Character Level Convolutional Neural Network for German Dialect Identification

Mohamed Ali
Cairo University, Egypt
mohamedali@aucegypt.edu

Abstract

This paper presents the systems submitted by the safina team to the German Dialect Identification (GDI) shared task at the VarDial Evaluation Campaign 2018. The GDI shared task included four German dialects: Basel, Bern, Lucerne and Zurich in addition to a fifth "surprise dialect" for which no training data is available. The proposed approach is to use character-level convolution neural network to distinguish the four dialects. We submitted three models with the same architecture except for the first layer. The first system uses one-hot character representation as input to the convolution layer. The second system uses an embedding layer before the convolution layer. The third system uses a recurrent layer before the convolution layer. The best results were obtained using the third model achieving 64.49% F1-score, ranked the second among eight teams.[1]

1 Introduction

German language has different national and regional variants. Standard national varieties spoken in Germany, Austria, and Switzerland co-exist with a number of dialects spoken in everyday communication. The German Dialect Identification task is concerned with identifying the specific German dialect in a written form. The German Dialect Identification was part of the VarDial Evaluation Campaign 2017 and it attracted many researchers, since 10 teams have participated in that task (Zampieri et al., 2017). That task included Swiss German dialects from four areas: Basel, Bern, Lucerne and Zurich and the goal was to train a model to detect the dialect using speech transcript.

In this paper we present the safina team's submissions for the 2018 GDI shared task which was organized as a part of Vardial Evaluation Campaign 2018 (Zampieri et al., 2018). In this year version, the organizers added a fifth "surprise dialect" for which no training data is available. The participants could take part in two sub-tracks: the four-way classification (without surprise dialect) and the five-way classification (with surprise dialect). We have participated in the four-way track only. We have used a Character-level Convolutional Neural Network approach to identify German dialects using lexical features. Our team ranked the second with F1-weighted score 64.49%.

2 Related Work

Dialect identification has two flavors: identifying dialect in spoken language and identifying dialect in written language. Research in German Dialect Identification took place in the two flavors. For spoken language, Schaeffler and Summers (1999) used prosodic features to discriminate between German dialects in spoken form.

For written language, Scherrer and Rambow (2010) used a bag-of-words approach to identify Germandialects in written-form. They were concerned with discriminating among six German dialects for

[1]The code for our submissions is available at: https://github.com/bigoooh/gdi

Proceedings of the Fifth Workshop on NLP for Similar Languages, Varieties and Dialects, pages 172–177
Santa Fe, New Mexico, USA, August 20, 2018.

six regions (known as Baseldytsch, Bärndütsch, Seislertütsch, Ostschwizertütsch, Wallisertiitsch and Züritüütsch). Hollenstein and Aepli (2015) developed a baseline for German Dialect Identification. They used character n-gram language model to build their system.

In 2017 GDI Shared Task, ten teams have participated (Zampieri et al., 2017); most of them used character n-grams for developing their models (Malmasi and Zampieri, 2017; Bestgen, 2017; Clematide and Makarov, 2017; Ionescu and Butnaru, 2017). Best result in this subtask achieved using a meta-classifier built on top of individual SVM classifiers using character n-grams (1-8) in addition to word-unigrams (Malmasi and Zampieri, 2017).

3 Methodology and Data

3.1 Character-Level Convolutional Neural Network

Convolutional Neural Networks (CNN) were invented to deal with images and they have achieved excellent results in computer vision (Krizhevsky et al., 2012; Sermanet et al., 2013; Ji et al., 2013). Later, it has been applied in Natural Language Processing (NLP) tasks and outperformed traditional models such as bag of words, n-grams and their TFIDF variants (Zhang et al., 2015). The architecture, shown in Figure 1, describes the character-level CNN model we have used in identifying the German dialects. We formulate the task as a multi-class classification problem. Given text transcript $t^{(i)}$ and the corresponding label $l^{(i)}$, we need to predict l using t. We designed a neural network classifier that takes as input the transcript as one-hot encoded array of characters (padded or truncated from the end to match a predefined maximum length). The network final output is the probability distribution over the 4 German dialects. The network layers are as follows:

- **Input Layer:** mapping each character to one-hot vector.
- Optional **Embedding or Recurrent Layer :** using embedding or GRU recurrent layer to capture the context of the character (Chung et al., 2014) .
- **Convolutional Layer:** contains multiple filter widths and feature maps which is applied to a window of characters to produce new features. Each convolution is followed by a Rectified Linear Unit (ReLU) nonlinearity and batch-normalization layers (Glorot et al., 2011; Ioffe and Szegedy, 2015).
- **Max-Pooling Layer:** apply max-over-time pooling operation over the feature map of each filter and take the maximum value as a feature for this filter (Collobert et al., 2011). The max-pooling operation is followed by a dropout layer to prevent over-fitting (Srivastava et al., 2014).
- **Softmax Layers**: represents the probability distribution over the labels.

