Using Ensemble Learning in Language Variety Identification

Mihaela Găman
Department of Computer Science
University of Bucharest
14 Academiei, Bucharest, Romania
mp.gaman@gmail.com

Abstract

The present work describes the solutions proposed by the UnibucNLP team to address the closed format of the DSL-TL task featured in the tenth VarDial Evaluation Campaign. The DSL-TL organizers provided approximately 11 thousand sentences written in three different languages and manually tagged with one of 9 classes. Out of these, 3 tags are considered *common label* and the remaining 6 tags are *variety-specific*. The DSL-TL task features 2 subtasks: *Track 1* - a three-way and *Track 2* - a two-way classification per language. In *Track 2* only the variety-specific labels are used for scoring, whereas in *Track 1* the common label is considered as well. Our team participated in both tracks, with three ensemble-based submissions for each. The meta-learner used for *Track 1* is XGBoost and for *Track 2*, Logistic Regression. With each submission, we are gradually increasing the complexity of the ensemble, starting with two shallow, string-kernel based methods. To the first ensemble, we add a convolutional neural network for our second submission. The third ensemble submitted adds a fine-tuned BERT model to the second one. In *Track 1*, ensemble three is our highest ranked, with an $F1 - score$ of 53.18%; 5.36% less than the leader. Surprisingly, in *Track 2* the ensemble of shallow methods surpasses the other two, more complex ensembles, achieving an $F1 - score$ of 69.35%.

1 Introduction

Discriminating between Similar Languages using a manually annotated data set of True Labels (Zampieri et al., 2023) was included on the list of shared tasks in the tenth VarDial evaluation campaign (Aepli et al., 2023), under the DSL-TL acronym. The topic of discriminating among language varieties and similar languages has been addressed in previous VarDial editions (Zampieri et al., 2017; Malmasi et al., 2016; Zampieri et al., 2015, 2014). However, we find the DSL-TL task compelling as it introduces qualitative human-annotations from multiple sources.

In DSL-TL organizers provide a set of sentences coming from news reports[1] written in either English, Spanish or Portuguese and split in a train, development and test subsets. The test split represents a collection of unlabelled sentences, with labels being subject to further submissions from participants. The examples in the train and development sets are tagged with one of nine labels, namely *EN-GB*, *EN-US*, *EN*, *ES-ES*, *ES-AR*, *ES*, *PT-PT*, *PT-BR* and *PT*. Six of the labels provided, aside from the language itself, also specify the language variety, marked with the initials of the country (i.e. *GB* – Great Britain, *US* – USA, *ES* – Spain, *AR* – Argentine, *PT* – Portugal, *BR* – Brazil). These are referenced to, by the organizers, as *variety-specific* labels. The remaining three tags, i.e. *EN*, *ES* and *PT*, are considered *common* labels. Based on this terminology, the task features two subtasks:

- *Track 1* - a nine-way classification, where both the variety-specific (e.g. *EN-GB* or *EN-US*) as well as the common label (e.g. *EN*) are considered for scoring.
- *Track 2* - evaluates a six-way classification setup, considering only the variety-specific labels.

The DSL-TL task is presented in both the open and closed formats for each of the two aforementioned tracks. Three submissions are allowed for each pair $(subtask, format)$, which amounts to a total of maximum 12 different sets of predictions that can be submitted by each team.

Our team chose the closed format and participated in both tracks, with three submissions for each subtask. All the models submitted are powered by ensemble learning. For *Track 1*, the meta-learning is based on Extreme Gradient Boosting

[1] https://github.com/LanguageTechnologyLab/DSL-TL

Tenth Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial 2023), pages 230–240

(XGBoost) (Chen and Guestrin, 2016), while for *Track 2* we employ Multinomial Logistic Regression (Peng et al., 2002) as our meta-classifier. The same subset of individual learners is used for each set of ensembles submitted , independent of its meta-learner (i.e. XGBoost or Logistic Regression) and destination (i.e. subtask).

With each submission, we gradually increase the complexity of the models plugged into the aforementioned ensembles. We start by combining the powers of two shallow methods, namely Support Vector Machines (SVM) (Cortes and Vapnik, 1995) and Kernel Ridge Regression (KRR) (Hoerl and Kennard, 1970), both using string kernels - a feature extraction technique that proved useful in previous endeavours of identifying language varieties (Ionescu and Popescu, 2016). For our second submission, we augment the ensemble of shallow models with a Character-level Convolutional Neural Network (Char-CNN) (Zhang et al., 2015), which adds depth to the ensemble and a new way of regarding the data (i.e. at the character level). The third ensemble submitted for each track contains the two string kernel based shallow methods, the Char-CNN and also a fine-tuned BERT (Devlin et al., 2019) as individual learners.

We fine-tuned and evaluated all of the individual models and meta-learners previously mentioned using the development set provided by Zampieri et al. (2023). Our final submissions include the development subset in the training routine. Moreover, our preference for only submitting ensemble models reflect the best results obtained locally with models trained on the training split and tested on the development data.

The rest of the present paper is structured as follows. In Section 2 we present related work in the space of language varieties identification. We describe in detail our approach for the DSL-TL task in Section 3. The experiments conducted and the empirical results obtained are discussed across Section 4. A set of conclusions will be drawn in Section 5.

2 Related Work

Usually modeled as a text classification problem and tackled using supervised learning approaches (Jauhiainen et al., 2019b), Language Identification (LI) research dates from the mid-60's (Mustonen, 1965), with periodic updates until the early 2000s (Takcı and Soğukpınar, 2004; Sibun, 1996; Grefenstette, 1995). Initially focused on dissimilar languages, LI has reached a peak when McNamee (2005) achieved a nearly-perfect outcome using character n-grams based models to discriminate among different languages in samples collected online.

