SC-UPB at the VarDial 2019 Evaluation Campaign: Moldavian vs. Romanian Cross-Dialect Topic Identification

Cristian Onose, Dumitru-Clementin Cercel and **Stefan Trausan-Matu**
Faculty of Automatic Control and Computers
University Politehnica of Bucharest, Romania
{onose.cristian, clementin.cercel}@gmail.com, stefan.trausan@cs.pub.ro

Abstract

This paper describes our models for the Moldavian vs. Romanian Cross-Topic Identification (MRC) evaluation campaign, part of the VarDial 2019 workshop. We focus on the three subtasks for MRC: binary classification between the Moldavian (MD) and the Romanian (RO) dialects and two cross-dialect multi-class classification between six news topics, MD to RO and RO to MD. We propose several deep learning models based on long short-term memory cells, Bidirectional Gated Recurrent Unit (BiGRU) and Hierarchical Attention Networks (HAN). We also employ three word embedding models to represent the text as a low dimensional vector. Our official submission includes two runs of the BiGRU and HAN models for each of the three subtasks. The best submitted model obtained the following macro-averaged F_1 scores: 0.708 for subtask 1, 0.481 for subtask 2 and 0.480 for the last one. Due to a read error caused by the quoting behaviour over the test file, our final submissions contained a smaller number of items than expected. More than 50% of the submission files were corrupted. Thus, we also present the results obtained with the corrected labels for which the HAN model achieves the following results: 0.930 for subtask 1, 0.590 for subtask 2 and 0.687 for the third one.

1 Introduction

The task of discriminating between two dialects or different languages is a popular research topic which has attracted a lot of interest from the research community. Specifically, the VarDial competition proposed in recent years a number of shared tasks on different languages such as dialect identification for Arabic or German, Indo-Aryan language identification, distinguish between Mainland and Taiwan Mandarin or discriminating between Dutch and Flemish (Zampieri et al., 2017, 2018). This year (Zampieri et al., 2019), the problem of discriminating between Romanian and Moldavian dialects was introduced as a series of three subtasks. It involves the processing of the MOROCO dataset (Butnaru and Ionescu, 2019) to construct several language classification models. The dataset contains text samples from online news outlets in the Romanian (RO) language or the Moldavian (MD) dialect. All the subtasks are closed, meaning that the use of external datasets is not allowed. Additionally, internal data, available for the MRC subtasks, must not be used between tasks. Thus, the first subtask is a binary classification between the two dialects. The second subtask involves a cross-dialect multi-class classification between six topics. More precisely, the classifier is trained using Moldavian dialect in order to classify samples from the Romanian dialect. The third subtask is similar to the second one, here the use of the dialects is reversed.

Generally, such tasks are approached using traditional machine learning algorithms, which unfortunately require handcrafted features. Recently, deep learning methods, where features are learned from the data, have been proposed (Ali, 2018). To address the MRC shared task, we propose the use of three state of the art deep learning architectures for text classification: Long Short-Term Memory cells (LSTM) (Hochreiter and Schmidhuber, 1997), Bidirectional Gated Recurrent Unit (BiGRU) (Graves and Schmidhuber, 2005) and Hierarchical Attention Networks (HAN) (Yang et al., 2016). The submission results are based only on BiGRU and HAN models for each of the three subtasks. After the competition deadline an error, caused by the quoting behaviour over the test file, was discovered. As a result our final submissions contained a smaller number of labels than expected, with approximately 50% of the files being corrupted. Thus, we present both the official

Proceedings of VarDial, pages 172–177

Subtask	Training	Validation	Test
1	21719	11845	5923
2	9968	5435	5923
3	11751	6410	5923

Table 1: Dataset sample distribution between training, validation and test for each of the three subtasks.

submissions as well as later work, that includes the correction of this problem.

