DTeam @ VarDial 2019: Ensemble based on skip-gram and triplet loss neural networks for Moldavian vs. Romanian cross-dialect topic identification

Diana-Elena Tudoreanu
University of Bucharest
14 Academiei, Bucharest, Romania
dianatudoreanu@gmail.com

Abstract

This paper presents the solution proposed by DTeam in the VarDial 2019 Evaluation Campaign for the Moldavian vs. Romanian cross-topic identification task. The solution proposed is a Support Vector Machines (SVM) ensemble composed of a two character-level neural networks. The first network is a skip-gram classification model formed of an embedding layer, three convolutional layers and two fully-connected layers. The second network has a similar architecture, but is trained using the triplet loss function. The results obtained on the test set show a macro-F_1 score of 0.89 for subtask 1 (binary classifications of the Moldavian and Romanian dialects), which places us on the first place among 5 teams. For subtask 2 (classifying Romanian samples into topics while training on Moldavian samples), we obtained a macro-F_1 of 0.39, which places us on the third place. For subtask 3 (classifying Moldavian samples into topics while training on Romanian samples), we obtained a macro-F_1 of 0.44, which places us once again on the third place.

1 Introduction

The VarDial 2019 Evaluation Campaign (Zampieri et al., 2019) proposes a Moldavian vs. Romanian cross-topic (MRC) identification problem, comprised of three tasks. The first task is a binary classification by dialect, meaning that a classifier would have to differentiate between Romanian and Moldavian dialects. The second and third tasks are cross-dialect multi-class categorization by topic tasks. The second tasks is classifying Romanian samples into topics while training on Moldavian samples and the third is classifying Moldavian samples into topics while training on Romanian samples. The samples for both training and testing are provided with the MOROCO – Moldavian and Romanian Dialectal Corpus – dataset (Butnaru and Ionescu, 2019).

The MOROCO dataset contains over 33k text samples in both Romanian and Moldavian collected from news domains covering six topics: culture, finance, politics, science, sports, tech. The samples were divided into training (21k), validation (6k), and test (6k) samples. For the VarDial 2019 Evaluation Campaign, the validation and test sets were combined into a single development set. The organizers provided an additional test set of 6k samples. In each text sample, proper nouns were replaced with a token, namely "NE" in order to prevent classifiers from taking decisions based on country-specific nouns.

Our approach for the MRC shared task is to build an ensemble model that combines two character-level neural networks through an SVM (Cortes and Vapnik, 1995) classifier. The first network is a skip-gram classification model formed of an embedding layer, three convolutional layers and two fully-connected layers. The second network has a similar architecture, but is trained using the triplet loss function (Schroff et al., 2015). We participated in all three MRC subtasks and we managed to rank on the first place in the first subtask (Moldavian vs. Romanian dialect identification), with a macro-F_1 score of 0.89, surpassing the other five participants by more than 10%. Due to the lack of time, we did not manage to properly train our models for subtasks 2 and 3. Consequently, we ranked after the other two participants in subtasks 2 and 3.

The rest of this paper is organized as follows. Related art on dialect identification is presented in Section 2. Our approach is presented in Section 3. The empirical results are presented in Section 2. Conclusions and future work directions are presented in Section 5.

Proceedings of VarDial, pages 202–208
Minneapolis, MN, June 7, 2019

2 Related Work

In recent years, there have been many approaches proposed for discriminating dialects (Ali, 2018; Ali et al., 2016; Belinkov and Glass, 2016; Butnaru and Ionescu, 2018; Çöltekin and Rama, 2016, 2017; Goutte and Léger, 2016; Ionescu and Popescu, 2016; Ionescu and Butnaru, 2017; Kumar et al., 2018; van der Lee and van den Bosch, 2017). While some of these approaches extract handcrafted features and apply linear classifiers on top (Butnaru and Ionescu, 2018; Ionescu and Popescu, 2016; Ionescu and Butnaru, 2017), other approaches are based on deep learning techniques (Ali, 2018). Although deep neural networks attain top results in many NLP tasks, e.g. machine translation (Gehring et al., 2017), language modelling (Kim et al., 2016; Dauphin et al., 2017), part-of-speech tagging (Santos and Zadrozny, 2014), it appears that shallow approaches attain superior results in dialect identification, at least according to the previous VarDial evaluation campaigns (Malmasi et al., 2016; Zampieri et al., 2017, 2018). Although the evidence points in this direction, our method is based on testing new deep learning models for dialect identification, that have the potential to improve the results. Our interest is focused on neural networks trained using triplet loss, which has not been applied before in dialect identification, to our knowledge.

