Detection of Similar Languages and Dialects Using Deep Supervised Autoencoders

Shantipriya Parida[1], **Esaú Villatoro-Tello**[1,2], **Sajit Kumar**[3], **Maël Fabien**[1,4], and **Petr Motlicek**[1]

[1]Idiap Research Institute, Martigny, Switzerland.
{firstname.lastname}@idiap.ch
[2]Universidad Autónoma Metropolitana, Unidad Cuajimalpa, Mexico City, Mexico.
evillatoro@correo.cua.uam.mx
[3]Great Learning, Bangalore, India.
kumar.sajit.sk@gmail.com
[4]Ecole Polytechnique Fédérale de Lausanne, Switzerland
mael.fabien@epfl.ch

Abstract

Language detection is considered a difficult task especially for similar languages, varieties, and dialects. With the growing number of online content in different languages, the need for reliable and robust language detection tools also increased. In this work, we use supervised autoencoders with a bayesian optimizer for language detection and highlight its efficiency in detecting similar languages with dialect variance in comparison to other state-of-the-art techniques. We evaluated our approach on multiple datasets (Ling10, Discriminating between Similar Language (DSL), and Indo-Aryan Language Identification (ILI)). Obtained results demonstrate that SAE is highly effective in detecting languages, up to a 100% accuracy in the Ling10. Similarly, we obtain a competitive performance in identifying similar languages, and dialects, 92%, and 85% for DSL and ILI datasets respectively.

1 Introduction

Internet content is growing exponentially over time, and as a direct consequence, more languages and dialects need to be processed, as they serve as key components in various Natural Language Processing (NLP) tasks (Kocmi and Bojar, 2017).

Language detection is the task of determining the language for a given text. Although language detection has significantly improved over the past years, challenges remain. Detecting similar languages, detecting languages when multiple language contents exist in a single document, and detecting language in short texts are still active research areas (Balazevic et al., 2016; Lui et al., 2014; Williams and Dagli, 2017). Discriminate between very close languages or dialects, for example, German dialect identification, Indo-Aryan language identification, is considered a difficult task (Parida et al., 2020; Jauhiainen et al., 2019a). Although dialect identification is commonly based on the distributions of letters or letter n-grams, these approaches might face serious difficulties when trying to distinguish related dialects that have similar phoneme and grapheme inventories (Scherrer and Rambow, 2010). In a multilingual country like India, there exist many languages and many of them have multiple dialects (Chittaragi and Koolagudi, 2019). For example, in the case of the Odia language, although the written text is the same, there exist many dialects (e.g. Baleswari, Ganjami, Sambalpuri, Desiya. etc.) (Swain et al., 2016). Moreover, the automatic identification of dialect in low resource languages suffers from the lack of large training datasets or pre-trained language models.

Most of the previous research on language identification has focused on using traditional machine learning approaches like Naive Bayes, Support Vector Machine (SVM), in combination with word n-grams, graph-based n-grams, prediction partial matching (PPM) or linear interpolation with post-independent weight optimization and majority voting for combining multiple classifiers (Jauhiainen et al., 2019b). However, more recently, deep learning techniques have shown substantial results in many NLP tasks including language detection (Oro et al., 2018; Villatoro-Tello et al., 2020a,b). For many deep learning tasks, semi-supervised autoencoders have proven to build reliable representations with few annotated data (Ranzato and Szummer, 2008; Rasmus et al., 2015). To the best of our knowledge, autoencoders (AE) have never been applied for similar language detection. In this paper, we explore the use of supervised autoencoders (SAE), hence leveraging labels in the latent space, for language detection.

Proceedings of the 17th International Conference on Natural Language Processing, pages 362–367
Patna, India, December 18 - 21, 2020.

2 Proposed Method

The overall architecture of the proposed method is shown in Figure 1. The following subsections briefly describe the main components of our approach.

2.1 Supervised Autoencoder

An AE is a neural network that learns a low-dimensional representation (encoding) of input data and then learns to reconstruct the original input from the learned representation. This type of architecture is mainly used for dimensionality reduction or feature extraction (Zhu and Zhang, 2019), in an unsupervised fashion. By learning to reconstruct the input, the AE extracts underlying abstract attributes that facilitate accurate prediction of the input.

