Training BERT Models to Carry Over a Coding System Developed on One Corpus to Another

Dalma Galambos[1], **Pál Zsámboki**[2]
[1]Pázmány Péter Catholic University
[2]HUN-REN Alfréd Rényi Institute of Mathematics
[2]Institute of Mathematics, Faculty of Science, Eötvös Loránd University
Budapest, Hungary
galambos.dalma@gmail.com, zsamboki.pal@renyi.hu

Abstract

This paper describes how we train BERT models to carry over a coding system developed on the paragraphs of a Hungarian literary journal to another. The aim of the coding system is to track trends in the perception of literary translation around the political transformation in 1989 in Hungary. To evaluate not only task performance but also the consistence of the annotation, moreover, to get better predictions from an ensemble, we use 10-fold crossvalidation. Extensive hyperparameter tuning is used to obtain the best possible results and fair comparisons. To handle label imbalance, we use loss functions and metrics robust to it. Evaluation of the effect of domain shift is carried out by sampling a test set from the target domain. We establish the sample size by estimating the bootstrapped confidence interval via simulations. This way, we show that our models can carry over one annotation system to the target domain. Comparisons are drawn to provide insights such as learning multilabel correlations and confidence penalty improve resistance to domain shift, and domain adaptation on OCR-ed text on another domain improves performance almost to the same extent as that on the corpus under study. See our code at `https://codeberg.org/zsamboki/bert-annotator-ensemble`.

Keywords: BERT, coding system, domain adaptation, domain shift, ensemble learning, imbalanced dataset, Literary Translation Studies, OCR impact, social perception

1. Introduction

1.1. Objective of the Large Pilot Project Providing the Broader Context of the Present Paper

From the aspect of cultural policy, transition from the Socialist Kádár era (1956–1989) to democracy in Hungary was a crucial period in time. Culture, particularly literature and by extension, literary translation had been heavily funded by the state before the so-called political transformation in 1989, until which literary translators, consequently, had enjoyed a much higher status than in the period since.

This large pilot project chooses a data-driven path to examine this change and blends qualitative and quantitative methods in order to provide a closer look at how literary translators were perceived in the two decades surrounding the regime change. It utilizes a new coding system (we also refer to this as *annotation system* in our paper) tailored to the domain, state-of-the-art classification technology, quantitative and qualitative analysis and network analysis. Background of the project in literary translation studies as well as a more detailed account of the manual coding process and results are discussed in the doctoral dissertation of Galambos (2024).

1.2. Scope of the Present Paper, Main Contributions

The present paper details the classification technology that we use. Since their discovery, transformers (Vaswani et al., 2017) have been dominating the Natural Language Processing (NLP) field. For classification, the BERT architectures (Devlin et al., 2019) are widely and successfully used. We train BERT models on a manually annotated dataset to apply the coding system to another domain, which we call the *target domain*.

Our main contributions discussed in this paper are as follows:

1. We show that with extensive hyperparameter tuning both in pretraining (§3.1) and finetuning, and with loss functions robust to label imbalance in the latter (§3.2), we can teach BERT models complex and highly imbalanced sequence labelling systems. This is verified via 10-fold crossvalidation, the resulting models forming model ensembles for prediction.

2. To evaluate the resistance of our models to domain shift, we select a test set from the target domain for manual validation. We introduce a method to estimate confidence intervals of test results with various sample sizes. We verify that our models can carry over one coding system to the target domain (§4).

LREC-COLING 2024, pages 16698–16712
20-25 May, 2024.

3. In addition to finetuning off-the-shelf Hungarian BERT models, and ones pretrained on the corpus under study, we also finetune models pretrained on OCR-ed text of similar layout and typography from the same time period but with very different subject matter. We show that adaptation to the peculiarities of the OCR-ed text without the domain knowledge gives almost as much improvement on an off-the-shelf model as adaptation to the corpus under study (§5.2.2).

4. We run further comparisons with different loss functions, and numerous low-cost baseline methods (§5). First of all, we show that transformers have a clear advantage over low-cost baselines based on bag-of-words and word embedding. We point out further tendencies such as a multilabel classifier is more resistant to domain shift than individual binary classifiers, and adding confidence penalty to the BERT finetuning loss also has a beneficial effect in domain change.

1.3. Related Work

Training word embeddings on a corpus from the journal *Pártélet*, the official journal of the governing party in Hungary in the Kádár era, Ring et al. (2020) study trends in the semantic changes of notions related to decisions and control, while Szabó et al. (2021) perform a similar study for notions related to agriculture and industry.

BERT has been successfully used to learn and predict complex sequence labelling systems in several domains. Bressem et al. (2020) train models on an annotated set of chest radiology reports. They show that their best model can then predict labels on CT reports. Grandeit et al. (2020) train models on counselling reports. They conclude that out of A) the labels predicted by their best model, B) the labels given by an expert annotator and C) the labels given by a novice annotator, A) and B) are the most similar pair. Limsopatham (2021) trains models on legal documents. Out of the solutions he tested, Longformer (Beltagy et al., 2020) is shown to give the best results when being taught on long sequences. Mehta et al. (2022) train models on therapist talk-turns. They show that even when their best model cannot always correctly classify the approach used in each talk-turn, it can still reliably tell which approaches have been used during a therapy session.

With regards to measuring the impact of OCR-ed text on NLP task performance: Jiang et al. (2021a) and Jiang et al. (2021b) compare BERT contextual token embeddings on pairs of the cleaned text (Guthenberg) and OCR-ed text (HathiTrust) of the same books. Both studies find that pretraining either on clean or OCR-ed text helps performance. Labusch and Neudecker (2020) perform Named Entity Recognition and Linking on OCR-ed documents kept at the Berlin State Library. They find that pretraining BERT on historical text worsens task performance on contemporary text.

2. Dataset

2.1. Corpus: *Alföld* and *Nagyvilág*, Two Hungarian Literary Journals from the Period under Examination

When it comes to examining the status of literary translators and translation, *Nagyvilág* is the single most significant journal of the Kádár era its primary focus being on world literature and related articles. On the other hand, its scope makes any in-depth longitudinal analysis a resource-intensive task to carry out. Which is why the training set was retrieved from the journal *Alföld*, that is somewhat less relevant to literary translation, as it predominantly features Hungarian literature and related articles. The page scans of these journals, as well as those used in the domain adaptation comparisons (§5.2.2) were downloaded from the Arcanum database (Arcanum Adatbázis Kiadó Magyarország, 2023).

2.2. Manual Annotation of *Alföld*

The training set consists of a manually annotated database listing all paragraphs from *Alföld* (with the exception of pieces of or excerpts from literary works) that mention translation to any extent thematically annotated with two kinds of labels.

1. Content labels indicate what implications, connotations, themes or topics are touched upon in each paragraph in reference to translation. Each paragraph may be coded with several content labels (multilabel coding). 38 content labels are used.