Depending on our cross-validation results we used the following parameters for the neural network architecture:

- **Sentence maximum length:** 256 characters
- **Embedding length:**32
- **GRU layer units:**128
- **Convolution filters sizes:** from 2 to 8
- **Convolution filters feature maps:** 256 feature map for each filter
- **Dropout rate:** 0.2

In our implementation, we used Keras framework with TensorFlow as a backend (Chollet and others, 2015; Abadi et al., 2015).

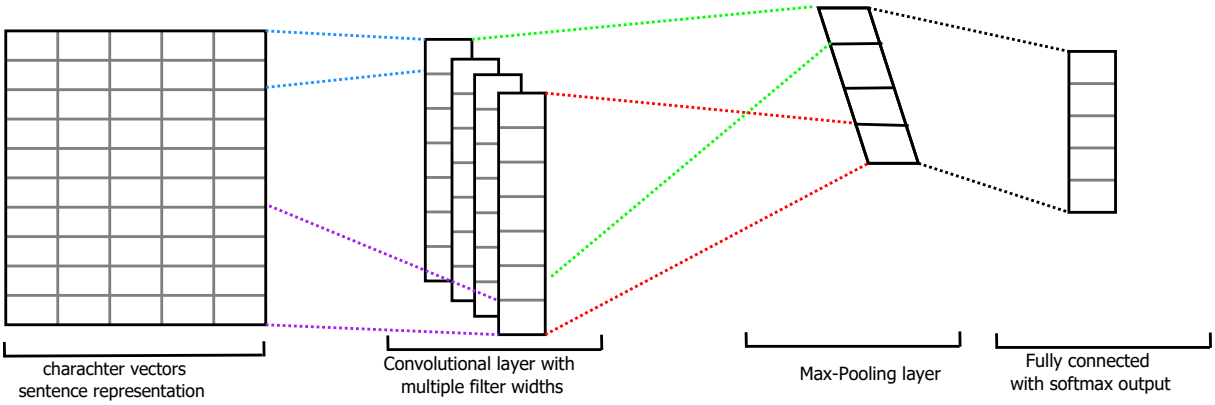

Figure 1: Character-level CNN architecture

3.2 Data

The GDI task data set contains transcriptions of video recordings collected by the ArchiMob association in the period 1999-2001 (Samardžić et al., 2016). This year's training set is an updated and expanded version of the 2017 training set. The data set contains utterances from four Swiss German dialects: Bern (BE), Basel (BS), Lucerne (LU) and Zurich (ZH). The training set contains transcripts for 14646 utterances, and the development set contains transcripts for 4658 utterances.

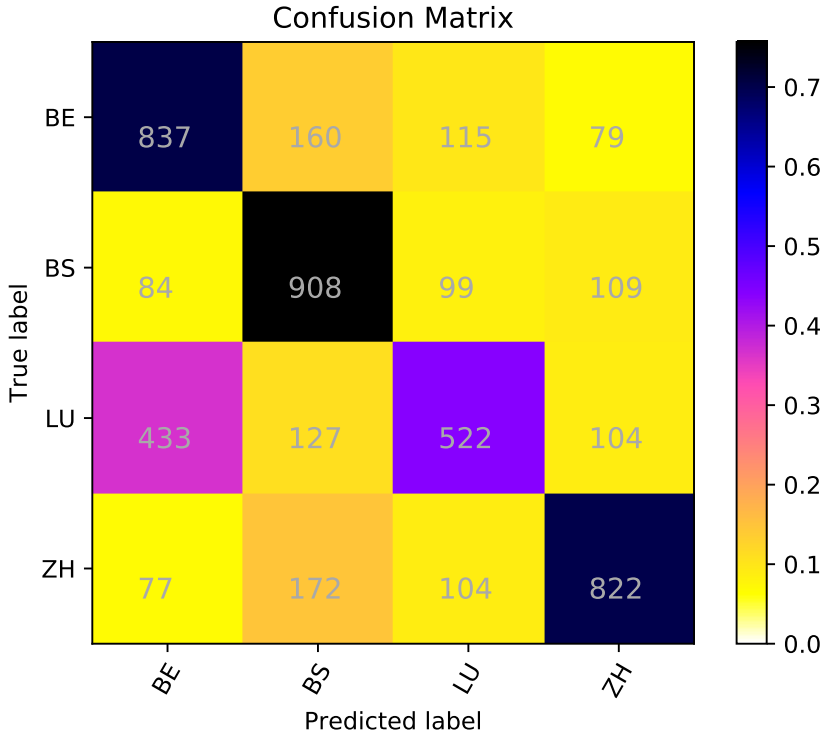

Figure 2: Confusion matrix for "CNN with a GRU recurrent layer" run

4 Results

4.1 Cross-Validation Results

We combined the training data and the validation data provided by the shared task to apply 10-fold cross validation. We tested our three different configurations in addition to a TF-IDF features based classifier, Logistic Regression classifier implemented in scikit-learn toolkit (Pedregosa et al., 2011), as a baseline. Results are shown in Table 1.

