.7715
	200K	0.3355	0.7825
	100K	0.3372	0.7794
	50K	0.3530	0.7716

adaptation by loading the obtained models from experiments on the Europarl data and training them with in domain data.

In-domain Models. Results are shown in Table 6. Recalling the dataset statistics from Table 2, the domain dataset is comparable, in terms of number of sentences, with the monolingual dataset for the 50K training setting. Comparing the in-domain model and generic model for this setting, when tested on data from the same distribution that they trained on, we observe that there is a significant offset between the evaluation metrics. This is evidence that there are structural differences between the generic and the domain data. This notion is supported by the evaluation results of the generic and multilingual models initially trained on Europarl data on the domain test set and will be furtherly explored next.

Structural Differences Between Domain and Europarl Data. By observing both datasets, we noticed some clear structural differences between them. For one, since the original samples were segmented (the average number of tokens per sample is 6.74 for the domain training data and 24.19 for the generic training data), there is much less context preservation in the domain data. Moreover, the segmentation for subtitles is, in some cases, made in such a way that multiple sentences fit into a single subtitle, i.e, a single training sample (see Table 7). Since, as we previously remarked, we did not want to use the ASR outputs' punctuation, the truecasing task is hardened as it is difficult for the model to capture when a subtitle ends and a new one start for there can be a non-singular number of ground-truth capitalized tags assigned to words that are not typically capitalized. Note that recovering the initial sentences from the subtitles, i.e, the pre-segmentation transcripts, would be a difficult and cumbersome task. Moreover, different chunks of the ASR outputs' have been post-edited by different annotators which creates some variance in the way the segmentation and capitalization are done for some words (e.g: the word "Portuguese" is written as "portuguese" and attributed the tag "L" two times out of the eleven times it appears in the training data set). Last, when compared with the Europarl data, the in-domain data is significantly more disfluent. This is mainly due to the

spontaneous nature of the source speech, since the ASR outputs are respective to content from video-sharing platforms that is considerably more unstructured (e.g: "Oh my God. Two more. You did it man!").

Table 7. In the example below, extracted from the domain data, we observe that the segmentation caused the capitalized tag respective to "The" to appear in the middle of the subtitle. Since we are not using any punctuation information, this significantly hardens the task of capitalization for this word. It is also noticeable that the length of each subtitle is small, hampering the use of context to solve the task.

ASR output	"After that it's all good, you get on the plane, and you're away. The airport is key to the start of a good beginning to the holiday."
Segmented subtitles	After that it's all good, you get on the plane, and you're away. The airport is key to the start of a good beginning to the holiday
Target tags	T L L L L L L L L L L L L L L L L **T** L L L L L L L L L L L L L

In-domain Adaptation. Given our interest in evaluating the ability to transfer capabilities from the models trained on generic data, we fine-tuned the monolingual models on in-domain data. These models will be referred to as **adapted models**. All four models trained on Europarl data, one for each training setting, are adapted to the domain data. Results shown in Table 8 reveal that all adapted models but the one initially trained in the total setting on Europarl data outperform the domain model. Moreover, the results shown in Fig. 2 indicate that by reducing the original training dataset size, we obtain models that are not only

Table 8. Evaluation results on the **domain** test set.

Model	Training setting	SER	F1-Score
In-domain		**0.2478**	**0.8489**
Monolingual	+1.9M	0.3128	0.7855
	200K	0.3136	0.7827
	100K	**0.3071**	**0.7927**
	50K	0.3193	0.7853
Adapted	+1.9M	0.2482	0.8467
	200K	0.2445	**0.8599**
	100K	0.2469	0.8503
	50K	**0.2404**	0.8540

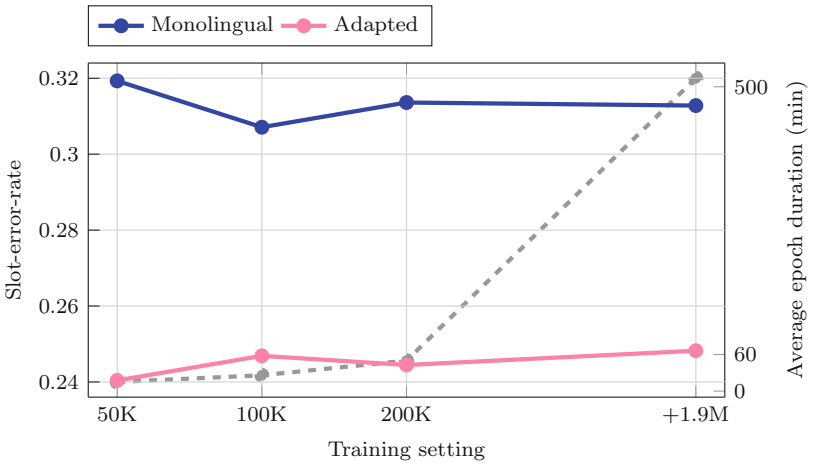

Fig. 2. In dashed, we represent the average duration of an epoch for the initial training of the monolingual model and, in full lines, we represent the SER for the monolingual and adapted models as a function of the original training dataset size. Domain adaptation is the most successful for the model that initially trained faster.

faster to train but also more adaptable to in-domain data, since they are not as prone to overfitting to the training data inner structural style as models that are trained on bigger training datasets.