In the last decade, language identification research has regained momentum, with social media becoming a rich and resourceful source of data. User-generated content (Tromp and Pechenizkiy, 2011) and free-form short texts (Anand, 2014) can be counted among the reasons why the research in the area of language identification was resumed. New challenges have arisen - e.g. mixing two or more different languages in social media content (Molina et al., 2016). Moreover, the idea of discriminating among similar languages or language varieties started gathering an entire community around it, especially in the VarDial evaluation campaign (Aepli et al., 2022; Chakravarthi et al., 2021; Gaman et al., 2020).

The problem of discriminating among similar languages has been tackled, to date, using a variety of ML-powered techniques practicing both shallow (Ljubešic and Kranjcic, 2014), as well as deep-learning (Li et al., 2018) with an accuracy surpassing a 95% threshold.

For language varieties on the other hand, we can observe fluctuations in performance as shown in the VarDial reports to date (Aepli et al., 2022; Chakravarthi et al., 2021; Gaman et al., 2020). For instance, Goutte et al. (2014) applies a common approach to three different language varieties: European vs Brazilian Portuguese, Castilian vs Argentine Spanish and British vs American English. The same model achieves an accuracy above 90% for the first 2 varieties and just below 53% for the third. In the Arabic dialect identification task (Malmasi et al., 2016), the highest ranking systems were based either on ensemble learning or on single SVMs trained on character and word-level n-grams (Malmasi and Zampieri, 2016; Eldesouki et al., 2016) and achieved accuracies of around 50%. Recent shared tasks (Aepli et al., 2022; Chakravarthi et al., 2021; Gaman et al., 2020; Zampieri et al., 2019, 2018, 2017) continued the work in the space of language varieties, with multiple different languages targeted over the years. Among these, we count German (Malmasi and Zampieri, 2017b), Chinese (Jauhiainen et al., 2019a) and Italian Jauhiainen et al. (2022) dialects, Dutch vs Flemish (Çöl-

tekin and Rama, 2017), Romanian vs Moldavian (Çöltekin, 2020), etc. Performance was consistent with the results in earlier campaigns (2014 - 2016) – the highest ranked results varied greatly from task to task, with n-gram based shallow models often outperforming other approaches. These works show that language identification is not a resolved problem, as we still see a struggle in performance in automatically identifying certain dialects and language varieties.

Among the most recent works on language identification, we should mention the one on which the current DSL-TL shared task is based. Zampieri et al. (2023) introduce DSL-TL as the first human-annotated multilingual data set for language variety classification. DSL-TL uses instances from DSLCC (Tan et al., 2014) - an extensive collection of samples for LI, introduced and enhanced in prior VarDial evaluation campaigns (Zampieri et al., 2017; Malmasi et al., 2016; Zampieri et al., 2014). DSL-TL also uses news reports from Zellers et al. (2019). The authors label the data from multiple human sources using a crowdsourcing platform. Moreover, alongside the qualitative data set, the authors train multiple models on these samples. The approaches used count Naive Bayes, Adaptive Naive Bayes and deep learning based methods such as mBERT, XLM-R, and XLM-R-LD and are employed as baselines in the shared task referred in the present paper. Intriguing perhaps, the authors observe similar performance across the shallow and deep learning based methods. Additionally, in some cases, the shallow methods even surpass the deep ones - an observation consistent with prior findings (Jauhiainen et al., 2019b; Medvedeva et al., 2017).

Analyzing the baselines introduced by Zampieri et al. (2023), we consider appropriate to tackle the classification problem posed by the DSL-TL task from both angles. Thus, as previously mentioned, we are combining shallow and deep learning techniques in our ensemble-powered solutions. Our choice is encouraged by prior research in the space of LI, which shows good results obtained by stacking ensembles (Malmasi and Zampieri, 2017b,a). Moreover, we choose most of the individual learners used based on their prior impact in language identification tasks: SVM with string kernels (Kruengkrai et al., 2005), CNNs (Jaech et al., 2016) and BERT (Zaharia et al., 2020). From our perspective, prior success in LI is an indication that these methods have a high chance of being suitable for the DSL-TL use-case as well. Additionally, each of the two choices of meta-learners were also used before in language variety identification: Logistic Regression (Porta and Sancho, 2014; Chen and Maison, 2003) and XGBoost (Barbaresi, 2016).

3 Methods

Our team submitted three distinct ensemble-based systems for each of the two tracks of the DSL-TL task. The choice of architecture for the meta-learner represents the one difference between the ensembles submitted for each track. For the first subtask, we use an XGBoost-based meta-learner, whereas for the second one, we rely on Logistic Regression. As mentioned in both Section 1 and Section 2, we gradually increase the complexity of the ensemble used in each submission. Figure 1 displays the prediction pipeline of the third and most complex system submitted, which is similar with a system that we used in a previous VarDial geo-location challenge (Gaman et al., 2021). From left to right, also in Figure 1, we can infer how the other pipelines are composed: the first system submitted only uses two shallow models (i.e. SVM and KRR) and the second submission adds a char-level CNN to the first system. In the continuation of this section, we briefly describe each individual machine learning technique used in the ensembles submitted, as well as the meta-learners.

3.1 Shallow Learning based on String Kernels

String Kernels. Introduced by Lodhi et al. (2001), string kernels represent an effective method (Cozma et al., 2018; Ionescu and Butnaru, 2018; Giménez-Pérez et al., 2017; Ionescu et al., 2014) of comparing two textual samples. String kernels use the inner product generated by all the character n-grams in a given document. We observe good performance of string kernel-based systems in dialect identification, with emphasis on previous VarDial editions (Butnaru and Ionescu, 2018; Ionescu and Popescu, 2016).