2. Context labels however signal what it is in the context that warrants mentioning translation (e.g. the paragraph is about the author of a book that was translated, etc.). Each paragraph may be coded with only one context label (multiclass coding). 11 context labels are used.

It is important to clarify here that even though the two labeling systems hold some similarly named labels, the two systems are drastically different in principle. Content labels show *what themes are mentioned* in a paragraph relating to translation and context labels show *why translation is mentioned* in a paragraph in the first place. As examples and because we use them in the qualitative analysis

(§5.4), we give the definitions of the content label *author as translator* and the context label *translator* in Subsection A.1.

The project being in its pilot phase, the system and list of codes are developed and annotation is conducted by Galambos to create the training set during the first phase of the project. Content analysis and certain features of thematic analysis (Braun and Clarke, 2022) are combined to achieve as accurate and unbiased results as possible considering that all coding system adjustments and annotation are implemented by a single researcher. For this purpose, annotation is performed twice with a significant time gap between the two iterations. This helps finetuning the coding system and eliminating inconsistencies and other mistakes. Deploying only one annotator at this phase is also one of the reasons for seeking rigorous validation options, as seen in Sub-subsection 3.2.1 and Section 4.

Despite its obvious advantages regarding robustness, at this stage, using several annotators was not an option, mainly due to the project's experimental nature and given that creating the coding system was an ongoing part of the annotation process itself to test certain hypotheses about eliminating biases by not establishing a fixed set of labels beforehand but rather making the identification of labels part of the annotation phase. A potential next stage would involve further changes to the method and again, working from the bottom up with improved conditions and, undoubtedly, more annotators, heavily building on the conclusions of the pilot stage to eliminate disadvantages we have identified.

2.3. Preprocessing Pipeline: from Page Scans to Paragraph Texts

We need to transform the *Alföld* and *Nagyvilág* journal scans and the annotated paragraphs from *Alföld* to a form that a Large Language Model (LLM) can process. To this end, the scanned journal pages first need to be transformed to a sequence of paragraph texts.

We use the Tesseract (The Tesseract Authors, 2023) OCR engine via the Python Tesseract (The Python Tesseract Authors, 2022) interface. It can accurately split pages to paragraphs. However, we also need to recognize cases when a page break is also a paragraph break.

For that purpose, we apply the DBSCAN clustering algorithm (Ester et al., 1996) via its Scikit-learn (Pedregosa et al., 2011) interface to bounding box statistics. We can use this to determine

1. the type of a paragraph such as main text, footnote, and heading, and
2. whether the horizontal coordinates of a line suggest that it is the first or last line of a paragraph.

We then match the paragraphs resulting from this pipeline with the quotes in the annotation dataset using a bag-of-words-based distance. It is verified by hand that the only matching errors come from occasionally incorrectly separating paragraphs.

2.4. Dataset Statistics and Further Transformations

2.4.1. Paragraph and Word Counts

Via the preprocessing pipeline described in Subsection 2.3, we collect 9,619,240 words in 206,921 paragraphs from the *Alföld* issues of 1980–1999, and 11,622,881 words in 322,970 paragraphs from the *Nagyvilág* issues of 1980–1999. Therefore, for domain adaptation we can use a dataset with 21,242,121 words.

2.4.2. Pruning *Alföld* for the Finetuning Set

Out of the 206,921 paragraphs in the *Alföld* issues from the years 1980–1999, only 1515, that is 0.73% concern translation. On the other hand, out of these 1515 paragraphs, 1467, that is 96.83% contain the subword "fordí" (a fragment of the word "translation" in Hungarian which is "fordítás"). Therefore, by restricting the train set to the 3994 paragraphs that contain the subword "fordí", we can discard a vast amount of unneeded data while losing only a handful of relevant entries.

A further restriction comes from the architecture: in January 2023, when setting up training, there is no Hungarian LLM to our knowledge that was pretrained on suitably long sequences. Based on our preliminary experiments, we choose PULI-BERT-Large (Yang et al., 2023), which we use via the Huggingface Transformer library (Wolf et al., 2020). This model has a maximum token size of 512. Therefore, we restrict the train set to sequences of 512 tokens at most. This results in a finetuning train set of 3134 sequences. Out of these 3134 paragraphs, 1975 do not concern translation. The main reason should be the fact that the Hungarian word for translation also has other unrelated meanings. To handle these cases, we use an "unrelated" context label.

2.4.3. Label Statistics

Both the content and context labels are highly imbalanced. The mean imbalance ratio (Charte et al., 2013, §3.1), that is the average ratio of the maximal label count to label counts, is 27.34 for content labels and 36.31 for context labels. For content labels, there is further imbalance in the imbalance ratios of concurrent labels: the SCUMBLE score (Charte et al., 2014, §3.2), that is the mean Atkinson index (Atkinson, 1970) of imbalance ratios of

