Investigating Machine Learning Methods for Language and Dialect Identification of Cuneiform Texts

Ehsan Doostmohammadi and **Minoo Nassajian**

Computational Lingusitics Group,
Sharif University of Technology, Tehran, Iran
{e.doostm72, m.nassajian2016}@student.sharif.edu

Abstract

Identification of the languages written using cuneiform symbols is a difficult task due to the lack of resources and the problem of tokenization. The Cuneiform Language Identification task in VarDial 2019 addresses the problem of identifying seven languages and dialects written in cuneiform; Sumerian and six dialects of Akkadian language: Old Babylonian, Middle Babylonian Peripheral, Standard Babylonian, Neo-Babylonian, Late Babylonian, and Neo-Assyrian. This paper describes the approaches taken by `SharifCL` team to this problem in VarDial 2019. The best result belongs to an ensemble of Support Vector Machines and a naive Bayes classifier, both working on character-level features, with macro-averaged F_1-score of 72.10%.

1 Introduction

A wide range of Natural Language Processing (NLP) tasks, such as Machine Translation (MT), speech recognition, information retrieval, data mining, and creating text resources for low-resource languages benefit from the upstream task of language identification. The Cuneiform Language Identification (CLI) task in VarDial 2019 (Zampieri et al., 2019) tries to address the problem of identifying languages and dialects of the texts written in cuneiform symbols.

Identifying languages and dialects of the cuneiform texts is a difficult task, since such languages lack resources and also there is the problem of tokenization. Although there are some work addressing the problem of tokenization in some of these languages or dialects, there is not any universal method or tool available for tokenization of cuneiform texts, as such a task depends on the rules of that language, simply because cuneiform writing system is a syllabic as well as a logographic one. As a result, all the endeavors in this paper are based on character-level features. This work investigates different machine learning methods which are proven to be effective in text classification and compares them by their obtained F_1-score, accuracy, and training time.

In this paper, we first review the literature of language identification and the work on languages written using cuneiform writing system in 2, introduce the models used to tackle the problem of identifying such languages and dialects in 3, describe the training data in 4, and discuss the results in 5.

2 Related Work

The majority of research conducted in the field of language identification has been on textual data. However, there are some studies focusing on speech samples, such as (Hategan et al., 2009; Ali et al., 2015; Malmasi and Zampieri, 2016). Language identification systems are meant to distinguish between similar languages (Goutte et al., 2016; Williams and Dagli, 2017), language varieties (Rangel et al., 2016; Castro et al., 2017), or a set of different dialects of the same language (Malmasi et al., 2016; El Haj et al., 2018). There has also been the annually held VarDial workshop since 2014, which deals with computational methods and language resources for closely related languages, language varieties, and dialects (Zampieri et al., 2017, 2018).

Various kinds of features are used to train these systems, including bytes and encodings (Singh and Gorla, 2007; Brown, 2012), characters (van der Lee and van den Bosch, 2017; Samih and Kallmeyer, 2017), morphemes (Gomez et al., 2017; Barbaresi, 2016), and words (Duvenhage et al., 2017; Clematide and Makarov, 2017).

The most recent studies use different language identification methods, such as decision trees

Proceedings of VarDial, pages 188–193

(Bora and Kumar, 2018), Bayesian network classifiers (Rangel et al., 2016), similarity measures (such as the out-of-place method (Jauhiainen et al., 2017), local ranked distance (Franco-Salvador et al., 2017), and cross entropy (Hanani et al., 2017)), SVM (Alrifai et al., 2017), and neural networks (Chang and Lin, 2014; Cazamias et al., 2015; Jurgens et al., 2017; Kocmi and Bojar, 2017).