5.3 Layer-Wise Attention Mechanism

All our models contain a layer-wise dot-product attention mechanism to compute the encoder output as a combination of the output of several encoder layers. This attention mechanism is devised in such a way that layer scalar weights are trained jointly with the encoder layers. By observing Fig. 3, it is clear that some layers contain more significant information than others for solving the truecasing task. Moreover, the effect of fine-tuning the monolingual model on domain data is also felt on the trained weights, in such a way that, generally, its original weight distribution approaches the in-domain model weight distribution.

Center of Gravity. To better interpret the weight distributions in Fig. 3, we computed the **center of gravity** metric as in [24] for each of the models. Intuitively, higher values indicate that the relevant information for the truecasing task is captured in higher layers. Results are shown in Table 9, and, as expected, they are similar across all the trained models. Moreover, comparing with the results obtained for this metric in [24], we observe that the truecasing task center of gravity is very similar to that of the NER task (6.86). This result further supports the previously mentioned notion of similarity between the task at hand and the NER task.

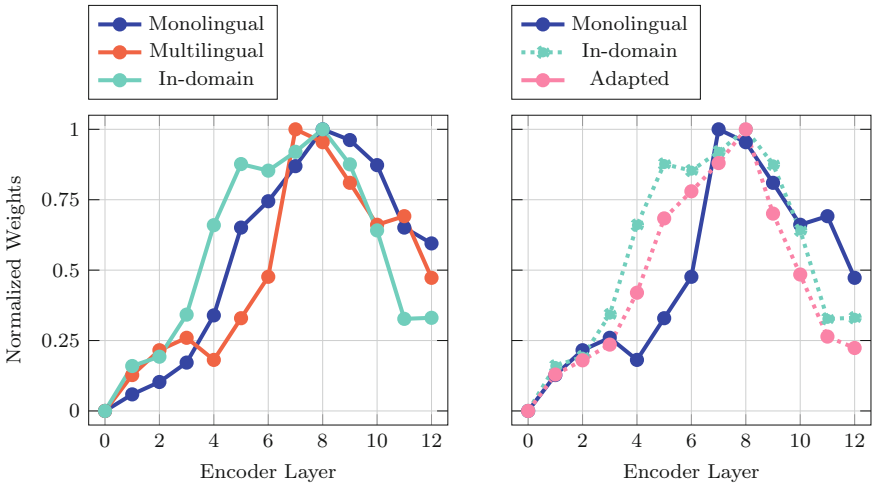

Fig. 3. Normalized weights distribution for the in-domain model, the monolingual and multilingual models trained with the +1.9M setting, and the adapted model initially trained with that same setting. Weight distributions are similar across different models. Moreover, by observing the adapted model weight distribution, we notice that, as expected, the adaptation process brings the weight distribution of the monolingual closer to that of the in-domain model.

Table 9. Center of gravity for the monolingual, adapted model and multilingual trained with the 50K setting.

Model	Training setting	Center of gravity
Monolingual	+1.9M	7.48
Multilingual		7.40
Adapted		6.93
In-domain		7.05

6 Conclusions and Future Work

We made use of pre-trained contextualized word embeddings to train monolingual and multilingual models to solve the truecasing task on transcripts of video subtitles produced by ASR systems. Our architecture, which makes use of a layer attention mechanism to combine information in several encoder layers, yielded consistent and very satisfactory results on the task at hand, outperforming previous state-of-the-art solutions while requiring less data. By performing domain adaptation, we furtherly improved these results, underscoring the notion that models initially trained on less data can adapt better and faster to in-domain data. In the future, we expect improvements on the task by addressing capitalization and punctuation simultaneously in a multitask setting and by making use of additional context by recovering the initial transcripts from the segmented

subtitles. Further gains on this task would constitute a major step towards an improved video processing pipeline for Unbabel.