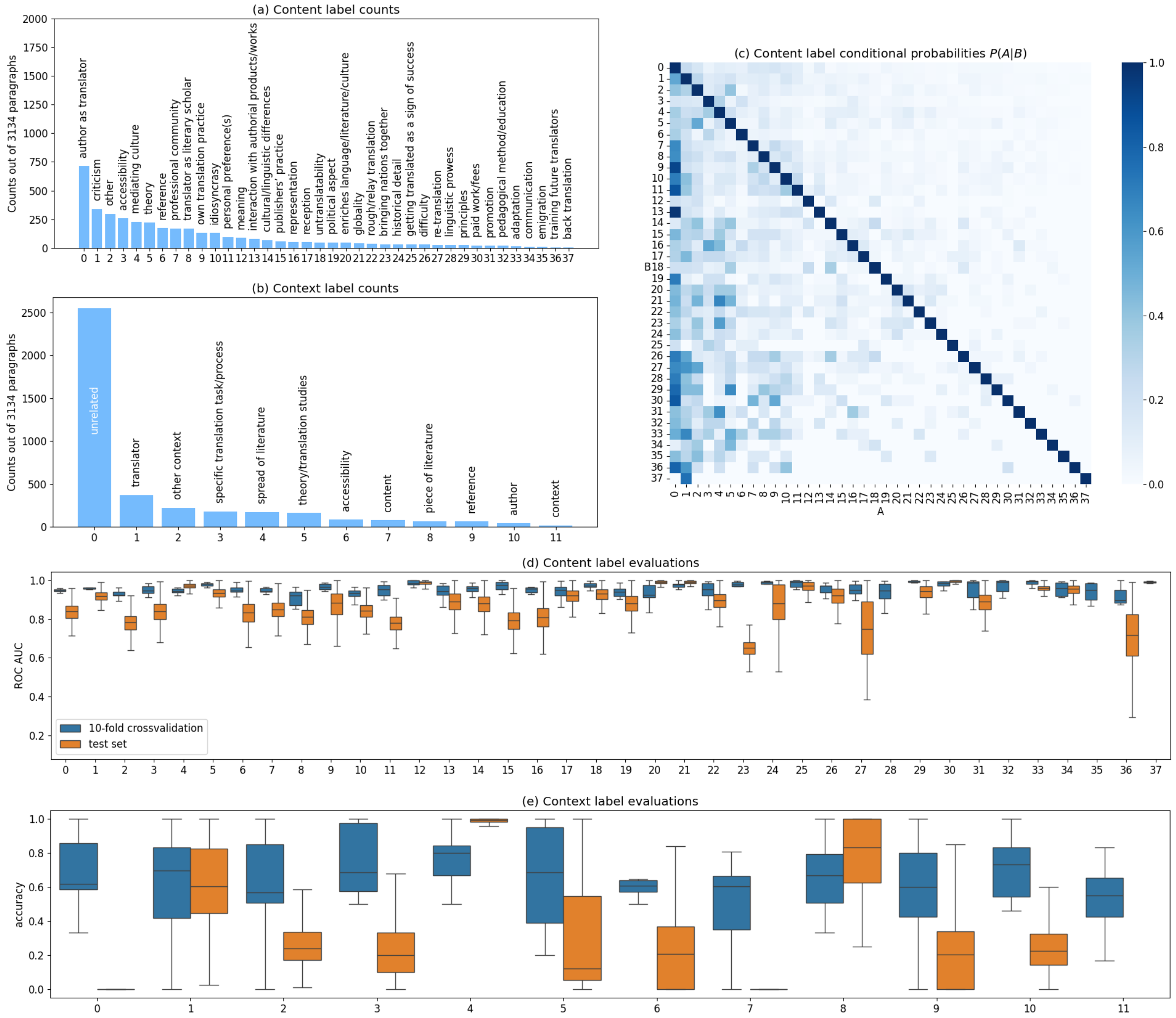

Figure 1: Dataset statistics and evaluation results. (a) Content label counts. (b) Context label counts. The context label with index 0 refers to paragraphs that contain the subword “fordí” but are unrelated to translation. (c) Content label correlations expressed as conditional probabilities. (d) Content label evaluation results by label (ROC AUC). (e) Context label evaluation results by label (accuracy). In (d) and (e), 10-fold crossvalidation results are dark blue, and test set results are orange.

labels present in a paragraph, is 0.3290. For more details, see Figure 1a-c.

2.4.4. 10-fold Stratification

We use stratification to get both the content and context label 10-folds. This is straightforward in the case of context labels, but not so for content labels. Following (Sechidis et al., 2011), we seek to find a partition where the individual label frequencies approximate those on the full dataset. Making use of GPU parallelization, we draw millions of partitions and choose one that is minimal in the reverse lexicographical ordering of individual label frequency error rates normalized by individual label frequencies.

2.4.5. Pruning and Truncating Paragraphs from *Nagyvilág* for the Target Domain

As for forming the target domain from the preprocessed *Nagyvilág* corpus, out of its 322,970 paragraphs, we select the 12,712 that contain the subword "fordí". The dataset is further filtered by discarding (i) tables of contents, (ii) references at the end of quotations, literary texts or other articles that only consist of “translated by ⟨translator⟩” and (iii) literary texts. This way, we end up with a target domain of 4589 sequences. We truncate the tokenized sequences at 512 tokens in front. See Section 4 to see how we use importance sampling to get a test set from this, which we then use to check the resistance of our models to domain shift.

3. Training

3.1. Pretraining: Domain Adaptation

We perform domain adaptation with Masked Language Modelling on the 21,242,121-word *Alföld–Nagyvilág* dataset described in Sub-subsection 2.4.1. The batch size is set to the largest value that fit in the NVIDIA A100 40GB GPU that we are using, and following Ma and Yarats (2021), the number of warmup steps is set to $\frac{2}{1-\beta_2}$ train steps, where β_2 denotes the second momentum in the AdamW optimizer (Loshchilov and Hutter, 2019). See Table 2 in the Appendix for the hyperparameters tuned. Training with the best hyperparameters that has been found brings down the perplexity score of 43.07 of the original PULI-BERT-Large model to 2.88. The cause of the magnitude of this decrease could be that the 21 million-word domain adaptation corpus is rather small in comparison to the 86 billion-word corpus the model was originally pretrained on (Yang et al., 2023, Table 1 ("1. táblázat")).

3.2. Finetuning: Imbalanced Label Classification

As described in Subsection 2.4, we work with a 3134-sequence finetuning train set with two highly imbalanced label sets: 38 content labels that are multilabel, and 12 context labels that are single label.

3.2.1. 10-fold Training and Evaluation

The small size of the train set is taken advantage of by using techniques requiring several train runs. One of these is 10-fold training. It offers 3 main benefits:

1. Consistency of evaluation scores across iterations confirm consistency of annotation.
2. More robust evaluation scores can be achieved with confidence intervals.
3. The 10 models acquired from training can be used for inference as an ensemble.

3.2.2. Population-Based Training

The other technique with several runs we use is the application of Population-Based Training (Jaderberg et al., 2017) for hyperparameter optimization. Its benefits are 2-fold:

1. It adapts hyperparameters on the fly and this way finds hyperparameter schedules on its own.
2. Since it trains the samples in parallel, it is highly scalable.

We perform this algorithm by training 100 models in parallel, epoch by epoch. We train them for 30 epochs for the content labels, and at least 30 epochs until there is no improvement for 10 epochs for the context labels. In our version of selection:

1. the top 10% elite is kept unchanged, and
2. roulette wheel selection is used for the rest with hyperparameter perturbation.

In contrast to Jaderberg et al. (2017), in perturbation, we do not choose between the multipliers 0.8 and 1.2, but pick a multiplier uniformly from the interval $[0.8, 1.2]$. We make these changes to have less rigid heuristics.

See Table 2 in the Appendix for the hyperparameters tuned.

3.2.3. Content Label Finetuning

In order to avoid fixing a threshold, and for its robustness to label imbalance, we use macro averaged ROC AUC as evaluation metric. Moreover, we use focal loss (Lin et al., 2017) as train loss. This results in a very satisfactory average ROC AUC of 0.9524 ± 0.0114 (we report all of the confidence intervals with confidence level 95%). See more detailed evaluation results in Figure 1d. Note that the less frequent labels do not get lower scores. We use the unweighted version (Lin et al., 2017, Equation 4), as the weighted one (Lin et al., 2017, Equation 5) does not bring any improvement.