To the extent of our knowledge, there is no work addressing the problem of language and dialect identification of cuneiform texts. Such languages, Sumerian and Akkadian for instance, are considered low-resource languages, meaning that there are only a few electronic resources for cuneiform processing. Some of these datasets include (Yamauchi et al., 2018) which developed a handwritten cuneiform character imageset, and (Chiarcos et al., 2018) which is an annotated cuneiform corpus with morphological, syntactic, and semantic tags. Furthermore, there are some early studies on rule-based morphological analyzers for these languages like (Kataja and Koskenniemi, 1988; Barthélemy, 1998; Macks, 2002; Barthélemy, 2009), and (Tablan et al., 2006).

Additionally, a small number of cuneiform text processing tasks have been carried out in which the transliterations of cuneiform characters were considered as the base feature. For instance, (Luo et al., 2015) adapted an unsupervised algorithm to recognize Sumerian personal names. Having transliterated the cuneiform corpus, they utilized the pre-knowledge and applied limited tags to pre-annotate the corpus. As another study, (Homburg and Chiarcos, 2016) conducted the first research on word segmentation on Akkadian cuneiform. They used three types of word segmentations algorithms including rule-based algorithms (such as bigram and prefix/suffix), dictionary-based algorithms (like MaxMatch, MaxMatchCombined, LCUMatching, MinWCMatch), and statistical and/or machine learning algorithms (such as C4.5, CRF, HMM, k-means, k Nearest Neighbors, MaxEnt, naive Bayes, multi-layer perceptron, and Support Vector Machines (SVM)) which work based on transliterations of cuneiform characters. The paper reports that the dictionary-based approaches obtained the best results. In addition, as one of the most recent studies on languages written in cuneiform, (Chiarcos et al., 2017) worked on a machine translation task. The used data consists of unannotated raw transliterations of Sumerian texts with their English translations. They use a morphological analyzer to extract word information to be used in the machine translation task. Moreover, a distantly supervised Part of Speech tagger and a dependency parser are applied to annotate data to facilitate the machine translation task.

3 Methodology

We investigated different machine learning methods, all of them based on character-level features, to tackle the problem. The following methods take 1- to 3-gram character TFIDF and 1- to 4-gram character count as input features and were implemented using Scikit-learn (Pedregosa et al., 2011):

- **SVM**: an SVM with a learning rate of $1e{-}6$, hinge loss, and `elasticnet` penalty, trained for 5 epochs with a random state of 11.
- **Naive Bayes**: a multinomial naive Bayes classifier with alpha of 0.14 and `fit_prior` as `True`.
- **Ensemble of SVM and naive Bayes**: a soft voting classifier which predicts the class label based on the argmax of the sums of the predicted probabilities of the SVM and the naive Bayes models.
- **Random Forest**: a random forest classifier with 25 estimators of depth 300.
- **Logistic Regression**: a logistic regression classifier with `lbfgs` optimizer, trained for 100 epochs.

We also experimented with deep learning approaches. The following two methods take character embeddings of size 32 for the 256 most common characters as input, and are trained using an Adam optimizer (Kingma and Ba, 2014) with batch size of 64 and learning rate of $1e{-}4$:

- **Convolutional Neural Network**: The concatenation of the output a set of parallel Convolutional Neural Network (CNN) layers, each with 32 filters and kernel size and stride of 2, 3, 4 and 5 which is fed to a a dense layer that maps to an $\mathbb{R}^{128}$ space and another one that maps to the $\mathbb{R}^{7}$ space of the labels. We also applied dropout with 0.5 keeping rate on CNNs output and another one

with the same keeping rate on the first dense layer's output.

- **Recurrent Neural Network**: A Recurrent Neural Network (RNN) with Long Short-Term Memory (LSTM) (Hochreiter and Schmidhuber, 1997) cell of size 256 and a dense layer mapping to an $\mathbb{R}^{128}$ space and another one mapping to the $\mathbb{R}^7$ space of the labels. We also applied dropout with 0.4 keeping rate on RNN's output and another one with 0.5 keeping rate on the first dense layer's output.