A supervised autoencoder (SAE) is an AE with the addition of a supervised loss on the representation layer. For the case of a single hidden layer, a supervised loss is added to the output layer and for a deeper AE, the innermost layer has a supervised loss added to the bottleneck layer that is usually transferred to the supervised layer after training the AE.

In supervised learning, the goal is to learn a function for a vector of inputs $\mathbf{x} \in \mathbb{R}^d$ to predict a vector of targets $\mathbf{y} \in \mathbb{R}^m$. Consider SAE with a single hidden layer of size k, and the weights for the first layer are $\mathbf{F} \in \mathbb{R}^{k \times d}$. The function is trained on a finite batch of independent and identically distributed (i.i.d.) data, $(\mathbf{x}_1, \mathbf{y}_1), ..., (\mathbf{x}_t, \mathbf{y}_t)$, with the goal of a more accurate prediction on new samples generated from the same distribution. The weight for the output layer consists of weights $\mathbf{W}_p \in \mathbb{R}^{m \times k}$ to predict $\mathbf{y}$ and $\mathbf{W}_r \in \mathbb{R}^{d \times k}$ to reconstruct $\mathbf{x}$. Let L_p be the supervised loss and L_r be the loss for the reconstruction error. In the case of regression, both losses might be represented by a squared error, resulting in the objective:

$$\frac{1}{t}\sum_{i=1}^{t}\Big[L_p(\mathbf{W}_p\mathbf{F}\mathbf{x}_i, \mathbf{y}_i) + L_r(\mathbf{W}_r\mathbf{F}\mathbf{x}_i, \mathbf{x}_i)\Big] = \frac{1}{2t}\sum_{i=1}^{t}\Big[||\mathbf{W}_p\mathbf{F}\mathbf{x}_i - \mathbf{y}_i||_2^2 + ||\mathbf{W}_r\mathbf{F}\mathbf{x}_i - \mathbf{x}_i||_2^2\Big] \quad (1)$$

The addition of supervised loss to the AE loss function acts as regularizer and results (as shown in equation 1) in the learning of the better representation for the desired task (Le et al., 2018). In summary, an SAE represents a neural network that jointly predicts targets and inputs.

2.2 Bayesian Optimizer

In the case of SAE, there are many hyperparameters related to model construction and optimization. AE training and performance often benefit from hyperparameter tuning to avoid over and underfitting.

Bayesian optimization (BO) is a state-of-the-art hyperparameter optimization algorithm that reached competitive performances on several optimizations benchmarks (Snoek et al., 2012; Bergstra and Bengio, 2012). BO is a technique based on Bayes theorem to direct a search for a global optimization problem that is efficient and effective. It works by building a probabilistic model of the objective function, called the surrogate function, that is then searched efficiently with an acquisition function before candidate samples are chosen for evaluation on the real objective function.

2.3 Textual Features

Character n-grams are fed as an input to the SAE. In comparison to word n-grams, which only capture the identity of a word and its possible neighbors, character n-grams are additionally capable of detecting the morphological makeup of a word (Wei et al., 2009; Kulmizev et al., 2017). The extracted n-gram features are input to the deep SAE as shown in the Figure 1. The deep SAE contains multiple hidden layers. Hyperparameters were optimized using BO.

3 Experimental Setup and Datasets

3.1 Hyperparameters

To verify the robustness of our proposed model, we have used datasets that are either short, contain similar dialects, or cover multiple languages, and long texts. The range of values for the hyperparameters search space is shown in Table 1. During training, BO chooses the best hyperparameters from this range. The overall configuration of the SAE model is shown in Table 2.

3.2 Datasets

A summary table of the number of texts per dataset is presented in Table 3. We also provide a brief description of each dataset.