Using the technique introduced by Popescu et al. (2017), we obtain a kernel matrix X where the element X_{ij} measures the similarity between two documents x_i and x_j. The similarity function used is the presence bits string kernel (Popescu and Ionescu, 2013), which is defined as follows:

$$k^{0/1}(x_i, x_j) = \sum_{g \in S^n} \#(x_i, g) \cdot \#(x_j, g), \quad (1)$$

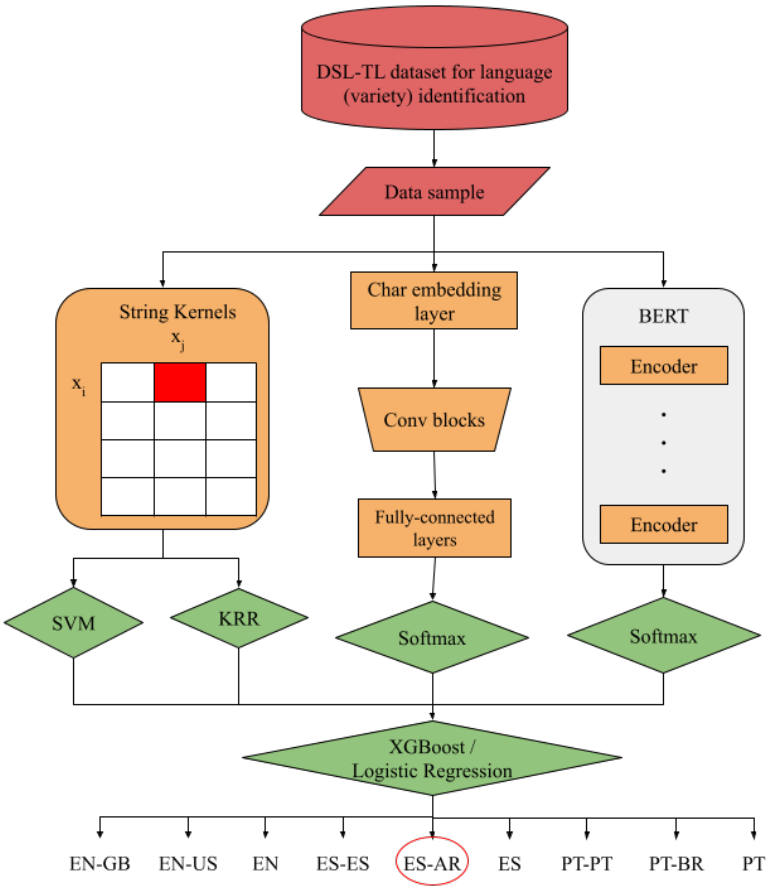

Figure 1: Full ensemble (submission 3) proposed by UnibucNLP for the DSL-TL shared task. Best viewed in color.

where S is a set of characters; x_i and x_j are the strings to be compared; n is the length of the char n-grams used and $\#(x, g)$ is a function with binary outcome that returns 1 when n-gram g occurs at least once in x.

Support Vector Machines – SVM(s). The goal in SVMs (Cortes and Vapnik, 1995) is to find the best hyperplane that separates the training data points in their respective classes. At the same time, in order to achieve better generalization, SVM tries to maximize the margin that separates the two classes, using *support vectors* (i.e. the points closest to the decision boundary). An advantage of SVM is the kernel trick (Shawe-Taylor and Cristianini, 2004) - a technique used to map the non-linearly separable data in a higher-dimensional space, where it becomes separable through a hyperplane. Although designed with 2-way classification in mind, SVMs can be used in the multi-class setup through the training of multiple models in a one-vs-one or one-vs-rest scheme. In our current experiments, we use the one-vs-one technique. Moreover, instead of using a standard kernel, we employ the SVMs with the custom n-gram based string kernel defined in Equation 1.

Kernel Ridge Regression (KRR). Considered a generalization of Ridge Regression (Hoerl and Kennard, 1970), KRR is obtained by combining L2 linear regression with the kernel trick (Saunders et al., 1998). Thus, KRR presents the same two big advantages as is the case with SVM - (1) it can model non-linearly separable data and (2) we can use a custom kernel function. For the DSL-TL task, we employ the presence bits kernel from Equation 1. We also follow two steps to repurpose the trained regressor for multi-class classification: (1) we round the continuous predictions to match the values in $\{-1, 1\}$ and (2) we use the one-versus-rest scheme.

3.2 Deep Learning

Character-level Convolutional Neural Network (Char-CNN). Regarded as the base unit in any given vocabulary, characters represent a popular (Al-Rfou et al., 2019; Kim et al., 2016; Zhang et al., 2015; Sutskever et al., 2011) non-pretentious source of features for text-based ML models. When working at character level, we remove dependen-

cies of syntax and semantic structure (Ballesteros et al., 2015). Given that in DSL-TL we have multiple languages mixed in the same data set, the aforementioned property represents a welcomed advantage for the present use-case.

CNNs are a type of neural network that joins convolutions and pooling operations in convolutional blocks. Towards the end of the network we usually add a sequence of fully connected layers, followed by a terminal prediction layer. In this work, we employ a convolutional neural network operating at char level (Zhang et al., 2015) with squeeze-and-excitation (SE) blocks, introduced and successfully used in dialect identification by Butnaru and Ionescu (2019).

Transformers (BERT). With an encoder-decoder based architecture, transformers (Vaswani et al., 2017) are among the most important advancements in NLP in the past decade. Widely used since its release, BERT (Devlin et al., 2019) is a special type of transformer, which pre-trains deep bidirectional representations of language in a self-supervised fashion. For downstream tasks, such as our current language varieties identification problem, it is straightforward to fine-tune a pretrained BERT model. BERT is our last choice of individual learner given the good results obtained in similar dialect / language variety identification setups (Zaharia et al., 2020).