4 Data Description

The data of CLI shared task is described in (Jauhiainen et al., 2019). This data consists of 7 classes: Sumerian (SUX), Old Babylonian (OLB), Middle Babylonian peripheral (MPB), Standard Babylonian (STB), Neo-Babylonian (NEB), Late Babylonian (LTB), and Neo-Assyrian (NEA). Figure 1 shows the number of samples for each label in the training data. The whole training data consists of 139,421 samples. The development set comprises 668 and the test set 985 samples per label.

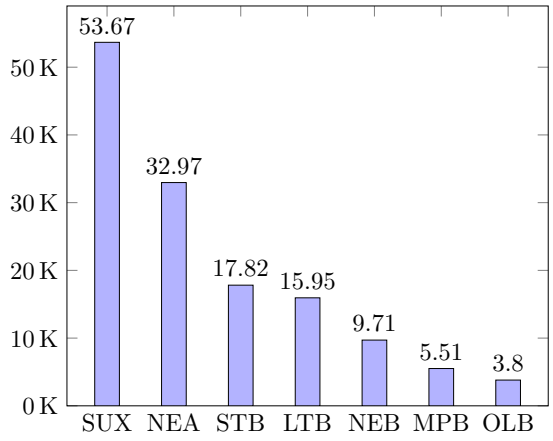

Figure 1: Number of samples for each label in the training set (in thousands).

Figure 1 shows that most of the training data belongs to SUX and NEA classes. Table 1 contains more detailed information on the data which shows that 86.35% of the data belongs to four classes of SUX, NEA, STB, LTB, whereas only 13.65% belongs to the other three.

5 Results and Discussion

Firstly, we trained the methods described in 3 and evaluated the models on development set. We continued with the best two methods, SVM and NB, and evaluated them on the test set. Table 2 shows the macro-averaged F_1-score, accuracy, and training time (in seconds) of the five non-deep and two deep methods on the development set. The non-deep models are trained using an `Intel(R) Core(TM) i7-7700K CPU @ 4.20GHz` CPU with 8 threads, and the deep ones using an `NVIDIA GeForce GTX 1080 Ti`.

Label	# of samples	% of all
SUX	53,673	38.49%
NEA	32,966	23.64%
STB	17,817	12.78%
LTB	15,947	11.44%
NEB	9,707	6.96%
MPB	5,508	3.95%
OLB	3,803	2.72%

Table 1: Number of samples in the training set for each label and their percentage of a total of 139,421 samples ordered from the highest to the lowest.

Method	F_1-score	Accuracy	T. Time
RF	0.5201	0.5615	264.14
LR	0.6861	0.6982	40.54
NB	*0.7194*	*0.7301*	**0.15**
SVM	0.7222	0.7309	1.67
Ens.	**0.7268**	**0.7356**	*3.34*
CNN	0.6192	0.6249	+4K
RNN	0.6259	0.6364	+4K

Table 2: Accuracy and F_1-score on the development set, and the training time (in seconds) of the methods described in section 3: Random Forest (RF), Logistic Regression (LR), naive Bayes (NB), Support Vector Machine (SVM), Ensemble of the last two (Ens.), and Convolutional and Recurrent Neural Networks (CNN and RNN, respectively). The best result in each column is in bold, the second best underlined, and the third best in italics.

The ensemble method obtained the best F_1-score and a very short training time. On the other hand, random forest model suffers from low performance (as it is usually the case in NLP) and a relatively long training time. The CNN and RNN with embedded characters as input features performed poorly, as it is usually the case in the language identification task (Jauhiainen et al., 2018). Deep methods see benefit from large amounts of data, however when being trained with fewer data,

hyperparameters play a more important role in the results, therefore further tuning them might improve the results in table 2. As of training time, the naive Bayes method was the fastest and the RNN and the CNN the slowest methods. We also experimented with one-hot encoded characters as RNN's and CNN's input features, which was not fruitful, and therefore are not included in the results.