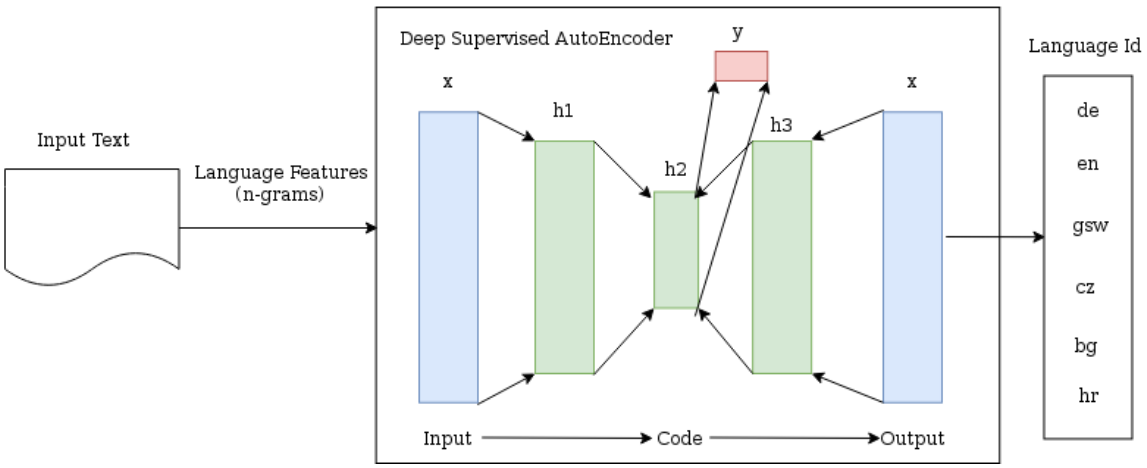

Figure 1: Proposed model architecture. The extracted features of the text are input to the supervised autoencoder. The target "y" are included. The classification output are the language id for the classified languages.

Hyper Parameter	Range
number of layer	1-5
learning rate	$10^{-5} - 10^{-2}$
weight decay	$10^{-6} - 10^{-3}$
activation functions	'relu', 'sigma'

Table 1: Search space hyper parameter range.

Parameter	DSL	Ling10	ILI
`n_gram` range	1-3	1-3	1-3
number of target	14	10	5
embedding dimension	300	300	300
supervision	'clf'	'clf'	'clf'
converge threshold	0.00001	0.00001	0.00001
number of epochs	300	500	500

Table 2: SAE model configurations for the dataset.

Dataset	Training	Development	Test
DSL	252,000	28,000	14,000
Ling10	140,000	-	50,000
ILI	70,351	10,329	9,692

Table 3: Dataset Statistics.

DSL Dataset: The data obtained from the "Discriminating between Similar Language (DSL) Shared Task 2015" contains 13 different languages belonging to 6 language groups, namely South Eastern Slavic (Bulgarian and Macedonian), South Western Slavic (Bosnian, Croatian and Serbian), West-Slavic (Czech and Slovak), Ibero-Romance Spanish (Peninsular Spanish and Argentinian Spanish), Ibero-Romance Portuguese (Brazilian Portuguese and European Portuguese), and Austronesian (Indonesian and Malay). The DSL corpus collection [1] have different versions based on different language groups, representing a benchmark dataset for evaluating language identification systems (Tan et al., 2014a). We used the DSLCCv2.0 [2] to perform our experiments (Tan et al., 2014b). In this version, the training set contains 18,000 sentences for each language and the development set contain 2,000 sentences in each language.

Ling10 Dataset: The Ling10 dataset [3] contains 190,000 sentences categorized into 10 languages (English, French, Portuguese, Chinese Mandarin, Russian, Hebrew, Polish, Japanese, Italian, Dutch) mainly used for language detection and benchmarking natural language processing (NLP) algorithms. It has three variants and we have considered "Ling10-train large" in our experiments.

ILI Dataset: The Indo-Aryan Language Identification (ILI) dataset used for the fifth workshop on NLP for similar languages, varieties and dialects (VarDial) at COLING 2018 [4] (Zampieri et al., 2018b). This task was aimed at identifying 5 closely-related languages of the Indo-Aryan language family – Hindi (also known as Khari Boli), Braj Bhasha, Awadhi, Bhojpuri, and Magahi. Considering Indian geographical location and states,