3.2.4. Context Label Finetuning

The focus being resistance to label imbalance here as well, we use balanced accuracy as evaluation metric. As training loss, we combine label distribution-aware margin loss (Cao et al., 2019) with a penalty for confident output distributions (Pereyra et al., 2017). This gives a balanced accuracy of 0.6357 ± 0.1266. See Figure 1e for more detailed evaluation results, and Sub-subsection 5.2.3 for comparisons with other loss functions.

Note that separate model ensembles are trained in the content and the context label case. We experiment with training combined models for the two label sets, but with significantly worse results.

4. Evaluation on the Target Domain

To measure if our model ensemble can successfully carry over the two coding systems to the target domain, we draw a sample from it. Manually annotating this, we get a test set. Note that our models never see the labels on the test set.

4.1. Deciding the Size of the Test Set

As manual validation is highly resource-intensive, when deciding on how many samples we should draw for the test set from the target domain, for a prospective sample size, we seek to estimate what confidence interval is to be expected. To that end, simulations are run on the results of the 10-fold crossvalidation: for both content and context labels, $\texttt{sample\_size} = 50, 60, \dots, 300$ and each fold evaluation set, 100 times, we draw a random sample of `sample_size` from the fold evaluation set, and via bootstrapping with size 10,000 the standard deviation of the relevant metric is approximated, see Figure 2. This we can use to estimate what confidence interval we would get from what sample size. Based on this data, we determine that going from 50 samples to 100 for a confidence interval decrease of about 20% is worthwhile, however, going up to 150 for a further confidence interval decrease of about 10% is not. Therefore a decision is made to draw 100 samples from the target domain for manual validation.

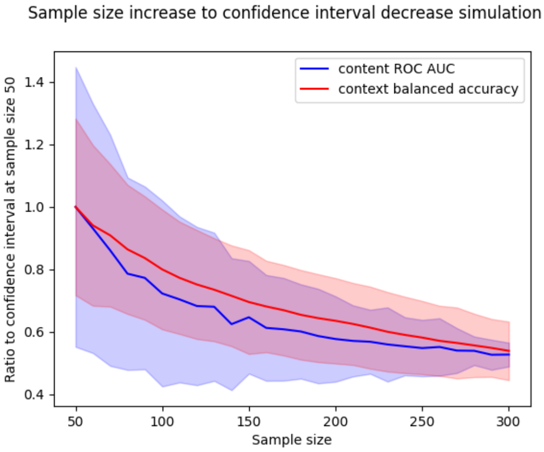

Figure 2: Sample size increase to confidence interval decrease simulation via bootstrap.

4.2. Importance Sampling on the Target Domain

As we naturally do not have labels for the target domain, only model prediction probabilities, a stratified sample for the test set cannot be used. Therefore, to address label imbalance, we opt for an importance sampling approach, that draws paragraphs with less frequent label prediction probabilities with higher probability.

4.3. Validation Results

On the content label set, we achieve a ROC AUC of 0.8757 ± 0.0252 (to get confidence intervals and box plots for the validation results, we use bootstrapping on 10,000 samples). For more detailed scores, see Figure 1d. This means that the model ensemble is capable of reliably carrying over the labeling to the target domain.

On the other hand, on the context label set, we only reach a balanced accuracy of 0.3287 ± 0.1297. For more detailed scores, see Figure 1e. We perform a preliminary investigation on the possible reasons for misclassification in Subsection 5.4. Based on this result, from *Nagyvilág*, only content labels are used in the findings.

5. Comparisons

5.1. Baseline Methods

We test baseline methods with low computational cost. We discuss transforming paragraphs to tabular data in Sub-subsection 5.1.1, resampling algorithms in Sub-subsection 5.1.2, gradient boosted tree algorithms in Sub-subsection 5.1.3 and All of the options are extensively tuned using the BlendSearch algorithm.

5.1.1. Transforming Paragraphs to Tabular Data

As most low-cost machine learning methods work on tabular data, we first need to transform the paragraphs to numerical vectors. For a bag-of-words-based approach, we test using TF-IDF vectors, via the `scikit-learn` (Pedregosa et al., 2011) implementation. For a low-cost word embedding-based approach, we test using the sentence vectors output by the fastText word representation model (Bojanowski et al., 2017), pretrained on Hungarian Common Crawl and Wikipedia, that is available on their webpage (fastText Authors, 2020). In both cases, dimension reduction is performed using the TruncatedSVD (Halko et al., 2011) algorithm. We also test the built-in fastText classifier (Joulin et al., 2017).

5.1.2. Resampling the Train Set to Make it more Balanced

To make the train set more balanced, we use resampling. On the context label dataset, we test the ADASYN (He et al., 2008) and SMOTE (Chawla et al., 2002) synthetic oversampling algorithms, via their `imbalanced-learn` (Lemaître et al., 2017) implementation. On the content label dataset, we test the REMEDIAL-HwR (Charte et al., 2019) and MLSOL (Liu and Tsoumakas, 2020) synthetic resampling algorithms. We change both algorithms:

REMEDIAL-HwR is in fact the composition of the REMEDIAL (Charte et al., 2019, Algorithm 4) and