Table 3 shows the results of the SVM and the ensemble of SVM and NB on the test set. The ensemble outperforms SVM, as on the development set.

System	F1 (macro)	Accuracy
SVM (T)	0.6660	0.6722
SVM (TD)	0.7171	0.7179
SVM + NB (TD)	**0.7210**	**0.7239**

Table 3: Results of the CLI task on the test set. T stands for training and D for development data. TD means that the model was trained on the combination of training and development data and T, only on the training data. The best result in each column is in bold.

Table 4 contains more detailed results of the best performing model on the test set, i.e. the ensemble. It shows the precision, recall, and F_1-score of the model on each class and their average. The results are ordered based on the F_1-score.

Label	Precision	Recall	F_1-score
LTB	0.8913	0.9655	0.9269
MPB	0.8109	0.8579	0.8337
OLB	0.8358	0.6924	0.7574
SUX	0.8273	0.6274	0.7136
NEA	0.5621	0.8772	0.6852
NEB	0.6775	0.5523	0.6085
STB	0.5515	0.4944	0.5214
Macro Avg.	0.7366	0.7239	0.7210

Table 4: Precision, Recall and F_1-score of all the classes and their macro average ordered from the highest to the lowest F_1-score.

Considering the results in table 4 and the confusion matrix, Late Babylonian (LTB) was the easiest class to identify with a recall of 96.55% and Middle Babylonian Peripheral (MPB) the second easiest, with a recall of 85.79% (with only 5,508 (+668) training samples). Old Babylonian (OLB) was also easy to identifiy, especially when we consider its amount training samples, 3,803 (+668). Standard Babylonian (STB) is mainly misclassified as Sumerian, and Neo-Babylonian as Standard Babylonian. Neo-Assyrian (NEA) is also among the classes with low F_1-score, but the model has achieved a very high recall, 87.72%, in this class. Neo-Assyrian (NEA) is mainly misclassified as Neo-Babylonian (NEB) and Standard Babylonian (STB).

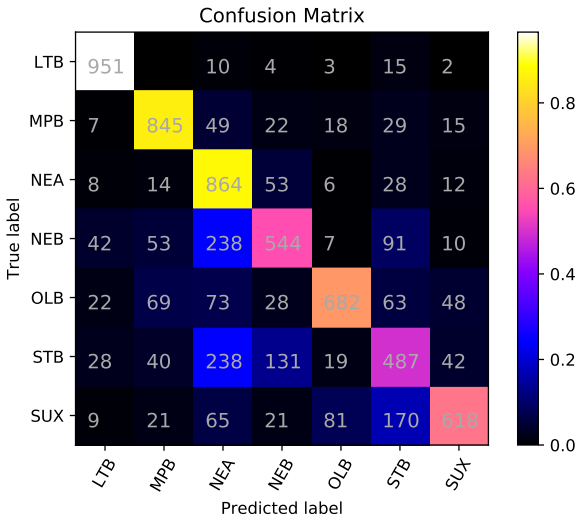

Figure 2: Confusion matrix of the ensemble model's results on the test data.

6 Conclusion

In this paper, we investigated different machine learning methods, such as SVM and neural networks, and compared their performance in the task of language and dialect identification of cuneiform texts. The best performance was achieved by a combination of SVM and naive Bayes, using only character-level features. It was shown that characters are enough to obtain at least 72.10% F_1-score. However, the best model was not able to achieve a good result classifying some of the dialects which indicates a need for other kinds of features, such as word-level ones, and/or embedded or transferred knowledge of these languages and dialects to be used in training the deep models.

References

Ahmed Ali, Najim Dehak, Patrick Cardinal, Sameer Khurana, Sree Harsha Yella, James Glass, Peter Bell, and Steve Renals. 2015. Automatic dialect detection in arabic broadcast speech. *arXiv preprint arXiv:1509.06928*.