System	Accuracy
Logistic Regression using TF-IDF features	0.7714
CNN with one-hot encoded input	0.7821
CNN with an embedding layer	0.7802
CNN with a GRU recurrent layer	**0.7964**

Table 1: Cross-validation results

4.2 Test Set Results

Our three runs results are shown in Table 2. We have used the same configuration for three runs except for the input to the convolution layer. In the first run, we fed the one-hot encoded vectors for the sequence of characters directly to the convolution layer. In the second run, we fed the one-hot encoded vectors to an embedding layer before the convolution layer. In the third run, we fed the one-hot encoded vectors to a GRU recurrent layer before the convolution layer. As shown in the results, using a recurrent layer achieved better results than feeding the one-hot encoded representation directly to the convolution layer or using a regular embedding layer. However, the cost of this enhancement was huge in the training time as training the network with recurrent layers took about 5 times the period of training the network without the recurrent layer. In the GDI shared task evaluation, the submitted systems were ranked according to their F1-weighted score. Our team ranked the second with F1-weighted score 64.49%. Figure 2 shows the confusion matrix for our best run. From the matrix, we can see that the Lucerne dialect is the most confusing one; it is highly recognized as Bern dialect.

System	F1 (macro)
Random Baseline	0.2521
CNN with one-hot encoded input	0.6223
CNN with an embedding layer	0.6171
CNN with a GRU recurrent layer	**0.6449**

Table 2: Our three runs results, the best run in bold

5 Conclusion

In this work, we presented our team's three submissions for the GDI shared task. Our approach is to use Character level CNN as a feature extractor from text. Our best submission achieved by using a GRU recurrent layer as an embedding layer before the convolutional layer. However, the training of a network with a recurrent layer takes a much longer time than training a network with a regular embedding layer.

References

Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dandelion Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. 2015. TensorFlow: Large-scale machine learning on heterogeneous systems. Software available from tensorflow.org.

Yves Bestgen. 2017. Improving the character ngram model for the dsl task with bm25 weighting and less frequently used feature sets. In *Proceedings of the Fourth Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial)*, pages 115–123, Valencia, Spain, April.

François Chollet et al. 2015. Keras. https://keras.io.

Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*.

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, Valencia, Spain, April.

Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural language processing (almost) from scratch. *Journal of Machine Learning Research*, 12(Aug):2493–2537.

Xavier Glorot, Antoine Bordes, and Yoshua Bengio. 2011. Deep sparse rectifier neural networks. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, pages 315–323.

Nora Hollenstein and Noëmi Aepli. 2015. A resource for natural language processing of swiss german dialects. In *GSCL*, pages 108–109.

Sergey Ioffe and Christian Szegedy. 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*.

Radu Tudor Ionescu and Andrei Butnaru. 2017. Learning to identify arabic and german dialects using multiple kernels. In *Proceedings of the Fourth Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial)*, pages 200–209, Valencia, Spain, April.

Shuiwang Ji, Wei Xu, Ming Yang, and Kai Yu. 2013. 3d convolutional neural networks for human action recognition. *IEEE transactions on pattern analysis and machine intelligence*, 35(1):221–231.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 2012. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105.

Shervin Malmasi and Marcos Zampieri. 2017. Arabic dialect identification using ivectors and asr transcripts. In *Proceedings of the Fourth Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial)*, pages 178–183, Valencia, Spain, April.

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

Tanja Samardžić, Yves Scherrer, and Elvira Glaser. 2016. ArchiMob–A corpus of spoken Swiss German. In *Proceedings of the Language Resources and Evaluation (LREC)*, pages 4061–4066, Portoroz, Slovenia).

Felix Schaeffler and Robert Summers. 1999. Recognizing german dialects by prosodic features alone. In *Proc. ICPhS*, volume 99, pages 2311–2314.

Yves Scherrer and Owen Rambow. 2010. Word-based dialect identification with georeferenced rules. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 1151–1161. Association for Computational Linguistics.

Pierre Sermanet, David Eigen, Xiang Zhang, Michaël Mathieu, Rob Fergus, and Yann LeCun. 2013. Overfeat: Integrated recognition, localization and detection using convolutional networks. *arXiv preprint arXiv:1312.6229*.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: A simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958.

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 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 the Fifth Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial)*, Santa Fe, USA.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. In *Advances in neural information processing systems*, pages 649–657.