Classifier	Re-sampler	Content labels (ROC AUC)		Context labels (balanced accuracy)	
		10-fold crossvalidation	test set	10-fold crossvalidation	test set
Based on TF-IDF vectors					
CatBoost[1]	ADASYN			0.3372 ± 0.0719	0.1365 ± 0.1027
CatBoost	MLSOL	0.8144 ± 0.0535	0.6822 ± 0.0376		
CatBoost	R-HwR[2]	0.8100 ± 0.0496	**0.6937 ± 0.0375**		
CatBoost	SMOTE			0.3469 ± 0.0951	0.1188 ± 0.1015
LightGBM	ADASYN			0.3433 ± 0.0873	0.1180 ± 0.0876
LightGBM	SMOTE			0.3454 ± 0.0720	0.1737 ± 0.1068
XGBoost[3]	ADASYN	**0.8530 ± 0.0255**	0.6703 ± 0.0347	**0.3629 ± 0.0761**	**0.1939 ± 0.1156**
XGBoost	SMOTE			0.3500 ± 0.1025	0.1432 ± 0.0951
Based on fastText vectors					
CatBoost	ADASYN			0.3309 ± 0.0495	0.1573 ± 0.1014
CatBoost	MLSOL	0.8475 ± 0.0418	0.7594 ± 0.0313		
CatBoost	R-HwR	0.8412 ± 0.0396	**0.7797 ± 0.0318**		
CatBoost	SMOTE			**0.3345 ± 0.0605**	0.1430 ± 0.1029
fastText		0.7330 ± 0.0383	0.6753 ± 0.0439	0.2536 ± 0.0607	**0.1780 ± 0.1204**
LightGBM	ADASYN			0.3157 ± 0.0462	0.1308 ± 0.0821
LightGBM	SMOTE			0.3274 ± 0.0855	0.1530 ± 0.0937
XGBoost	ADASYN	**0.8702 ± 0.0256**	0.7176 ± 0.0363	0.3333 ± 0.0983	0.1336 ± 0.0677
XGBoost	SMOTE			0.3162 ± 0.0816	0.1330 ± 0.0971
BERT training methods					
Model	Domain Adaptation	Content label results with Focal Loss		Context label results with LDAM + CP loss	
huBERT	Corpus	0.9503 ± 0.0175	0.8727 ± 0.0268	0.6173 ± 0.0872	0.3032 ± 0.1422
huBERT	Extended	0.9468 ± 0.0064	0.8702 ± 0.0362	0.6187 ± 0.0486	0.3255 ± 0.1285
huBERT	OCR	0.9478 ± 0.0113	0.8737 ± 0.0268	0.6004 ± 0.1147	0.2788 ± 0.1175
huBERT	None	0.9093 ± 0.2135	0.8711 ± 0.0263	0.5584 ± 0.1227	**0.3671 ± 0.1355**
PULI[4]	Corpus	0.9524 ± 0.0114	0.8757 ± 0.0252	0.6357 ± 0.1266	0.3287 ± 0.1297
PULI	Extended	**0.9534 ± 0.0065**	**0.8808 ± 0.0312**	**0.6410 ± 0.0481**	0.2672 ± 0.1332
PULI	OCR	0.9481 ± 0.0139	0.8767 ± 0.0252	0.6265 ± 0.0842	0.3560 ± 0.1487
PULI	None	0.9193 ± 0.0673	0.8513 ± 0.0280	0.5545 ± 0.1422	0.2752 ± 0.1416
Domain Adapted Model	Loss				
PULI	Focal			0.5985 ± 0.0991	0.3428 ± 0.1431
PULI	LDAM			0.6263 ± 0.0938	0.2382 ± 0.1356
Double Context Length Domain Adapted Model					
huBERT				0.6048 ± 0.0318	0.3460 ± 0.1312
Single Model with no Domain Adaptation					
huBERT				0.6029	0.3637 ± 0.1365
PULI				0.5751	0.3471 ± 0.1307
Llama 2				0.4009	0.2605 ± 0.1296

Table 1: Comparisons on baseline models and BERT training methods. [1]In the case of CatBoost on content labels, a single multilabel classifier is tuned. [2]Short for REMEDIAL-HwR. [3]In the case of XGBoost on content labels, individual binary classifiers are tuned for each label. [4]Short for PULI-BERT-Large.

MLSMOTE (Charte et al., 2015, Algorithm 1) algorithms. In REMEDIAL, entries with a SCUMBLE score higher than a tuneable threshold are selected for decoupling. In MLSOL on the other hand, entries with an IRLbl score larger than the mean are selected to serve as synthetic instance sources. We replace the mean IRLbl score with a tuneable threshold.

The inbalance metric of MLSOL is based on the *local imbalance matrix* C_{ij} (Liu and Tsoumakas, 2020, Equation 1). To decide the labels of the synthetic entries, a threshold θ is used (*ibid.*, Algorithm 3, line 17). In the paper, this threshold is determined by hardcoded rules (*ibid.*, lines 12-16). We let the threshold linearly depend on local label imbalance: $\theta = \frac{1+C_{ij}}{2}$.

5.1.3. Using Gradient Boosted Tree Algorithms on the Resampled Train Set

Based on their performance in the low-cost domain, we train CatBoost (Prokhorenkova et al., 2018), LightGBM (Ke et al., 2017) and XGBoost (Chen and Guestrin, 2016) models on the resampled context, that is multiclass dataset. As by 2023 October only CatBoost has stable multilabel classification support, we only use that to train multilabel models. We moreover pick the combination that performs best on context labels: XGBoost + ADASYN, to train 38 individual binary classification models on each content label. Note that in the latter case tuning is also performed separately for each binary classifier.

5.1.4. Discussion of Baseline Results

On content labels, we see an advantage of word embedding (fastText) vectors over bag-of-words (TF-IDF) ones. This could be attributed to the fact that content labels are based on more local information. On the other hand, on context labels, we have a somewhat better result using bag-of-words. This could be due to the tendency that the information expressed in a bag-of-words vector, albeit more reduced, is more balanced in terms of influence by individual words.

As content labels are multilabel, one can also compare training individual binary classifiers, one for each label, to training only one multilabel classifier. We use XGBoost and CatBoost for the two respective approaches. Whichever of bag-of-words or word embedding vector-based feature vectors we use, on source domain 10-fold crossvalidation, one can notice a slight advantage of training individual binary classifiers, but on the target domain test set, one can observe a more pronounced advantage of training a unique multilabel classifier. Learning label correlations may help robustness.

The fastText classifier has significantly worse results in all respects besides context label results on the target domain test set.

5.2. BERT Training Methods

In this subsection, we detail different aspects of the training that we test with a number of options. As the best training procedure has already been described in Section 3, here we discuss results right after explaining the component that we change, and not in a separate sub-subsection.

5.2.1. Pretrained Models

We test two Hungarian pretrained models. The earlier model, huBERT (Nemeskey, 2021) has a BERT-Base architecture, and it was trained for 189,000 steps on the Hungarian WebCorpus 2.0 (Nemeskey, 2020), that was built from Common Crawl and includes a little over 9 billion words. PULI-BERT-Large (Yang et al., 2023) has a BERT-Large architecture, and it was trained for 750,000 steps on a corpus assembled from the Hungarian WebCorpus 2.0, the Hungarian Wikipedia and a number of other resources, totalling 36,285,941,699 words. The two models seem to perform very similarly on content labels. This seems to indicate that for the content label classification problem a BERT-Base model is big enough. On context label 10-fold results on the other hand, we see a slight advantage of the larger model. The context label test set results appear noisy, without further study, a clear explanation does not seem possible.

5.2.2. Domain Adaptation

We evaluate finetuning after four different options for domain adaptation:

1. No domain adaptation.
2. Domain adaptation on the corpus under study, that is the 1980–1999 issues of *Alföld* and *Nagyvilág*.
3. A corpus of similar size of OCR-ed journals of similar layout and typography from the same time period, but with entirely different subjects. See Section C for the list of the journals.
4. Domain adaptation on an extended contextual corpus consisting of the 1960–2021 issues of *Alföld* and the 1960–2015 issues of *Nagyvilág*.

Based on our results, adaptation to domain text gives the most perfomance boost, in particular, more contextual data is yet better. However, adaptation to the peculiarities of OCR-ed text is almost as effective, and still significantly better than no adaptation.