Khaled Alrifai, Ghaida Rebdawi, and Nada Ghneim.

2017. Arabic tweeps gender and dialect prediction. In *CLEF (Working Notes)*.

Adrien Barbaresi. 2016. An unsupervised morphological criterion for discriminating similar languages. In *3rd Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial 2016)*, pages 212–220. Association for Computational Linguistics.

François Barthélemy. 1998. A morphological analyzer for akkadian verbal forms with a model of phonetic transformations. In *Proceedings of the Workshop on Computational Approaches to Semitic Languages*, pages 73–81. Association for Computational Linguistics.

François Barthélemy. 2009. The karamel system and semitic languages: structured multi-tiered morphology. In *Proceedings of the EACL 2009 Workshop on Computational Approaches to Semitic Languages*, pages 10–18. Association for Computational Linguistics.

Manas Jyoti Bora and Ritesh Kumar. 2018. Automatic word-level identification of language in assamese english hindi code-mixed data. In *4th Workshop on Indian Language Data and Resources, Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, pages 7–12.

Ralf D Brown. 2012. Finding and identifying text in 900+ languages. *Digital Investigation*, 9:S34–S43.

Dayvid W Castro, Ellen Souza, Douglas Vitório, Diego Santos, and Adriano LI Oliveira. 2017. Smoothed n-gram based models for tweet language identification: A case study of the brazilian and european portuguese national varieties. *Applied Soft Computing*, 61:1160–1172.

Jordan Cazamias, Chinmayi Dixit, and Martina Marek. 2015. Large-scale language classification.

Joseph Chee Chang and Chu-Cheng Lin. 2014. Recurrent-neural-network for language detection on twitter code-switching corpus. *arXiv preprint arXiv:1412.4314*.

Christian Chiarcos, Ilya Khait, Émilie Pagé-Perron, and Maria Sukhareva. 2017. Machine translation and automated analysis of the sumerian language. In *Joint SIGHUM Workshop on Computational Linguistics for Cultural Heritage, Social Sciences, Humanities and Literature, ACL*, pages 10–16. Association for Computational Linguistics.

Christian Chiarcos, Emilie Page-Perron, Niko Schenk, Jayanth, and Lucas Reckling. 2018. Annotating sumerian: A llod-enhanced workflow for cuneiform corpora. In *Proceedings of the Language Resources and Evaluation Conference*.

Simon Clematide and Peter Makarov. 2017. Cluzh at vardial gdi 2017: Testing a variety of machine learning tools for the classification of swiss german dialects. In *Proceedings of the Fourth Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial)*, pages 170–177.

Bernardt Duvenhage, Mfundo Ntini, and Phala Ramonyai. 2017. Improved text language identification for the south african languages. In *2017 Pattern Recognition Association of South Africa and Robotics and Mechatronics (PRASA-RobMech)*, pages 214–218. IEEE.

Mahmoud El Haj, Paul Edward Rayson, and Mariam Aboelezz. 2018. Arabic dialect identification in the context of bivalency and code-switching.

Marc Franco-Salvador, Greg Kondrak, and Paolo Rosso. 2017. Bridging the native language and language variety identification tasks. *Procedia computer science*, 112:1554–1561.

Helena Gomez, Ilia Markov, Jorge Baptista, Grigori Sidorov, and David Pinto. 2017. Discriminating between similar languages using a combination of typed and untyped character n-grams and words. In *Proceedings of the Fourth Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial)*, pages 137–145.

Cyril Goutte, Serge Léger, Shervin Malmasi, and Marcos Zampieri. 2016. Discriminating similar languages: Evaluations and explorations. *arXiv preprint arXiv:1610.00031*.

Abualsoud Hanani, Aziz Qaroush, and Stephen Taylor. 2017. Identifying dialects with textual and acoustic cues. In *Proceedings of the Fourth Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial)*, pages 93–101.