3.3 Ensemble Learning

XGBoost. XGBoost is a tree-based ensemble model (Chen and Guestrin, 2016; Friedman, 2001), effectively employed in both academic research (Li, 2010; Burges, 2010; Bennett et al., 2007) as well as the industry (He et al., 2014). In our experiments, XGBoost is the chosen meta-learner for *Track 1*. We train XGBoost over the predictions of each individual models previously described in the current section.

Logistic Regression (LR). Multinomial Logistic Regression is a generalization of LR (Peng et al., 2002) to multi-class classification problems. Logistic Regression has been historically employed in language identification tasks (Porta and Sancho, 2014; Chen and Maison, 2003). Moreover, in our experiments, the ensembles that used multinomial Logistic Regression as meta-learner achieved similar performance when compared to the XGBoost meta-learner. Thus, we decided to also submit the predictions of the set of ensembles based on LR. We should mention that we trained the LR-based ensembles on all of the tags available, including the common labels (i.e. GB, ES and PT). No language-variety specifics were enforced for this ensemble whose predictions were submitted for *Track 2*.

4 Experiments

4.1 Data Set

The DSL-TL data set (Zampieri et al., 2023) is targeted towards the task of discriminating between language varieties. Consistent with its purpose, the data set contains a total of $12,900$ instances written in either English (EN), Spanish (ES) or Portuguese (PT) and manually labelled from multiple sources. DSL-TL makes a distinction among two different varieties for each of the three languages included. Thus, we observe the following six composed labels in the data set: *EN-GB* - British English, *EN-US* - American English, *ES-ES* - Castilian Spanish, *ES-AR* - Argentine Spanish, *PT-PT* - European Portuguese and *PT-BR* - Brazilian Portuguese. Moreover, we also have 3 common labels, namely *EN*, *ES* and *PT*, for the samples not containing any variety specific markers.

DSL-TL provides three splits for training, development and the final testing of the solutions proposed to address the task. The split was performed following the $70/20/10$ rule. The training and development textual samples are provided alongside their respective language labels. The test set only contains the textual samples, pending further submission of predictions such that the organizers can evaluate them against the ground truth.

4.2 Hyperparameter Tuning

SVM. In our experiments, we use SVM with a pre-computed string kernel and the regularization parameter $C = 10$. We select the best regularization value via grid search from a range of values from 10^{-4} to 10^{4}, with a multiplication step of 10. For the string kernel used, we experiment with multiple presence-bits string kernels based on various n-gram lengths, from 3 to 6 characters long. The best performance in terms of accuracy and macro $F1 - score$ was achieved by a string kernel based on the blended spectrum of 3 to 5 character n-grams.

KRR. For KRR, we tune the regularization λ using a set of values that range from 10^{-6} to 10^{-1}, and a multiplication step of 10. The best value for λ

in our 9-way classfication setup was 10^{-2}. Similar with the SVMs, the string kernel used in KRR is based on a blended spectrum of 3 to 5 character n-grams.

CharCNN. The third individual learner used is a character-level CNN (Zhang et al., 2015), operating on an input window of maximum 256 characters in each sample, as indicated by a closer inspection of the data. The architecture used is very similar with the one employed by Butnaru and Ionescu (2019) in Romanian dialect identification. Each of the maximum 256 characters considered in the input layer is embedded into a vector of size 128, selected from a set of powers of 2 as potential embedding sizes, ranging from 16 up until 256. Three convolutional blocks follow, each having a convolutional layer with 128 filters, a stride of 1 and filter sizes 7, 5 and 3. We use max pooling with a filter of size 3 to downsample the output of the convolutional layer. Each convolutional block is followed by a Squeeze-and-Excitation (SE) block with a reduction ratio $r = 64$. The sequence of convolutional blocks is followed by one fully connected layer with 128 neural units, out of which we drop neurons with a probability of 0.5. The neural network is also equipped with a final Softmax-activated prediction layer, of size 9 to retrieve a probability for each of the classes in DSL-TL. We use a learning rate of 10^{-4} and train the network for 100 epochs on mini-batches of 128 samples. Early stopping is used with a tolerance of 10 consecutive epochs for stalled performance.

Fine-tuned BERT. Our fourth and last individual learner consists in a fine-tuned multilingual BERT model (Devlin et al., 2019). Prior to fine tuning the model, we use the multilingual BERT tokenizer to encode each example into a list of token IDs. Then, each token is translated into a 768-dimensional embedding vector. Furthermore, the architecture is augmented with a global average pooling layer to achieve a Continuous Bag-of-Words (CBOW) representation of the data. In the end, a Softmax output layer predicts the likeliness of a sample being marked with each of the nine language tags provided. We fine-tune the model described above for 30 epochs with early stopping. We train on mini-batches of 32 samples and optimize using Adam with decoupled weight decay (AdamW) (Loshchilov and Hutter, 2019), a learning rate of $5 \cdot 10^{-5}$ and an ϵ equal to 10^{-8}. We tuned the learning rate using a few different values in the range of 10^{-5} and 10^{-4} and tested two loss options, cross-entropy vs. negative log-likelihood. In the end, we opted for the cross-entropy loss.

XGBoost. We fine-tune the XGBoost meta-learner separately, for each of the three submissions. The set of values considered for the maximum depth of a tree is $[3, 5, 7, 9, 10]$. We fine-tuned the learning rate in a range starting from 10^{-4} up to 10^{-1}, with a multiplying step of 10. The subsample ratio of columns when constructing each tree was picked from $[0.1, 0.3, 0.5, 0.7]$. The number of estimators is gradually initialized with values ranging from 50 and up to 400 with an additive step of 50. For each submission, a different set of parameters was deemed optimal. Thus, for the ensemble composed of shallow models, the best parameters were: max_depth=5, learning_rate=10^{-1}, n_estimators=50 and colsample_bytree=0.5. When adding the character-level CNN into the mix of shallow models, the best choice of hyper parameters changes slightly: max_depth and learning_rate remain the same as previously mentioned; however, in this case, n_estimators=100 and colsample_bytree=0.7. With BERT included in the ensemble of shallow and deep models, all the optimal parameters change as follows: max_depth=7, learning_rate=10^{-3}, n_estimators=200 and colsample_bytree=0.5.