Closer to our work, Butnaru and Ionescu (2019) proposed a dataset and several models to discriminate between Moldavian and Romanian dialects. The authors proposed two approaches that use character-level features, inspired by previous VarDial evaluation campaigns (Malmasi et al., 2016; Zampieri et al., 2017, 2018), in which dialect identification methods (Ali, 2018; Belinkov and Glass, 2016; Butnaru and Ionescu, 2018) based on character n-grams attained top ranks. Their first approach is a shallow model based on string kernels (Butnaru and Ionescu, 2018; Cozma et al., 2018; Ionescu and Popescu, 2016; Ionescu and Butnaru, 2017; Ionescu et al., 2016) and Kernel Ridge Regression. Their second approach is based on convolutional neural networks with squeeze-and-excitation blocks. Different from Butnaru and Ionescu (2019), we explore only deep learning approaches, by combining two neural networks trained using different loss functions. Similar to Butnaru and Ionescu (2019), our networks take as input character encodings.

3 Methodology

3.1 Preprocessing

Each sample in the MOROCO dataset is preprocessed using the same method regardless of the subtask or the algorithm used. We have reduced the alphabet size down to 85 characters consisting of uppercase and lowercase Romanian letters, digits and commonly used symbols. Each unused character is replaced with a blank character and the named entity token NE is replaced with a single character.

3.2 Feature Extraction

For the competition results, we added an embedding layer consisting of a $85 * 128$ matrix, where 85 is the alphabet size and 128 the embedding size. We apply a one-hot transformation to the input, then multiply the one-hot vector with the embedding layer. This layer is trained at the same time with the neural networks.

For the post-competition results, we built a skip-gram model (Mikolov et al., 2013) based on character n-grams. The model is trained on the top 40k 5-grams from the corpus, in order to learn the n-gram embeddings of the most common n-grams. The embedding size of each n-gram is set to 150. We pre-train the skip-gram model using sub-sampling and negative sampling techniques, which are shown to improve accuracy and convergence speed.

During pre-training, each text sample is divided into contiguous substrings of 5000 characters. When a text sample has less than 5000 characters, we apply 0-padding at the right. During inference, we keep the first 5000 characters of each text sample, also 0-padding the shorter strings.

In order to build our representation, we apply the skip-gram model to each text sample, by replacing every 5-gram with its corresponding embedding learned by the skip-gram model. If the 5-gram is not in the top 40k, it is replaced by a zero vector of size 150. After generating the representation corresponding to each text sample, we provide it as input for training our two neural networks.

3.3 Ensemble of Neural Networks

The method used for predicting the labels on the test is based on an SVM ensemble that combines the predictions of two deep neural networks, a triplet loss network and a skip-gram convolutional

Subtask	macro-F1	weighted-F1	accuracy
1	0.9296	0.9301	0.9302
2	0.6594	0.6596	0.6672
3	0.7621	0.8094	0.8114

Table 1: Results of our ensemble method on the development set of the MRC shared task comprised of three subtasks.