Andrea Hategan, Bogdan Barliga, and Ioan Tabus. 2009. Language identification of individualwords in a multilingual automatic speech recognition system. In *2009 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 4357–4360. IEEE.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Timo Homburg and Christian Chiarcos. 2016. Akkadian word segmentation. In *Proceedings Tenth International Conference on Language Resource Evaluation.(LREC 2016)*, pages 4067–4074.

Tommi Jauhiainen, Heidi Jauhiainen, Tero Alstola, and Krister Lindén. 2019. Language and Dialect Identification of Cuneiform Texts. In *Proceedings of the Sixth Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial)*.

Tommi Jauhiainen, Krister Lindén, and Heidi Jauhiainen. 2017. Evaluation of language identification methods using 285 languages. In *Proceedings of the 21st Nordic Conference on Computational Linguistics*, pages 183–191.

Tommi Jauhiainen, Marco Lui, Marcos Zampieri, Timothy Baldwin, and Krister Lindén. 2018. Automatic language identification in texts: A survey. *arXiv preprint arXiv:1804.08186*.

David Jurgens, Yulia Tsvetkov, and Dan Jurafsky. 2017. Incorporating dialectal variability for socially equitable language identification. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, volume 2, pages 51–57.

Laura Kataja and Kimmo Koskenniemi. 1988. Finite-state description of semitic morphology: A case study of ancient accadian. In *Coling Budapest 1988 Volume 1: International Conference on Computational Linguistics*, volume 1.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Tom Kocmi and Ondřej Bojar. 2017. Lanidenn: Multilingual language identification on character window. *arXiv preprint arXiv:1701.03338*.

Chris van der Lee and Antal van den Bosch. 2017. Exploring lexical and syntactic features for language variety identification. In *Proceedings of the Fourth Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial)*, pages 190–199.

Liang Luo, Yudong Liu, James Hearne, and Clinton Burkhart. 2015. Unsupervised sumerian personal name recognition. In *The Twenty-Eighth International Flairs Conference*.

Aaron Macks. 2002. Parsing akkadian verbs with prolog. In *Proceedings of the ACL-02 workshop on Computational approaches to semitic languages*.

Shervin Malmasi and Marcos Zampieri. 2016. Arabic dialect identification in speech transcripts. In *Proceedings of the Third Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial3)*, pages 106–113.

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 the Third Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial3)*, pages 1–14.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Édouard Duchesnay. 2011. Scikit-learn: Machine learning in python. *J. Mach. Learn. Res.*, 12:2825–2830.

Francisco Rangel, Marc Franco-Salvador, and Paolo Rosso. 2016. A low dimensionality representation for language variety identification. In *International Conference on Intelligent Text Processing and Computational Linguistics*, pages 156–169. Springer.

Younes Samih and Laura Kallmeyer. 2017. *Dialectal Arabic processing Using Deep Learning*. Ph.D. thesis, Ph. D. thesis, Düsseldorf, Germany.

Anil Kumar Singh and Jagadeesh Gorla. 2007. Identification of languages and encodings in a multilingual document. In *Building and Exploring Web Corpora (WAC3-2007): Proceedings of the 3rd Web as Corpus Workshop, Incorporating Cleaneval*, volume 4, page 95. Presses univ. de Louvain.

Valentin Tablan, Wim Peters, Diana Maynard, Hamish Cunningham, and K Bontcheva. 2006. Creating tools for morphological analysis of sumerian. In *LREC*, pages 1762–1765.

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.

Kenji Yamauchi, Hajime Yamamoto, and Wakaha Mori. 2018. Building a handwritten cuneiform character imageset. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC-2018)*.

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 the Fourth Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial)*, Valencia, Spain.

Marcos Zampieri, Shervin Malmasi, Preslav Nakov, Ahmed Ali, Suwon Shuon, 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 the Fifth Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial)*, Santa Fe, USA.

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 the Sixth Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial)*. Association for Computational Linguistics.