Logistic Regression. In the case of the Logistic Regression based meta-learner, we use L2 regularization and only fine tune the inverse of the regularization strength parameter, noted as C. The range of values tested starts with 10^{-5} and ends with 10^{5}. Different optimal values are observed for each run, as we gradually increase the number of learners and their respective depths. For the ensemble of shallow methods, we observe that a C=10^{3} gives the best scores both in terms of accuracy, as well as for the $macroF1-score$. The optimal value for C decreases to 10^{2} when we combine the Char-CNN with the two shallow models. We observe a further decrease in the best value for C, i.e. 10^{1}, when we add the BERT model to the second ensemble.

4.3 Results

Track 1 For *Track 1* we submitted 3 XGBoost stacking ensembles, gradually adding more complex individual learners to the ensemble as follows. For the first run, we combine only the powers of two shallow models, namely SVM and KRR. In the second run, we add a character-level CNN to

the ensemble of shallow models. Finally, in the third run, we add a fine tuned BERT model to the second run. In our local testing, the performance on the development set increased with the addition of each individual learner. Thus, we deemed our first run, *UnibucNLP-run-1*, as being the weakest of the three submissions for this track, followed by the second run, *UnibucNLP-run-2* and with *UnibucNLP-run-3* being the top performing system that we have submitted.

Method	Rank	F1-score
VaidyaKane-run-3	1	0.5854
baseline-mBERT	4	0.54
baseline-XLM-R	5	0.536
UnibucNLP-run-3	6	0.5318
baseline-XLM-R-LD	7	0.529
baseline-NB	8	0.503
UnibucNLP-run-1	11	0.4875
UnibucNLP-run-2	13	0.4572

Table 1: The final results for the closed format of *Track 1* obtained by our XGBoost based ensembles on the DSL-TL test set. For simplicity, we compare ourselves only against the baseline and the top scoring method. In bold are the methods that we submitted and described in the current work.

Tag	Method	Rank	F1-score
EN	VaidyaKane-run-3	1	0.3333
EN	**UnibucNLP-run-3**	2	0.32
EN	baseline-mBERT	3	0.303
EN-GB	VaidyaKane-run-1	1	0.8148
EN-GB	**UnibucNLP-run-3**	4	0.8034
EN-GB	baseline-XLM-R	5	0.793
EN-US	**UnibucNLP-run-3**	1	0.8454
EN-US	baseline-mBERT	3	0.829
ES	VaidyaKane-run-2	1	0.4738
ES	**UnibucNLP-run-3**	2	0.4573
ES	baseline-mBERT	3	0.455
ES-AR	VaidyaKane-run-1	1	0.6204
ES-AR	baseline-mBERT	4	0.518
ES-AR	**UnibucNLP-run-3**	9	0.4884
ES-ES	VaidyaKane-run-1	1	0.7692
ES-ES	baseline-XLM-R	3	0.719
ES-ES	**UnibucNLP-run-1**	7	0.6858
PT	VaidyaKane-run-2	1	0.1633
PT	baseline-NB	4	0.126
PT	**UnibucNLP-run-3**	7	0.1165
PT-PT	ssl-run-1	1	0.7923
PT-PT	baseline-XLM-R	5	0.769
PT-PT	**UnibucNLP-run-3**	7	0.7618
PT-BR	baseline-XLM-R	1	0.562
PT-BR	**UnibucNLP-run-1**	12	0.4683
PT-BR	**UnibucNLP-run-2**	13	0.378
PT-BR	**UnibucNLP-run-3**	14	0.3575

Table 2: The performance per class reported on the test set for the closed format of *Track 1* obtained by our best performing ensemble compared to the baseline and the top scoring method. We mark in bold our own work.

Table 1 partially confirms our intuition, as our third run is indeed out-performing the other two ensemble-based systems. Surprisingly perhaps, the ensemble that combines the predictions of the Char-CNN and the ones of SVM and KRR falls behind the model that employs only the shallow individual models. Our best performing submission is situated just below two of the best performing baselines provided for Track 1, and immediately above the worst-performing baselines in this subtask. The 9-way classification proved to be a difficult problem, as most of the submissions are below the worst performing baseline provided by the organizers. Three submissions of the same team (i.e. *VaidyaKane*) are above all of the baselines, then our best performing system is right in the middle, ranking sixth if we consider the baselines and fourth if we don't, then, below the baselines we can see the scores of all the other systems submitted (including ours - run 1 and run 2).

Table 2 shows the ranking and score of our best performing method for each of the 9 classes considered. We achieve a good position in classifying the samples that are written in English - ranking first for *EN-US*, second for the common label *EN* and fourth for *EN-GB*. Although for the common Spanish tag we rank second, for the Castilian and Argentine language varieties, we only achieve the seventh and ninth positions respectively. The common label for Portuguese seems to bring ourselves and everyone other participant down, with the best model not being able to obtain an $F1 - score$ greater than 0.1633. The results for European Portuguese are better, and with values very close to each other across all of the systems submitted. In these conditions, for *PT-PT* we achieve an $F1 - score$ of 0.7618. In the end, as shown in the final rows of Table 2, all of our systems achieve the worst results for Brazilian Portuguese.

Track 2 *Track 2* tests a six-way classification, using only the variety-specific tags and ignoring the common labels. For this subtask, we submit three stacking ensembles, following the same logic as for the submissions in *Track 1*, the only difference being that we use Logistic Regression as meta-learner. We do not perform any variety-specific transformations and we do not exclude the common

labels at training for the three runs submitted for *Track 2*. Thus, our expectations are consistent with the results obtained and displayed in Table 3.