The study of Romanian dialects was first approached by Ciobanu and Dinu (2016). They construct binary classifiers to distinguish between Romanian and three dialects (Macedo-Romanian, Megleno-Romanian and Istro-Romanian) by exploring information provided by a set of 108 word pairs. Consequently, Butnaru and Ionescu (2019) proposed a first Moldavian and Romanian Dialectal Corpus (MOROCO) assembled from multiple news websites. On top of this dataset they construct several deep learning models for dialect identification: Character level Convolutional Neural Network (CharCNN) and an improvement CNN model using squeeze and excitation blocks (Hu et al., 2018). Additionally, they also investigate shallow string kernel methods (Ionescu et al., 2016). They conclude that string kernels achieve best performance among the studied methods.

The remainder of this paper is organized as follows. In Section 2 we briefly discuss the dataset for the three tasks. Section 3 describes the methodology behind our solution, while the experimental setup and the results are presented in Section 4. Finally, Section 5 contains details regarding our conclusions.

2 Dataset

The MOROCO dataset contains Moldavian and Romanian samples collected from one of the following news categories: culture, finance, politics, science, sports and technology. It is divided between training, validation and test for each of the three tasks as described in Table 1. The test set is combined for all the subtasks such that the labels for the first task can not be inferred. This is necessary because the second and the third subtasks are based entirely on just one of the dialects.

The data samples are provided preprocessed by replacing named entities, which could act as biases for the classifiers, with a special identifier: NE. For instance, city names or important public figures from both countries, Romania or the Republic of Moldova, are anonymized.

Besides the default processing, we also took extra steps to clean up the dataset. Text usually contains expressions which carry little to no meaning, thus, we choose to remove the following: stop words, special characters and punctuation marks all except end of sentence. Additionally, we remove the named entity identifier as they interfere with the text representations. Another important aspect is given by how we deal with the diacritics. During our experiments we analyze their impact on the performance.

3 Deep learning models

In recent years, with the increasing availability of computational resources, deep neural networks became successful for classification and regression problems (LeCun et al., 2015). At first, simple feedforward networks were used. These networks lack loops or cycles and the information moves only forward, from the input to the output nodes. The switch to other types of representations, namely Recurrent Neural Networks (RNNs), was made because of the need to map input and output nodes of varying types and sizes.

Recurrent neural networks. RNNs are neural networks that form connections between nodes along a sequence. This allows the network to exhibit internal memory with respect to the inputs which in turn enables the prediction of future steps. Due to this memory, RNNs are the preferred method for processing sequential data such as time series, text or video. Unfortunately, RNNs can suffer from training instability, exploding and vanishing gradients.

Long short-term memory. Long Short-Term Memory (LSTM) units, introduced by Hochreiter and Schmidhuber (1997), are used in recurrent neural networks as a way to prevent vanishing or exploding gradients. The units allow the errors to flow backwards through endless virtual layers which are unfolded in space. Besides the usual input and output gates, LSTM units are augmented by recurrent gates called forget gates which regulate the movement of information through the cell (Gers et al., 2000).

Gated recurrent unit. Similar to LSTM, the Gated Recurrent Unit (GRU) was introduced by Cho et al. (2014) as a method to solve the van-

Name	Vector size	Min. word count	Unique tokens	Diacritics	Training algorithm
CoRoLa	300	20	250942	Yes	FastText
NLPL	100	10	2153518	No	Word2Vec Skipgram
CC	300	-	2000000	Yes	FastText

Table 2: Word embeddings: statistics regarding training methods and dataset/parameters details.

ishing gradient problem that occurs when using standard RNNs. These types of units are closely related to LSTM having similar performance and design. The GRU layers are popular due their simpler structure, which results in faster training time. A bidirectional extension for such recurrent layers was proposed by Graves and Schmidhuber (2005). It connects two hidden layers of opposite directions in a backward and forward manner to the same output. This is useful for text processing since it can encode the context present in such structures: characters and words.

Hierarchical Attention Networks. Hierarchical Attention Networks (HAN) were introduced for document classification by Yang et al. (2016). They model the hierarchical structure of documents by using two levels of attention, for words and sentences. This translates into a document representation that differentiates between the importance of the content in various parts of the text.

The model constructs a vector representation of the raw document. They follow the two-level architecture by first encoding sequences of words to embeddings using bidirectional GRU units to preserve context information. The second attention level of the model encodes sequences of vectors representing sentences received as input from the first attention mechanism. The resulting encoding, which is constructed via the two-level attention scheme, is then used for classification.