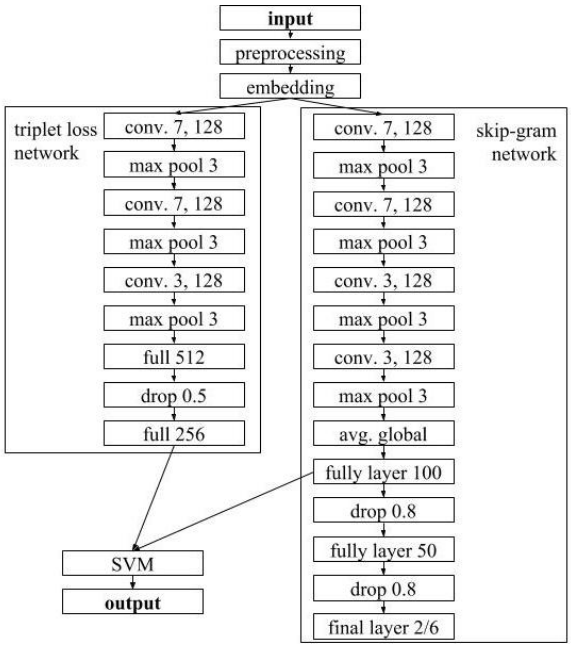

Figure 1: Our processing pipeline.

network. The ensemble schema is pictured in Figure 1.

The triplet loss neural network learns an embedding of size 256 for each sentence. The network is trained using the triplet loss (Schroff et al., 2015). The goal of a triplet loss network is to minimize the distance in the embedding space between two samples of the same class and maximize the distance between two samples of different classes. In other words, we want the distance between the differently labeled samples to be larger than the distance between same labeled samples by a margin α.

Given a triplet composed of an anchor sample, a positive sample, and a negative one, the triplet loss tries to minimize the distance between the anchor and the positive sample, while maximizing at the same time the distance between the anchor and the negative sample. Formally, the loss that we are trying to minimize is:

$$\mathcal{L}(\theta, a, p, n) = \max(d(a, p) - d(a, n) + \alpha, 0), \quad (1)$$

where θ represents the weights learned by the neural network, a is the anchor sample, p is the positive sample, n is the negative sample and α is the margin. In our experiments, we set the margin hyper-parameter to $\alpha = 1$. As distance metric, we use the Euclidean distance:

$$d(x, y) = \sqrt{(x_1 - y_1)^2 + ... + (x_m - y_m)^2}, \quad (2)$$

where m is the number of features in vectors x and y. Each iteration consists of a mini-batch of 30 triplets (a, p, n) from the training samples. The number of iterations is 9000 on all sub-tasks and was chosen based on the loss obtained on the development samples. After every epochs, the training data is shuffled.

The triplet loss network has 3 convolutional layers, each one of 128 unidimensional filters. The filter support on each convolutional layer is 7, 7, and 3 respectively. Each convolutional layer is followed by an unidimensional max pooling layer of size 3. The last max pooling layer is followed by a fully connected layer of size 512, having a dropout rate of 0.5. The network ends with a fully connected layer of 256 neurons, which represent the final embedding of the input text sample. Both the fully and the convolutional layers have a leaky Rectified Linear Unit (ReLU) activation function. The learning algorithm is the Adam optimizer and the learning rate is set to 0.00005.

The skip-gram convolutional network has a similar architecture, with a extra convolutional layer. There are 4 convolutional layers with unidimensional filters of sizes 7, 7, 3, and 3, respectively. Each convolutional layer is followed by a unidimensional max pooling of size 3. This is followed by an average global pooling layer, which reduces the size of the representation. Two fully-connected layers with 100 and 50 neurons, respectively, come after the global pooling layer. Each fully-connected layer has a dropout rate of 0.8. Finally, the last layer is composed of 2 neurons for the first MRC subtask, each corresponding to one dialect (Moldavian or Romanian). For the second

and the third MRC substaks, the last layer is composed of 6 neurons, each corresponding to one of the six topics. The last layer is based on softmax loss. Similarly to the triplet loss network, the layers have leaky ReLU activation functions and the optimization algorithm is Adam. In this case, the learning rate is 0.0001 and the mini-batch size is 120 samples. The number of iterations is 9000 for the first sub-task, 10000 for the second and 12000 for the third, with the data being reshuffled every epoch.