Method	Rank	F1-score
VaidyaKane-run-1	1	0.8561
baseline-ANB	4	0.799
baseline-NB	5	0.794
baseline-XLM-R	6	0.78
baseline-XLM-R-LD	7	0.772
baseline-mBERT	9	0.755
UnibucNLP-run-1	13	0.6935
UnibucNLP-run-3	14	0.6855
UnibucNLP-run-2	15	0.6182

Table 3: The final results for the closed format of *Track 2* obtained by our Logistic Regression based ensembles on the DSL-TL test set. For simplicity, we compare ourselves only against the baselines and the top scoring method. In bold are the methods that we submitted and described in the current work.

One interesting fact observed in Table 3 is that our first run - an ensemble of string kernel based shallow models, outperforms our other two runs, based on more complex models such as the Char-CNN and BERT models.

5 Conclusions

In this work we propose six ensemble models to address the problem of language-variety identification in news reports. To tackle the two tracks proposed by the DSL-TL task, we employ two similar sets of ensembles which differ only in the choice of meta-learner: XGBoost for the 9-way classification in the first track, and Logistic Regression for the 6-way classification in the second one. By the definition of *Track 2*, our Logistic Regression based systems are evaluated only on the variety-specific labels provided. However, we have trained these LR powered ensembles also on the common labels, in hopes that the model will learn additional useful representations. For each set of ensembles submitted, we follow a similar strategy: increase the number of models and individual models' complexity for each run. Thus, our first submission only combines predictions from KRR and SVM - two shallow models. In the second ensemble we add a CNN working at character level, and in the third one, we augment the second ensemble with a fine tuned multilingual BERT model.

For the 9-way classification, our best performing model achieves a macro F1-score of 53.18%, 5% less than the top scoring submission. Overall, our model ranks fourth out of 9 total submissions and surpasses two of the four strong baselines proposed by the organizers. In the variety-specific, 6-way classification of Track 2, most of the models submitted by participants (including ours) fall behind the proposed baselines. Interestingly, our best performing submission in this case is the ensemble of shallow models, which obtains a score of 69.35%, surpassing the other 2, more complex ensembles, that we submitted.

Given the final results, we conclude that in future similar endeavours we should not underestimate the power of shallow models, as they consistently seem to achieve good results in language identification setups. Moreover, we intend on performing a closer analysis of the baselines proposed in Zampieri et al. (2023) - the paper that introduces DSL-TL, try to replicate and perhaps enhance the already impressive methods that the authors used for this task.

Limitations

Limitations of the present work and results include tackling the closed format of the DSL-TL task. As shown in Zampieri et al. (2023) using additional data, from the broader DSLCC corpus (Tan et al., 2014), would have likely helped both the 9-way as well as the 6-way classification attempted in our submissions.

Hardware limitations represent another disadvantage, due to which a better, broader fine-tuning of the deep learning based models could not be fully achieved in time.

Acknowledgements

The authors would like to kindly thank reviewers for their suggestions, which were deemed to be very helpful in improving the present writing.

References

Noëmi Aepli, Antonios Anastasopoulos, Adrian-Gabriel Chifu, William Domingues, Fahim Faisal, Mihaela Gaman, Radu Tudor Ionescu, and Yves Scherrer. 2022. Findings of the VarDial evaluation campaign 2022. In *Proceedings of VarDial*, pages 1–13.

Noëmi Aepli, Çağrı Çöltekin, Rob van der Goot, Tommi Jauhiainen, Mourhaf Kazzaz, Nikola Ljubešić, Kai North, Barbara Plank, Yves Scherrer, and Marcos Zampieri. 2023. Findings of the VarDial evaluation campaign 2023. In *Proceedings of VarDial*.

Rami Al-Rfou, Dokook Choe, Noah Constant, Mandy Guo, and Llion Jones. 2019. Character-Level Language Modeling with Deeper Self-Attention. In *Proceedings of AAAI*, pages 3159–3166.

Supriya Anand. 2014. Language identification for transliterated forms of indian language queries. In *Proceedings of FIRE*.

Miguel Ballesteros, Chris Dyer, and Noah A. Smith. 2015. Improved Transition-Based Parsing by Modeling Characters instead of Words with LSTMs. In *Proceedings of EMNLP*, pages 349–59.

Adrien Barbaresi. 2016. An unsupervised morphological criterion for discriminating similar languages. In *Proceedings of VarDial*, pages 212–220.

James Bennett, Stan Lanning, et al. 2007. The Netflix Prize. In *Proceedings of KDD*, volume 2007, page 35.

Christopher J.C. Burges. 2010. From RankNet to LambdaRank to LambdaMART: An Overview. *Learning*, 11(23-581):81.

Andrei Butnaru and Radu Tudor Ionescu. 2018. UnibucKernel Reloaded: First Place in Arabic Dialect Identification for the Second Year in a Row. In *Proceedings of VarDial*, pages 77–87.

Andrei M. Butnaru and Radu Tudor Ionescu. 2019. MOROCO: The Moldavian and Romanian Dialectal Corpus. In *Proceedings of ACL*, pages 688–698.

Bharathi Raja Chakravarthi, Gaman Mihaela, Radu Tudor Ionescu, Heidi Jauhiainen, Tommi Jauhiainen, Krister Lindén, Nikola Ljubešić, Niko Partanen, Ruba Priyadharshini, Christoph Purschke, Eswari Rajagopal, Yves Scherrer, and Marcos Zampieri. 2021. Findings of the VarDial evaluation campaign 2021. In *Proceedings of VarDial*, pages 1–11.

Stanley Chen and Benoît Maison. 2003. Using place name data to train language identification models.

Tianqi Chen and Carlos Guestrin. 2016. Xgboost: A scalable tree boosting system. In *Proceedings of SIGKDD*, page 785–794.