5.2.3. Context Label Losses

As discussed above, we finetune the domain adapted PULI-BERT-Large on the content label set with Focal Loss (Lin et al., 2017), and this gives a very satisfactory result. On the other hand, on the context label set we want to see if we can improve our result. Therefore, we test Label Distribution-Aware Margin (LDAM) loss (Cao et al., 2019) with and without a Confident output distribution Penalty (CP) (Pereyra et al., 2017). Based on our results, LDAM gives better 10-fold crossvalidation results already on its own, but CP improves the test set results significantly. This may indicate that the regularization effect of CP helps robustness.

5.2.4. Double Context Length

As one of the possible reasons for the inferior performance on context labels is that that task requires a larger context length, we experiment with domain adaptation and finetuning with a modified huBERT model, where instead of the original context length of 512, we use 1024. As (for an undisclosed reason) BERT models use learned positional embeddings, following (Beltagy et al., 2020, §5) to extend these to positions $512, \ldots, 1023$, we copy the 512 embeddings twice. In the end, the results do not improve.

5.3. Llama 2

Preliminary studies on the performance of the open family of foundation and chat models Llama 2 (Touvron et al., 2023) are conducted.

5.3.1. Chain-of-Thought Few Shot Learning

As even evaluating examples is resource-intensive, we only experiment with the content label *author as translator*. We prompt the model with the task description and some randomly chosen examples with chain-of-thought (CoT) explanations (Wei et al., 2022), and tell it to do the same for an additional paragraph.

It does generate its answers according to the CoT pattern it received, but the linguistic and factual knowledge required to answer questions of this complexity seem to be missing. We do not observe a difference in this respect between the chat models of different sizes. Again, this is an exploratory experiment. It is quite possible that for example with instruction finetuning, better results can be achieved by a generative model.

5.3.2. Finetuning on the Context Label Set

We also try finetuning the smallest, 7-billion-parameter foundation model on the context label set. This is computationally very intensive: Even with QLoRA (Dettmers et al., 2023) and Distributed Data Parallel in its `accelerate` implementation (Gugger et al., 2022) with 5 NVIDIA A100 40GB GPU an epoch takes 20 times as long as training PULI on 1 GPU.

Therefore, instead of 10-fold crossvalidation with Population-Based Training, we opt for BlendSearch on a single train-validation split. See Table 2 for the hyperparameter initial distributions. We run the tuning algorithm for 3 days on the 5 GPUs. For a fairer comparison, we finetune the original huBERT and PULI models for 12 hours on 1 GPU the same way. Still, the BERT models surpass Llama 2.

5.4. Qualitative Analysis

We go through the test set and try to explain from the text the misclassifications of the content label *author as translator* and the context label *translator* by the PULI model domain adapted on the main corpus and finetuned with LDAM+CP loss. Here, in the main text, we summarize our findings. See Subsection A.2 for more details.

In several cases, the model's predictions indicate manual annotation mistakes. Moreover, some potential sources for model misclassification are 1. data scarcity 2. inadequate context window 3. some very interesting patterns that – according to our hypothesis – do not match previous patterns from the source domain regarding how translators are represented. Again, see Subsection A.2 for more details.

6. Conclusion

We study 2 complex coding systems which were developed on paragraphs of a Hungarian literary journal to track trends in the social perception of literary translation: a multilabel content label set, and a multiclass context label set. Although both label sets are highly imbalanced, we show that with extensive hyperparameter tuning and loss functions robust to imbalance it is possible to teach BERT models both label sets. This result is verified with 10-fold crossvalidation. We further investigate if the resulting ensemble of models is capable of carrying over the coding systems to another literary journal. To that end, we introduce a method to estimate the confidence interval of evaluation results on a test set sampled on the target domain with a given sample size. With this, we verify that our ensemble of models can fulfill this task in the case of content labels. We conduct numerous comparisons to low-cost baseline methods and variations in our BERT training procedure. In particular, we show that domain adaptation to OCR-ed text of distinct subject matter already significantly helps task performance.

7. Acknowledgements

The authors would like to thank Gábor Berend, Adrián Csiszárik, Péter Kőrösi-Szabó, Anikó Sohár and Dániel Varga for their insight and many helpful discussions. We are also grateful to the reviewers for their valuable remarks and suggestions. Zsámboki is supported by the Ministry of Innovation and Technology NRDI Office within the framework of the Artificial Intelligence National Laboratory (RRF-2.3.1-21-2022-0004) and the ELTE TKP 2021-NKTA-62 funding scheme.

8. Ethical Statement: Carbon Footprint

We estimate that in total, experiments related to this project took 16 months, that is 11520 NVIDIA A100 40GB GPU hours. The Machine Learning Emissions Calculator by Lacoste et al. (2019) estimates that this emitted 1244.16 kg CO_2eq. Based on data published in Our World in Data (in Data, 2019), travelling as a passenger 7318.59 km in a car, 6220.8 km on a flight or 12826.39 km by bus would emit a similar amount.

9. Bibliographical References

Anthony B Atkinson. 1970. On the measurement of inequality. *Journal of Economic Theory*, 2(3):244–263.

Iz Beltagy, Matthew E. Peters, and Arman Cohan. 2020. Longformer: The long-document transformer. *arXiv:2004.05150*.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

Virginia Braun and Victoria Clarke. 2022. *Thematic Analysis: a practical guide*. SAGE, London; Thousand Oaks, California.

Keno K Bressem, Lisa C Adams, Robert A Gaudin, Daniel Tröltzsch, Bernd Hamm, Marcus R Makowski, Chan-Yong Schüle, Janis L Vahldiek, and Stefan M Niehues. 2020. Highly accurate classification of chest radiographic reports using a deep learning natural language model pre-trained on 3.8 million text reports. *Bioinformatics*, 36(21):5255–5261.

Kaidi Cao, Colin Wei, Adrien Gaidon, Nikos Arechiga, and Tengyu Ma. 2019. Learning imbalanced datasets with label-distribution-aware margin loss. In *Advances in Neural Information Processing Systems*, pages 1567–1578.

Francisco Charte, Antonio Rivera, María José del Jesus, and Francisco Herrera. 2013. A first approach to deal with imbalance in multi-label datasets. In *Hybrid Artificial Intelligent Systems*, pages 150–160, Berlin, Heidelberg. Springer Berlin Heidelberg.

Francisco Charte, Antonio Rivera, María José del Jesus, and Francisco Herrera. 2014. Concurrence among imbalanced labels and its influence on multilabel resampling algorithms. In *Hybrid Artificial Intelligence Systems*, pages 110–121, Cham. Springer International Publishing.

Francisco Charte, Antonio J. Rivera, María J. del Jesus, and Francisco Herrera. 2015. Mlsmote: Approaching imbalanced multilabel learning through synthetic instance generation. *Knowledge-Based Systems*, 89:385–397.

Francisco Charte, Antonio J. Rivera, María J. del Jesus, and Francisco Herrera. 2019. Remedial-hwr: Tackling multilabel imbalance through label decoupling and data resampling hybridization. *Neurocomputing*, 326-327:110–122.