The input of the ensemble is composed of the last layer of the triplet loss network concatenated with the intermediate fully-connected layer of size 100 of the classification network. The final model is represented by an SVM classifier. The hyper-parameters of the SVM (kernel type and C) are selected through grid search on the development set.

4 Experiments

4.1 Dataset

The MOROCO dataset used in the MRC shared task contains over 39k text samples in both Romanian and Moldavian collected from news domains covering six topics: culture, finance, politics, science, sports, tech. The samples were divided into training (21k), development (12k), and test (6k) samples. In each text sample, proper nouns were replaced with a token, namely "NE" in order to prevent classifiers from taking decisions based on country-specific nouns.

4.2 Parameter Tuning

We apply grid search on the development set to tune the hyper-parameters of the final SVM ensemble. The parameters considered are 0.1, 0.5, 1, and 5 for the regularization parameter C, and *RBF* or *linear* for the kernel. We use the implementation available in LibSVM (Chang and Lin, 2011). Upon applying grid search, we found the following optimal parameters: the regularization parameter C is equal to 0.5, the kernel type is RBF. The corresponding parameter of the RBF kernel, γ, is set to 0.001.

The accuracy of the classification model were better than the triplet loss network in all tasks and in some cases the ensemble did not achieve better accuracy than the classifier, as shown in Table 1.

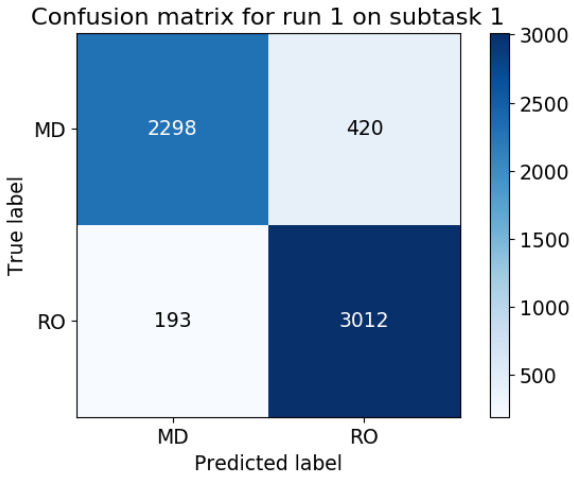

Figure 2: Confusion matrix of our ensemble model on the first MRC subtask.

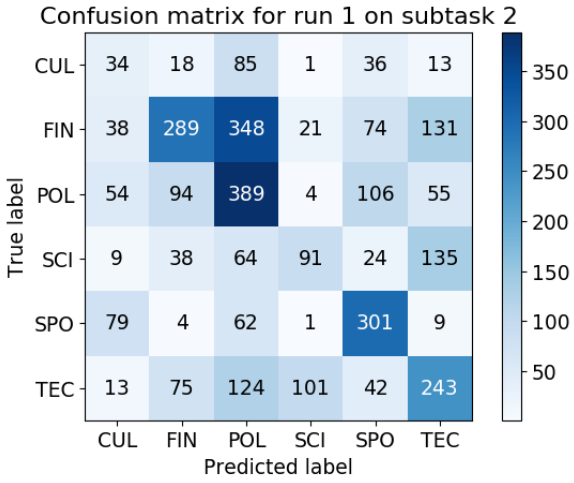

Figure 3: Confusion matrix of our ensemble model on the second MRC subtask.

4.3 Results

Our results on the test set of the MRC shared task are presented in Table 2. The results obtained on the test set show a macro-F_1 score of 0.89 for subtask 1 (binary classification of the Moldavian and Romanian dialects), which places us on the first place among 5 teams. Compared to the second team, our performance is 0.09 higher. The corresponding confusion matrix is illustrated in Figure 2. We notice that the number of misclassified Moldavian samples is twice the number of misclassified Romanian samples. This suggests that the algorithm is slightly biased in favor of Romanian.

For subtask 2 (classifying Romanian samples into topics while training on Moldavian samples), we obtained a macro-F_1 of 0.39, which places us on the third place. Compared to the first team,

Subtask	macro-F1	weighted-F1	accuracy
1	0.8949	0.8960	0.8965
2	0.3856	0.4145	0.4202
3	0.4472	0.5440	0.5367

Table 2: Results of our ensemble method on the test set of the MRC shared task comprised of three subtasks.