Çağrı Çöltekin. 2020. Dialect identification under domain shift: Experiments with discriminating Romanian and Moldavian. In *Proceedings of VarDial*, pages 186–192.

Çağrı Çöltekin and Taraka Rama. 2017. Tübingen system in VarDial 2017 shared task: experiments with language identification and cross-lingual parsing. In *Proceedings of VarDial*, pages 146–155.

Corinna Cortes and Vladimir Vapnik. 1995. Support-vector networks. *Machine Learning*, 20(3):273–297.

Mădălina Cozma, Andrei Butnaru, and Radu Tudor Ionescu. 2018. Automated essay scoring with string kernels and word embeddings. In *Proceedings of ACL*, pages 503–509.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of NAACL*, pages 4171–4186.

Mohamed Eldesouki, Fahim Dalvi, Hassan Sajjad, and Kareem Darwish. 2016. QCRI @ DSL 2016: Spoken Arabic dialect identification using textual features. In *Proceedings of VarDial*, pages 221–226.

Jerome H Friedman. 2001. Greedy function approximation: a gradient boosting machine. *Annals of statistics*, pages 1189–1232.

Mihaela Gaman, Sebastian Cojocariu, and Radu Tudor Ionescu. 2021. UnibucKernel: Geolocating Swiss German jodels using ensemble learning. In *Proceedings of VarDial*, pages 84–95.

Mihaela Gaman, Dirk Hovy, Radu Tudor Ionescu, Heidi Jauhiainen, Tommi Jauhiainen, Krister Lindén, Nikola Ljubešić, Niko Partanen, Christoph Purschke, Yves Scherrer, and Marcos Zampieri. 2020. A report on the VarDial evaluation campaign 2020. In *Proceedings of VarDial*, pages 1–14.

Rosa M. Giménez-Pérez, Marc Franco-Salvador, and Paolo Rosso. 2017. Single and Cross-domain Polarity Classification using String Kernels. In *Proceedings of EACL*, pages 558–563.

Cyril Goutte, Serge Léger, and Marine Carpuat. 2014. The NRC system for discriminating similar languages. In *Proceedings of VarDial*, pages 139–145.

Gregory Grefenstette. 1995. Comparing two language identification schemes. In *Proceedings of JADT*, volume 95.

Xinran He, Junfeng Pan, Ou Jin, Tianbing Xu, Bo Liu, Tao Xu, Yanxin Shi, Antoine Atallah, Ralf Herbrich, Stuart Bowers, et al. 2014. Practical lessons from predicting clicks on ads at facebook. In *Proceedings of ADKDD*, pages 1–9.

Arthur E. Hoerl and Robert W. Kennard. 1970. Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 12(1):55–67.

Radu Tudor Ionescu and Andrei Butnaru. 2018. Improving the results of string kernels in sentiment analysis and Arabic dialect identification by adapting them to your test set. In *Proceedings of EMNLP*, pages 1084–1090.

Radu Tudor Ionescu and Marius Popescu. 2016. UnibucKernel: An Approach for Arabic Dialect Identification based on Multiple String Kernels. In *Proceedings of VarDial*, pages 135–144.

Radu Tudor Ionescu, Marius Popescu, and Aoife Cahill. 2014. Can characters reveal your native language? A language-independent approach to native language identification. In *Proceedings of EMNLP*, pages 1363–1373.

Aaron Jaech, George Mulcaire, Shobhit Hathi, Mari Ostendorf, and Noah Smith. 2016. Hierarchical character-word models for language identification. In *Proceedings of SocialNLP*, pages 84–93.

Tommi Jauhiainen, Heidi Jauhiainen, and Krister Lindén. 2022. Italian language and dialect identification and regional French variety detection using adaptive naive Bayes. In *Proceedings of VarDial*, pages 119–129.

Tommi Jauhiainen, Krister Lindén, and Heidi Jauhiainen. 2019a. Discriminating between Mandarin Chinese and Swiss-German varieties using adaptive language models. In *Proceedings of VarDial*, pages 178–187.

Tommi Jauhiainen, Marco Lui, Marcos Zampieri, Timothy Baldwin, and Krister Lindén. 2019b. Automatic language identification in texts: A survey. 65(1):675–682.

Yoon Kim, Yacine Jernite, David Sontag, and Alexander M. Rush. 2016. Character-Aware Neural Language Models. In *Proceedings of AAAI*, pages 2741–2749.

C. Kruengkrai, P. Srichaivattana, V. Sornlertlamvanich, and H. Isahara. 2005. Language identification based on string kernels. In *Proceedings of IEEE*, volume 2, pages 926–929.

Ping Li. 2010. Robust Logitboost and Adaptive Base Class (ABC) Logitboost. In *Proceedings of UAI*, pages 302–311.

Yitong Li, Timothy Baldwin, and Trevor Cohn. 2018. What's in a domain? learning domain-robust text representations using adversarial training. In *Proceedings of NAACL*, pages 474–479.

Nikola Ljubešic and Denis Kranjcic. 2014. Discriminating between very similar languages among twitter users. In *Proceedings of LTC*, pages 90–94.

Huma Lodhi, John Shawe-Taylor, Nello Cristianini, and Christopher J.C.H. Watkins. 2001. Text Classification Using String Kernels. In *Proceedings of NIPS*, pages 563–569.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled Weight Decay Regularization. In *Proceedings of ICLR*.

Shervin Malmasi and Marcos Zampieri. 2016. Arabic dialect identification in speech transcripts. In *Proceedings of VarDial*, pages 106–113.

Shervin Malmasi and Marcos Zampieri. 2017a. Arabic dialect identification using iVectors and ASR transcripts. In *Proceedings of VarDial*, pages 178–183.