Nitesh V. Chawla, Kevin W. Bowyer, Lawrence O. Hall, and W. Philip Kegelmeyer. 2002. Smote: Synthetic minority over-sampling technique. *J. Artif. Int. Res.*, 16(1):321–357.

Tianqi Chen and Carlos Guestrin. 2016. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, page 785–794, New York, NY, USA. Association for Computing Machinery.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2023. Qlora: Efficient finetuning of quantized llms. In *Advances in Neural Information Processing Systems*, volume 36, pages 10088–10115. Curran Associates, Inc.

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)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. 1996. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, KDD'96, page 226–231. AAAI Press.

The fastText Authors. 2020. Pre-trained word vectors for 157 languages. `https://github.com/facebookresearch/fastText/blob/master/docs/crawl-vectors.md`, Last accessed on 2023-10-18.

Dalma Galambos. 2024. *Title TBA*. Ph.D. thesis, Pázmány Péter Catholic University. Under revision.

Philipp Grandeit, Carolyn Haberkern, Maximiliane Lang, Jens Albrecht, and Robert Lehmann. 2020. Using BERT for qualitative content analysis in psychosocial online counseling. In *Proceedings of the Fourth Workshop on Natural Language Processing and Computational Social Science*, pages 11–23, Online. Association for Computational Linguistics.

Sylvain Gugger, Lysandre Debut, Thomas Wolf, Philipp Schmid, Zachary Mueller, Sourab Mangrulkar, Marc Sun, and Benjamin Bossan. 2022. Accelerate: Training and inference at scale made simple, efficient and adaptable. `https://github.com/huggingface/accelerate`.

N. Halko, P. G. Martinsson, and J. A. Tropp. 2011. Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions. *SIAM Review*, 53(2):217–288.

Haibo He, Yang Bai, Edwardo A. Garcia, and Shutao Li. 2008. Adasyn: Adaptive synthetic sampling approach for imbalanced learning. In *2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence)*, pages 1322–1328.

Our World in Data. 2019. Carbon footprint of travel per kilometer. `https://ourworldindata.org/grapher/carbon-footprint-travel-mode`, Last accessed on 2023-08-06.

Max Jaderberg, Valentin Dalibard, Simon Osindero, Wojciech M. Czarnecki, Jeff Donahue, Ali Razavi, Oriol Vinyals, Tim Green, Iain Dunning, Karen Simonyan, Chrisantha Fernando, and Koray Kavukcuoglu. 2017. Population based training of neural networks.

Ming Jiang, Yuerong Hu, Glen Worthey, Ryan C Dubnicek, Ted Underwood, and J Stephen Downie. 2021a. Evaluating bert's encoding of intrinsic semantic features of ocr'd digital library collections. In *2021 ACM/IEEE Joint Conference on Digital Libraries (JCDL)*, pages 308–309.

Ming Jiang, Yuerong Hu, Glen Worthey, Ryan C Dubnicek, Ted Underwood, and J Stephen Downie. 2021b. Impact of ocr quality on bert embeddings in the domain classification of book excerpts. In *CEUR Workshop Proceedings*, volume 2989, pages 266–279. CEUR-WS.

Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. 2017. Bag of tricks for efficient text classification. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 427–431. Association for Computational Linguistics.

Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. 2017. Lightgbm: A highly efficient gradient boosting decision tree. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Kai Labusch and Clemens Neudecker. 2020. Named entity disambiguation and linking on historic newspaper ocr with bert. In *Conference and Labs of the Evaluation Forum (CLEF 2020)*, volume 2696 of *Conference and Labs of the Evaluation Forum (CLEF 2020)*.

Alexandre Lacoste, Alexandra Luccioni, Victor Schmidt, and Thomas Dandres. 2019. Quantifying the carbon emissions of machine learning. *arXiv:1910.09700*.

Guillaume Lemaître, Fernando Nogueira, and Christos K. Aridas. 2017. Imbalanced-learn: A python toolbox to tackle the curse of imbalanced datasets in machine learning. *Journal of Machine Learning Research*, 18(17):1–5.

Nut Limsopatham. 2021. Effectively leveraging BERT for legal document classification. In *Proceedings of the Natural Legal Language Processing Workshop 2021*, pages 210–216, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. 2017. Focal loss for dense object detection. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 2999–3007.

Bin Liu and Grigorios Tsoumakas. 2020. Synthetic oversampling of multi-label data based on local label distribution. In *Machine Learning and Knowledge Discovery in Databases*, pages 180–193, Cham. Springer International Publishing.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *2019 International*

Conference on Learning Representations (ICLR). ArXiv:1711.05101.

Jerry Ma and Denis Yarats. 2021. On the adequacy of untuned warmup for adaptive optimization. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(10):8828–8836.

Maitrey Mehta, Derek Caperton, Katherine Axford, Lauren Weitzman, David Atkins, Vivek Srikumar, and Zac Imel. 2022. Psychotherapy is not one thing: Simultaneous modeling of different therapeutic approaches. In *Proceedings of the Eighth Workshop on Computational Linguistics and Clinical Psychology*, pages 47–58, Seattle, USA. Association for Computational Linguistics.

Dávid Márk Nemeskey. 2021. Introducing `huBERT`. In *XVII. Magyar Számítógépes Nyelvészeti Konferencia (MSZNY2021)*, pages 3–14, Szeged.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Gabriel Pereyra, George Tucker, Jan Chorowski, Łukasz Kaiser, and Geoffrey Hinton. 2017. Regularizing neural networks by penalizing confident output distributions. In *ICLR Workshop*.

Liudmila Prokhorenkova, Gleb Gusev, Aleksandr Vorobev, Anna Veronika Dorogush, and Andrey Gulin. 2018. Catboost: unbiased boosting with categorical features. In *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc.

Orsolya Ring, Zoltán Kmetty, Martina Katalin Szabó, László Kiss, Balázs Nagy, and Veronika Vincze. 2020. Kulcsfogalmak jelentésváltozása a kádár-korszak politikai diskurzusában. In *Magyar Számítógépes Nyelvészeti Konferencia*, pages 333–342.

Konstantinos Sechidis, Grigorios Tsoumakas, and Ioannis Vlahavas. 2011. On the stratification of multi-label data. In *Machine Learning and Knowledge Discovery in Databases*, pages 145–158, Berlin, Heidelberg. Springer Berlin Heidelberg.

Martina Katalin Szabó, Orsolya Ring, Balázs Nagy, László Kiss, Júlia Koltai, Gábor Berend, László Vidács, Attila Gulyás, and Zoltán Kmetty. 2021. Exploring the dynamic changes of key concepts of the hungarian socialist era with natural language processing methods. *Historical Methods: A Journal of Quantitative and Interdisciplinary History*, 54(1):1–13.