[1] http://ttg.uni-saarland.de/resources/DSLCC/

[2] https://github.com/Simdiva/DSL-Task/tree/master/data/DSLCC-v2.0

[3] https://github.com/johnolafenwa/Ling10

[4] https://github.com/kmi-linguistics/vardial2018

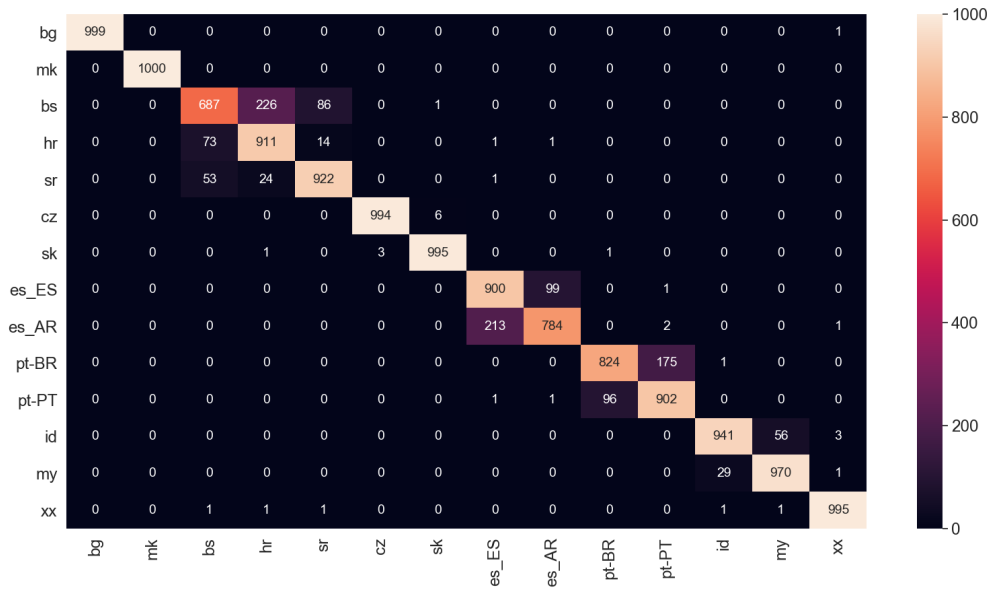

Figure 2: Confusion matrix for the DSL test dataset.

these languages form part of a continuum starting from Western Uttar Pradesh (Hindi and Braj Bhasha) to Eastern Uttar Pradesh (Awadhi and Bhojpuri) and the neighboring Eastern state of Bihar (Bhojpuri and Magahi).

4 Results and Discussion

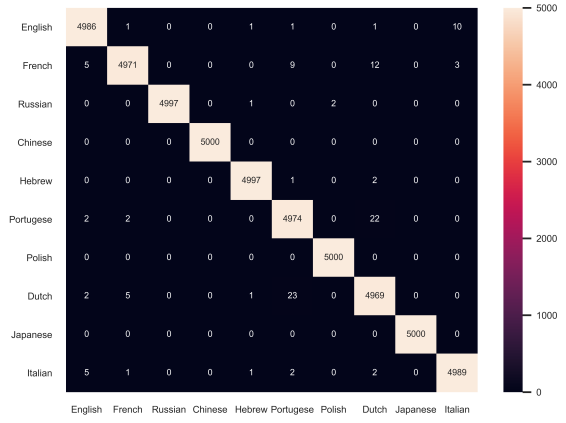

Figure 3: Confusion matrix for Ling10 test dataset.

The SAE model performance for the used dataset is shown in Table 4. Since we are interested in potential confusion between languages, we plot the confusion matrices for the DSL (Figure 2), Ling10 (Figure 3), and for the ILI (Figure 4) datasets.

Observe that for the case of Ling10 (dissimilar

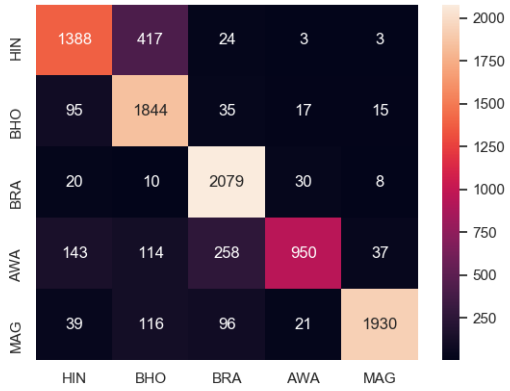

Figure 4: Confusion matrix for ILI test dataset.

language families) the SAE approach performs almost perfect (few confusions between Dutch and Portuguese and Dutch and French) reaching an accuracy of 100%. On the contrary, for DSL and ILI datasets, we can notice more errors. For example, in DSL, there are many mistakes between Spanish, Portuguese, and South Western Slavic families respectively, nevertheless, our SAE gets an accuracy of 92%. Similarly, observe the complexity of the task in the ILI dataset, where the obtained accuracy was of 85%.