Shervin Malmasi and Marcos Zampieri. 2017b. German dialect identification in interview transcriptions. In *Proceedings of VarDial*, pages 164–169.

Shervin Malmasi, Marcos Zampieri, Nikola Ljubešić, Preslav Nakov, Ahmed Ali, and Jörg Tiedemann. 2016. Discriminating between similar languages and Arabic dialect identification: A report on the third DSL shared task. In *Proceedings of VarDial*, pages 1–14.

Paul McNamee. 2005. Language identification: A solved problem suitable for undergraduate instruction. *J. Comput. Sci. Coll.*, 20(3):94–101.

Maria Medvedeva, Martin Kroon, and Barbara Plank. 2017. When sparse traditional models outperform dense neural networks: the curious case of discriminating between similar languages. In *Proceedings of VarDial*, pages 156–163.

Giovanni Molina, Fahad AlGhamdi, Mahmoud Ghoneim, Abdelati Hawwari, Nicolas Rey-Villamizar, Mona Diab, and Thamar Solorio. 2016. Overview for the second shared task on language identification in code-switched data. In *Proceedings of CodeSwitch*, pages 40–49.

Seppo Mustonen. 1965. Multiple discriminant analysis in linguistic problems. *Statistical Methods in Linguistics*, 4:37–44.

Joanne Peng, Kuk Lee, and Gary Ingersoll. 2002. An introduction to logistic regression analysis and reporting. *Journal of Educational Research - J EDUC RES*, 96:3–14.

Marius Popescu, Cristian Grozea, and Radu Tudor Ionescu. 2017. HASKER: An efficient algorithm for string kernels. Application to polarity classification in various languages. In *Proceedings of KES*, pages 1755–1763.

Marius Popescu and Radu Tudor Ionescu. 2013. The Story of the Characters, the DNA and the Native Language. In *Proceedings of BEA-8*, pages 270–278.

Jordi Porta and José-Luis Sancho. 2014. Using maximum entropy models to discriminate between similar languages and varieties. In *Proceedings of VarDial*, pages 120–128.

Craig Saunders, Alexander Gammerman, and Volodya Vovk. 1998. Ridge Regression Learning Algorithm in Dual Variables. In *Proceedings of ICML*, pages 512–521.

John Shawe-Taylor and Nello Cristianini. 2004. *Kernel Methods for Pattern Analysis*. Cambridge University Press.

Penelope Sibun. 1996. Language identification: Examining the issues. In *Proceedings of SDAIR*.

Ilya Sutskever, James Martens, and Geoffrey Hinton. 2011. Generating Text with Recurrent Neural Networks. In *Proceedings of ICML*, pages 1017–1024.

Hidayet Takcı and İbrahim Soğukpınar. 2004. Centroid-based language identification using letter feature set. In *Proceedings of CICLing*, pages 640–648. Springer.

Liling Tan, Marcos Zampieri, Nikola Ljubešic, and Jörg Tiedemann. 2014. Merging comparable data sources for the discrimination of similar languages: The dsl corpus collection. In *Proceedings of the 7th Workshop on Building and Using Comparable Corpora (BUCC)*, pages 11–15.

Erik Tromp and Mykola Pechenizkiy. 2011. Graph-based n-gram language identification on short texts. In *Proceedings of BeNeLearn*, pages 27–34.

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 NIPS*, pages 5998–6008.

George-Eduard Zaharia, Andrei-Marius Avram, Dumitru-Clementin Cercel, and Traian Rebedea. 2020. Exploring the power of Romanian BERT for dialect identification. In *Proceedings of VarDial*, pages 232–241.

Marcos Zampieri, Shervin Malmasi, Nikola Ljubešić, Preslav Nakov, Ahmed Ali, Jörg Tiedemann, Yves Scherrer, and Noëmi Aepli. 2017. Findings of the VarDial evaluation campaign 2017. In *Proceedings of VarDial*, pages 1–15.

Marcos Zampieri, Shervin Malmasi, Preslav Nakov, Ahmed Ali, Suwon Shon, James Glass, Yves Scherrer, Tanja Samardžić, Nikola Ljubešić, Jörg Tiedemann, Chris van der Lee, Stefan Grondelaers, Nelleke Oostdijk, Dirk Speelman, Antal van den Bosch, Ritesh Kumar, Bornini Lahiri, and Mayank Jain. 2018. Language identification and morphosyntactic tagging: The second VarDial evaluation campaign. In *Proceedings of VarDial*, pages 1–17.

Marcos Zampieri, Shervin Malmasi, Yves Scherrer, Tanja Samardžić, Francis Tyers, Miikka Silfverberg, Natalia Klyueva, Tung-Le Pan, Chu-Ren Huang, Radu Tudor Ionescu, Andrei M. Butnaru, and Tommi Jauhiainen. 2019. A report on the third VarDial evaluation campaign. In *Proceedings of VarDial*, pages 1–16.

Marcos Zampieri, Kai North, Tommi Jauhiainen, Mariano Felice, Neha Kumari, Nishant Nair, and Yash Bangera. 2023. Language variety identification with true labels. *arXiv preprint arXiv:2303.01490*.

Marcos Zampieri, Liling Tan, Nikola Ljubešić, and Jörg Tiedemann. 2014. A report on the DSL shared task 2014. In *Proceedings of VarDial*, pages 58–67.

Marcos Zampieri, Liling Tan, Nikola Ljubešić, Jörg Tiedemann, and Preslav Nakov. 2015. Overview of the DSL shared task 2015. In *Proceedings of VarDial*, pages 1–9.

Rowan Zellers, Ari Holtzman, Hannah Rashkin, Yonatan Bisk, Ali Farhadi, Franziska Roesner, and Yejin Choi. 2019. *Defending against Neural Fake News*.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level Convolutional Networks for Text Classification. In *Proceedings of NIPS*, pages 649–657.