Subtask	macro-F1	weighted-F1	accuracy
1	0.9340	0.9345	0.9346
2	0.6509	0.6531	0.6620
3	0.7521	0.8004	0.8027

Table 3: Post-competition results of our ensemble method on the test set of the MRC shared task comprised of three subtasks.

Subtask	macro-F1	weighted-F1	accuracy
1	0.9334	0.9340	0.9341
2	0.6306	0.6321	0.6383
3	0.7360	0.7893	0.7895

Table 4: Post-competition results of the neural network based on softmax loss on the test set of the MRC shared task comprised of three subtasks.

Subtask	macro-F1	weighted-F1	accuracy
1	0.8690	0.8701	0.8705
2	0.6350	0.6368	0.6468
3	0.7308	0.7816	0.7855

Table 5: Post-competition results of the neural network based on triplet loss on the test set of the MRC shared task comprised of three subtasks.

Subtask	Method	macro-F1	weighted-F1	accuracy
1	CNN	0.9275	0.9276	0.9271
	Ensemble	0.9340	0.9345	0.9346
2	CNN	0.5504	0.5627	0.5367
	Ensemble	0.6509	0.6531	0.6620
3	CNN	0.7249	0.7160	0.6270
	Ensemble	0.7521	0.8004	0.8028

Table 6: Comparison between post-competition results of the ensemble and our reimplementation of the character-level CNN of Butnaru and Ionescu (2019) on the test set of the MRC shared task.

our performance is 0.2 lower. The corresponding confusion matrix is illustrated in Figure 3. The confusion matrix suggests a bias towards the *politics* label, one of the reasons being that the number of Moldavian politics samples is six times higher than the Romanian ones. We also notice a strong confusion between *finance* and *politics*. The classes best classified are *politics* and *sports* while the worse classified are *science* and *culture*. It seems that *science* is often confused for *technology*.

For subtask 3 (classifying Moldavian samples into topics while training on Romanian samples), we obtained a macro-F_1 of 0.44, which places us once again on the third place. Compared to the first team, our performance is 0.08 lower. The corresponding confusion matrix is illustrated in Figure 4. Similarly to the second subtask, our model provides better precision in classifying *politics* and *sports*, followed closely by *finance*. This time, *politics* is being labeled as *finance* more frequently, as opposed to what we notice in Figure

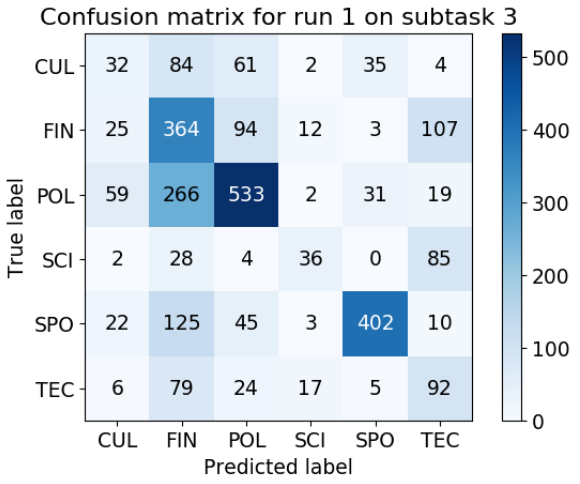

Figure 4: Confusion matrix of our ensemble model on the third MRC subtask.

3. Again, *culture* and *science* are misclassified the most, with *science* being labeled as *technology* and *culture* as either *finance* or *politics*.

Given the overall results, we can conclude that the models were better in discriminating between dialects than in distinguishing the topics in cross-dialect settings. A possible reason for this is the fact that the embedding layer is trained at the same time as the networks and it might require more training for cross-dialect multi-class classifiers. We notice that when we pre-train the skip-gram (in post-competition) we obtain less discrepancy between results.