As a comparison point, the best-reported result for the DSL dataset is based on a classifier ensemble approach (using 8 SVM classifiers); each one

trained on a single feature type and reached an accuracy of 95.54 % in the test partition during the DSL 2015 shared task using the DSLv2.0 dataset (Malmasi and Dras, 2015; Zampieri et al., 2015). For the case of the ILI dataset, the best score reported during the Second VarDial Evaluation Campaign was of 95% F1-macro (Zampieri et al., 2018a). The winning approach is based on adaptive language models based on character n-grams from 1 to 6 (Jauhiainen et al., 2018). Contrary to these approaches, the proposed SAE represents a much less complex and competitive alternative, obtaining good performance results.

		Accuracy	
Model	Dataset	Validation	Test
SAE (char-3gram)	Ling10	-	100 %
SAE (char-3gram)	DSL	92%	92%
SAE (char-3gram)	ILI	94%	85%

Table 4: Overall performance of the proposed approach.

4.1 Discussion

SAE is less computationally expensive than other deep-learning architectures, while it generalizes well to a wide variety of languages and dialects. The proposed model is extendable by creating a host of features such as character n-gram, word n-gram, word counts, etc, and then passing it through AE to choose the best features. As future work, we are planning to i) verify our model (SAE + BO) with other language detection data sets ii) try to create a dialect detection dataset for other Indian languages and apply SAE for classifying the dialects.

5 Conclusion

In this paper, we introduced SAE with BO for language detection using N-grams at the character level and illustrated its performance on the discrimination of very close languages or dialects on several well-known corpora. We also presented some advantages of the proposed approach, and discuss some of the future directions for SAE-based language detection. [5]

Acknowledgments

This work was supported by the European Union's Horizon 2020 research and innovation program under grant agreement No. 833635 (project ROXANNE: Real-time network, text, and speaker analytics for combating organized crime, 2019-2022). The second author, Esaú Villatoro-Tello, was supported partially by Idiap Research Institute, SNI-CONACyT, CONACyT project grant CB-2015-01-258588, and UAM-C Mexico during the elaboration of this work.

[5] SAE code is available here

References

Ivana Balazevic, Mikio Braun, and Klaus-Robert Müller. 2016. Language detection for short text messages in social media. *arXiv preprint arXiv:1608.08515*.

James Bergstra and Yoshua Bengio. 2012. Random search for hyper-parameter optimization. *Journal of machine learning research*, 13(Feb):281–305.

Nagaratna B Chittaragi and Shashidhar G Koolagudi. 2019. Automatic dialect identification system for kannada language using single and ensemble svm algorithms. *Language Resources and Evaluation*, pages 1–33.

Tommi Jauhiainen, Heidi Jauhiainen, and Krister Lindén. 2018. Iterative language model adaptation for Indo-Aryan language identification. In *Proceedings of the Fifth Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial 2018)*, pages 66–75, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Tommi Jauhiainen, Krister Lindén, and Heidi Jauhiainen. 2019a. Language model adaptation for language and dialect identification of text. *Natural Language Engineering*, 25(5):561–583.

Tommi Sakari Jauhiainen, Marco Lui, Marcos Zampieri, Timothy Baldwin, and Krister Lindén. 2019b. Automatic language identification in texts: A survey. *Journal of Artificial Intelligence Research*, 65:675–782.

Tom Kocmi and Ondřej Bojar. 2017. Lanidenn: Multilingual language identification on character window. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 927–936.

Artur Kulmizev, Bo Blankers, Johannes Bjerva, Malvina Nissim, Gertjan van Noord, Barbara Plank, and Martijn Wieling. 2017. The power of character n-grams in native language identification. In *Proceedings of the 12th Workshop on Innovative Use of NLP for Building Educational Applications*, pages 382–389.

Lei Le, Andrew Patterson, and Martha White. 2018. Supervised autoencoders: Improving generalization performance with unsupervised regularizers. In *Advances in Neural Information Processing Systems*, pages 107–117.

Marco Lui, Jey Han Lau, and Timothy Baldwin. 2014. Automatic detection and language identification of multilingual documents. *Transactions of the Association for Computational Linguistics*, 2:27–40.