The Python Tesseract Authors. 2022. Python tesseract. `https://github.com/madmaze/pytesseract`, Last accessed on 2023-07-31.

The Tesseract Authors. 2023. Tesseract. `https://github.com/tesseract-ocr/tesseract`, Last accessed on 2023-07-31.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. Llama 2: Open foundation and fine-tuned chat models.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 6000–6010, Red Hook, NY, USA. Curran Associates Inc.

Chi Wang, Qingyun Wu, Silu Huang, and Amin Saied. 2021. Economical hyperparameter optimization with blended search strategy. In *ICLR*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, brian ichter, Fei Xia, Ed Chi, Quoc V Le, and Denny Zhou. 2022. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*, volume 35, pages 24824–24837. Curran Associates, Inc.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi 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 Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Zijian Győző Yang, Réka Dodé, Gergő Ferenczi, Enikő Héja, Kinga Jelencsik-Mátyus, Ádám Kőrös, László János Laki, Noémi Ligeti-Nagy, Noémi Vadász, and Tamás Váradi. 2023. Jönnek a nagyok! bert-large, gpt-2 és gpt-3 nyelvmodellek magyar nyelvre. In *XIX. Magyar Számítógépes Nyelvészeti Konferencia (MSZNY 2023)*, pages 247–262, Szeged, Hungary. Szegedi Tudományegyetem, Informatikai Intézet.

10. Language Resource References

Arcanum Adatbázis Kiadó Magyarország. 2023. Arcanum digitális tudománytár. `https://adtplus.arcanum.hu/`, Last accessed on 2023-07-31.

Dávid Márk Nemeskey. 2020. *Natural Language Processing methods for Language Modeling*. Ph.D. thesis, Eötvös Loránd University. `https://hlt.bme.hu/hu/publ/nemeskey_2020`.

A. Label Examples

Below, we give the definition of two of the labels, the content label *author as translator* and the context label *translator*. Afterwards, we provide the details of the qualitative analysis that was described in Subsection 5.4.

A.1. Definitions

The content label *author as translator* means that in the paragraph, somewhere a translator is mentioned who is more known by their work as an author. There can be many other content labels applied to the paragraph in question, the part labeled *author as translator* can be a minor detail and it is also important that it is about the fact that the person mentioned is more famous about something other than translation.

The context label *translator* on the other hand means that the *reason* translation in that paragraph is mentioned is that the topic is a specific translator for any reason at all. The depth of the discussion, other themes or the way translation is mentioned are irrelevant here.

There can be correlation between these two labels but in spite of the similar themes they explore, they do not necessarily occur together and there is no overlap in their function.

A.2. Qualitative Analysis Details

A.2.1. Author as Translator

For this binary label, we view a paragraph as positive if the average label probability by the model ensemble is larger than 50%. This holds for 18 paragraphs out of 100. Out of these 18 misclassifications, 6 turn out to be a mistake in manual annotation. This gives a strong indication as to how helpful our tool can be for annotation.

Of the rest, in 2 false negative cases we hypothesize that it is mostly the name of the translator that indicates the validity of the label and it is a relatively less well-known name that might not have been frequently present in the rest of the corpus.

We also noticed that in 4 false positive cases, the translator in question is most known for their work as a translator, however, the discourse exhibits traits rarely displayed without the *author as translator* label in the corpus according to our hypothesis based on the train set. These are the following: 1. details of the translator being famous, i.e. winning prizes and having a prestigious portfolio and 2. writing extensively/being interviewed about themselves or their practice.

What is even more intriguing is that each of these four instances are about translators who work from Hungarian. We very tentatively pose the hypothesis that translating from Hungarian is an activity that could be viewed more important in the Hungarian discourse because it leads to the visibility and representation of Hungarian literature and that could be a more personal matter in the Hungarian literary field than the accessibility and representation of foreign literature in Hungary. This, however, requires further investigation.

A.2.2. Translator

Here, out of the 15 misclassifications, 2 turned out to be manual annotation mistakes. Of the rest, in 3 cases, it is possible that the broader context cannot be inferred from the paragraph and would require knowledge of a larger or different portion of the text. In another 3 cases, although the paragraph does extensively feature the translation activity of a single individual, the broader context is not about the translator.

B. Hyperparameter search spaces

In the following table, the hyperparameter initial distributions used in the domain adaptation Blend-

Search (§3.1) and finetuning Population-Based Training (§3.2.2) are provided. For an interval $I \subseteq \mathbf{R}$, let $\ell\mathcal{U}$ denote the log uniform distribution $\exp(\mathcal{U}\log(I))$, and let $d\ell\mathcal{U}I$ denote the discrete log uniform distribution $\lfloor \ell\mathcal{U}X \rfloor$.

C. Constituents of the OCR Corpus

The OCR corpus consists of the 1980–1999 issues of the following journals:

1. *Akadémiai Közlöny* (later *Akadémiai Értesítő*) was the bulletin of the Hungarian Academy of Sciences during the period under examination.

2. *Állam és Igazgatás* (later *Magyar Közigazgatás*) was a social studies journal specializing in public administration.

3. *Gyermekgyógyászat* was the journal of the Pediatrists Group in the Medical and Sanitary Workers Union.

Hyperparameter	Initial Distribution
AdamW first moment, β_1 in (Loshchilov and Hutter, 2019, Algorithm 2)[1][2][3]	$1 - \ell\mathcal{U}[10^{-2}, \frac{1}{2}]$
AdamW second moment, β_2 in (Loshchilov and Hutter, 2019, Algorithm 2)[1][2][3]	$1 - \ell\mathcal{U}[10^{-4}, 10^{-1}]$
Complexity, C in (Cao et al., 2019, Equation 11)[3]	$\ell\mathcal{U}[10^{-2}, 10^{2}]$
Confidence penalty strength, β in (Pereyra et al., 2017, §3)[3]	$\ell\mathcal{U}[10^{-2}, 10^{2}]$
Focusing parameter, γ in (Lin et al., 2017, Equation 4)[2]	$\ell\mathcal{U}[2^{-4}, 2^{4}]$
Learning rate[1][2][3]	$\ell\mathcal{U}[10^{-5}, 10^{-2}]$
Learning Rate scheduler (after warmup)[1]	$\mathcal{U}\{\mathrm{constant}, \mathrm{cosine}, \mathrm{linear}\}$
Maximum gradient norm[1][2][3]	$\ell\mathcal{U}[10^{-2}, 10^{2}]$
Maximum train epochs[1]	$d\ell\mathcal{U}[1, 101)$
Weight decay rate, λ in (Loshchilov and Hutter, 2019, Algorithm 2)[1][2][3]	$\ell\mathcal{U}[10^{-4}, 1]$

Table 2: Hyperparameter initial distributions. [1] Used in domain adaptation. [2] Used in content label finetuning. [3] Used in context label finetuning.