4.4 Post-competition Results

We report post-competition results in Table 3. The ensemble model shows a macro-F_1 score of 0.93 for subtask 1, a 0.66 score for subtask 2 and a 0.80 score for subtask 3. The increased results are due to the pre-trained skip-gram applied to the data as opposed to the embedding layer. Results of individual models are presented in Tables 4 and 5. Although the individual triplet loss network does not attain good results by itself, it provides useful information to the ensemble.

We also offer a comparison to the results of the character-level CNN method in Butnaru and Ionescu (2019), since it is closer to our work, in Table 6.

5 Conclusion

We conclude that our ensemble model does a good job for binary classification, while the results for the last two subtasks can be further improved. Some future work could be testing other classification algorithms for the final ensemble, such as Logistic Regression.

Acknowledgments

We thank reviewers for their useful comments.

References

Ahmed Ali, Najim Dehak, Patrick Cardinal, Sameer Khurana, Sree Harsha Yella, James Glass, Peter Bell, and Steve Renals. 2016. Automatic Dialect Detection in Arabic Broadcast Speech. In *Proceedings of INTERSPEECH*, pages 2934–2938.

Mohamed Ali. 2018. Character level convolutional neural network for arabic dialect identification. In *Proceedings of VarDial*, pages 122–127.

Yonatan Belinkov and James Glass. 2016. A Character-level Convolutional Neural Network for Distinguishing Similar Languages and Dialects. In *Proceedings of VarDial*, pages 145–152.

Andrei Butnaru and Radu Tudor Ionescu. 2019. MOROCO: The Moldavian and Romanian Dialectal Corpus. *arXiv preprint arXiv:1901.06543*.

Andrei M. 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.

Çağri Çöltekin and Taraka Rama. 2016. Discriminating Similar Languages with Linear SVMs and Neural Networks. In *Proceedings of VarDial*, pages 15–24, Osaka, Japan.

Çağr 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, Valencia, Spain.

Chih-Chung Chang and Chih-Jen Lin. 2011. LibSVM: A Library for Support Vector Machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27:1–27:27. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

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

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

Yann Dauphin, Angela Fan, Michael Auli, and David Grangier. 2017. Language Modeling with Gated Convolutional Networks. In *Proceedings of ICML*, pages 933–941.

Jonas Gehring, Michael Auli, David Grangier, and Yann Dauphin. 2017. A Convolutional Encoder Model for Neural Machine Translation. In *Proceedings of ACL*, pages 123–135.

Cyril Goutte and Serge Léger. 2016. Advances in Ngram-based Discrimination of Similar Languages. In *Proceedings of VarDial*, pages 178–184, Osaka, Japan.

Radu Tudor Ionescu and Andrei M. Butnaru. 2017. Learning to Identify Arabic and German Dialects using Multiple Kernels. In *Proceedings of VarDial*, pages 200–209.

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. 2016. String kernels for native language identification: Insights from behind the curtains. *Computational Linguistics*, 42(3):491–525.

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

Ritesh Kumar, Bornini Lahiri, Deepak Alok, Atul Kr. Ojha, Mayank Jain, Abdul Basit, and Yogesh Dawar. 2018. Automatic Identification of Closely-related Indian Languages: Resources and Experiments. In *Proceedings of LREC*.

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.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient Estimation of Word Representations in Vector Space. In *Proceedings of ICLR*.

Cicero D. Santos and Bianca Zadrozny. 2014. Learning character-level representations for part-of-speech tagging. In *Proceedings of ICML*, pages 1818–1826.

Florian Schroff, Dmitry Kalenichenko, and James Philbin. 2015. FaceNet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 815–823.

Chris van der Lee and Antal van den Bosch. 2017. Exploring Lexical and Syntactic Features for Language Variety Identification. In *Proceedings of VarDial*, pages 190–199, Valencia, Spain.

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, 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 Butnaru, and Tommi Jauhiainen. 2019. A Report on the Third VarDial Evaluation Campaign. In *Proceedings of VarDial*.