Shervin Malmasi and Mark Dras. 2015. Language identification using classifier ensembles. In *Proceedings of the joint workshop on language technology for closely related languages, varieties and dialects*, pages 35–43.

Ermelinda Oro, Massimo Ruffolo, and Mostafa Sheikhalishahi. 2018. Language identification of similar languages using recurrent neural networks. In *ICAART*.

Shantipriya Parida, Esaú VILLATORO-TELLO, Sajit Kumar, Petr Motlicek, and Qingran Zhan. 2020. Idiap submission to swiss-german language detection shared task. In *Proceedings of the 5th Swiss Text Analytics Conference (SwissText) & 16th Conference on Natural Language Processing (KONVENS)*, CONF. CEUR Workshop Proceedings.

Marc'Aurelio Ranzato and Martin Szummer. 2008. Semi-supervised learning of compact document representations with deep networks. In *Proceedings of the 25th international conference on Machine learning*, pages 792–799.

Antti Rasmus, Mathias Berglund, Mikko Honkala, Harri Valpola, and Tapani Raiko. 2015. Semi-supervised learning with ladder networks. In *Advances in neural information processing systems*, pages 3546–3554.

Yves Scherrer and Owen Rambow. 2010. Natural language processing for the swiss german dialect area. In *Semantic Approaches in Natural Language Processing-Proceedings of the Conference on Natural Language Processing 2010 (KONVENS)*, pages 93–102. Universaar.

Jasper Snoek, Hugo Larochelle, and Ryan P Adams. 2012. Practical bayesian optimization of machine learning algorithms. In *Advances in neural information processing systems*, pages 2951–2959.

Monorama Swain, Aurobinda Routray, P Kabisatpathy, and Jogendra N Kundu. 2016. Study of prosodic feature extraction for multidialectal odia speech emotion recognition. In *2016 IEEE Region 10 Conference (TENCON)*, pages 1644–1649. IEEE.

Liling Tan, Marcos Zampieri, Nikola Ljubešic, and Jörg Tiedemann. 2014a. 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.

Liling Tan, Marcos Zampieri, Nikola Ljubešic, and Jörg Tiedemann. 2014b. 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, Reykjavik, Iceland.

Esaú Villatoro-Tello, Shantipriya Parida, Sajit Kumar, Petr Motlicek, and Qingran Zhan. 2020a. Idiap & uam participation at germeval 2020: Classification and regression of cognitive and motivational style from text. In *Proceedings of the GermEval 2020 Task 1 Workshop in conjunction with the 5th SwissText & 16th KONVENS Joint Conference*, pages 11–16.

Esaú Villatoro-Tello, Gabriela Ramírez-de-la Rosa, Sajit Kumar, Shantipriya Parida, and Petr Motlicek. 2020b. Idiap and uam participation at mex-a3t evaluation campaign. In *Notebook Papers of 2nd SEPLN Workshop on Iberian Languages Evaluation Forum (IberLEF), Malaga, Spain*.

Zhihua Wei, Duoqian Miao, Jean-Hugues Chauchat, Rui Zhao, and Wen Li. 2009. N-grams based feature selection and text representation for chinese text classification. *International Journal of Computational Intelligence Systems*, 2(4):365–374.

Jennifer Williams and Charlie Dagli. 2017. Twitter language identification of similar languages and dialects without ground truth. In *Proceedings of the Fourth Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial)*, pages 73–83.

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. 2018a. Language identification and morphosyntactic tagging: The second VarDial evaluation campaign. In *Proceedings of the Fifth Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial 2018)*, pages 1–17, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Marcos Zampieri, Preslav Nakov, Nikola Ljubešić, Jörg Tiedemann, Shervin Malmasi, and Ahmed Ali, editors. 2018b. *Proceedings of the Fifth Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial 2018)*. Association for Computational Linguistics, Santa Fe, New Mexico, USA.

Marcos Zampieri, Liling Tan, Nikola Ljubešić, Jörg Tiedemann, and Preslav Nakov. 2015. Overview of the dsl shared task 2015. In *Proceedings of the Joint Workshop on Language Technology for Closely Related Languages, Varieties and Dialects*, pages 1–9.

Qiuyu Zhu and Ruixin Zhang. 2019. A classification supervised auto-encoder based on predefined evenly-distributed class centroids. *arXiv preprint arXiv:1902.00220*.