References

1. Agbago, A., Kuhn, R., Foster, G.: Truecasing for the portage system. In: Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2005), Borovets (2005)
2. Batista, F., Caseiro, D., Mamede, N., Trancoso, I.: Recovering capitalization and punctuation marks for automatic speech recognition: case study for Portuguese broadcast news. Speech Commun. **50**(10), 847–862 (2008)
3. Batista, F., Mamede, N., Trancoso, I.: The impact of language dynamics on the capitalization of broadcast news. In: INTERSPEECH 2008, September 2008
4. Batista, F., Mamede, N., Trancoso, I.: Language dynamics and capitalization using maximum entropy. In: Proceedings of ACL 2008: HLT, Short Papers, pp. 1–4. ACL (2008). http://www.aclweb.org/anthology/P/P08/P08-2001
5. Batista, F., Moniz, H., Trancoso, I., Mamede, N.J.: Bilingual experiments on automatic recovery of capitalization and punctuation of automatic speech transcripts. IEEE Trans. Audio Speech Lang. Process. Spec. Issue New Front. Rich Transcr. **20**(2), 474–485 (2012). https://doi.org/10.1109/TASL.2011.2159594
6. Chelba, C., Acero, A.: Adaptation of maximum entropy capitalizer: little data can help a lot. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP 2004) (2004)
7. Conneau, A., et al.: Unsupervised cross-lingual representation learning at scale. arXiv preprint arXiv:1911.02116 (2019)
8. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: BERT: pre-training of deep bidirectional transformers for language understanding. In: NAACL-HLT (2019)
9. Falcon, W.E.A.: Pytorch lightning. https://github.com/PytorchLightning/pytorch-lightning (2019)
10. Gravano, A., Jansche, M., Bacchiani, M.: Restoring punctuation and capitalization in transcribed speech. In: Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2009), Taipei (2009)
11. Howard, J., Ruder, S.: Universal language model fine-tuning for text classification. In: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 328–339. Association for Computational Linguistics, Melbourne, July 2018. https://doi.org/10.18653/v1/P18-1031. https://www.aclweb.org/anthology/P18-1031
12. Jones, D., et al.: Measuring the readability of automatic speech-to-text transcripts. In: Proceedings of EUROSPEECH, pp. 1585–1588 (2003)
13. Jurafsky, D., Martin, J.H.: Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition, 2nd edn., Prentice Hall PTR (2009)
14. Khare, A.: Joint learning for named entity recognition and capitalization generation. Master's thesis, University of Edinburgh (2006)
15. Kim, J.H., Woodland, P.C.: Automatic capitalisation generation for speech input. Comput. Speech Lang. **18**(1), 67–90 (2004)
16. Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization (2014)
17. Kondratyuk, D., Straka, M.: 75 languages, 1 model: parsing universal dependencies universally (2019). https://www.aclweb.org/anthology/D19-1279

18. Lita, L.V., Ittycheriah, A., Roukos, S., Kambhatla, N.: tRuEcasIng. In: Proceedings of the 41^{st} Annual Meeting on ACL, pp. 152–159. ACL (2003)
19. Liu, Y., et al.: Roberta: a robustly optimized BERT pretraining approach. ArXiv abs/1907.11692 (2019)
20. Makhoul, J., Kubala, F., Schwartz, R., Weischedel, R.: Performance measures for information extraction. In: Broadcast News Workshop (1999)
21. Nguyen, B., et al.: Fast and accurate capitalization and punctuation for automatic speech recognition using transformer and chunk merging (2019)
22. Peters, M.E., et al.: Deep contextualized word representations (2018)
23. Stüker, S., et al.: The ISL TC-STAR spring 2006 ASR evaluation systems. In: Proceedings of the TC-STAR Workshop on Speech-to-Speech Translation, Barcelona, June 2006
24. Tenney, I., Das, D., Pavlick, E.: BERT rediscovers the classical NLP pipeline (2019)
25. Thu, H.N.T., Thai, B.N., Nguyen, V.B.H., Do, Q.T., Mai, L.C., Minh, H.N.T.: Recovering capitalization for automatic speech recognition of Vietnamese using transformer and chunk merging. In: 2019 11th International Conference on Knowledge and Systems Engineering (KSE), pp. 1–5 (2019)
26. Vaswani, A., et al.: Attention is all you need (2017)
27. Wang, W., Knight, K., Marcu, D.: Capitalizing machine translation. In: HLT-NAACL, pp. 1–8. ACL (2006)
28. Wolf, T., et al.: HuggingFace's transformers: state-of-the-art natural language processing. ArXiv abs/1910.03771 (2019)
29. Yadav, V., Bethard, S.: A survey on recent advances in named entity recognition from deep learning models. ArXiv abs/1910.11470 (2018)
30. Yarowsky, D.: Decision lists for lexical ambiguity resolution: application to accent restoration in Spanish and French. In: Proceedings of the 2^{nd} Annual Meeting of the Association for Computational Linguistics (ACL 1994), pp. 88–95 (1994)