Word embeddings. Word embeddings are methods of representing text as low dimensional fixed length numerical vectors. This representation maintains semantic and syntactic relations such as synonyms, antonyms as well as context. Neural network methods for training such embeddings were first introduced by Mikolov et al. (2013).

4 Experiments and results

We aim to provide classification models for all the subtasks from the challenge. The solutions are based on word embeddings which are used as a preprocessing step to create inputs for the classifiers. In order to achieve this, we rely on a number of pretrained word vector models: Romanian Language Corpus (CoRoLa) introduced by Mititelu et al. (2018), Nordic Language Processing Laboratory (NLPL) word embedding repository (Kutuzov et al., 2017) and Common Crawl (CC) word vectors (Grave et al., 2018). The relevant details for each word vector representation model can be viewed in Table 2.

LSTM and BiGRU Models. The input for the RNN flavour models is computed by taking the mean of all word embeddings present in the text. Missing words are considered zero valued vectors. The result is a representation of the whole news item as a single embedding vector.

The LSTM architecture consists of a starting LSTM layer of size 256. This is followed by a secondary LSTM layer of 512 neurons. Next, we use dropout as a regularization technique for reducing overfitting in neural networks (Srivastava et al., 2014). The method refers to dropping out individual units during training with a probability $p = 0.3$. We use a fully connected layer consisting of 512 neurons between the recurrent layers and the output one. All LSTM layers use the tanh activation function while the fully connected one uses Rectified Linear Unit (ReLU), both empirically chosen. Finally, the output consists of a softmax activation layer of variable size depending on the subtask, 2 dimensions for the first and 6 for the second and third.

The BiGRU model is similar, it uses an initial GRU layer of 256 size followed by a bidirectional GRU layer of size 512. For both layers we apply batch normalization to accelerate the training. Similarly, we use an empirically chosen tanh activation function. This connects to two fully connected layers of 1024 and 512 neurons, both with dropout mechanism with $p = 0.3$. The output layer is the same as for the LSTM architecture.

HAN Model. Due to the two-level hierarchical attention architecture, the HAN model learns the importance of the words as a weighted sum be-

Model	Embeddings	Training F_1 Macro	Training F_1 Weighted	Training F_1 Micro	Evaluation F_1 Macro	Evaluation F_1 Weighted	Evaluation F_1 Micro	Test F_1 Macro	Test F_1 Weighted	Test F_1 Micro
BiGRU	CC	-	-	-	-	-	-	0.708	0.711	0.712
HAN	CC	-	-	-	-	-	-	0.508	0.513	0.515
LSTM	CoRoLa	0.836	0.838	0.839	0.828	0.830	0.831	0.825	0.826	0.827
LSTM	NLPL	0.804	0.806	0.806	0.796	0.797	0.797	0.798	0.799	0.799
LSTM	CC	0.858	0.858	0.858	0.854	0.855	0.855	0.847	0.848	0.848
BiGRU	CoRoLa	0.913	0.914	0.914	0.870	0.872	0.872	0.868	0.870	0.871
BiGRU	NLPL	0.871	0.872	0.873	0.835	0.837	0.838	0.834	0.836	0.838
BiGRU	CC	0.946	0.946	0.946	0.908	0.909	0.909	0.903	0.904	0.904
HAN	CC	**0.978**	**0.978**	**0.978**	**0.928**	**0.928**	**0.928**	**0.930**	**0.931**	**0.931**

Model	Embeddings	Training F_1 Macro	Training F_1 Weighted	Training F_1 Micro	Evaluation F_1 Macro	Evaluation F_1 Weighted	Evaluation F_1 Micro	Test F_1 Macro	Test F_1 Weighted	Test F_1 Micro
BiGRU	CC	-	-	-	-	-	-	0.481	0.489	0.490
HAN	CC	-	-	-	-	-	-	0.157	0.196	0.211
LSTM	CoRoLa	0.877	0.892	0.892	0.877	0.902	0.902	0.689	0.687	0.692
LSTM	NLPL	0.857	0.892	0.892	0.862	0.891	0.891	0.693	0.684	0.691
LSTM	CC	0.825	0.870	0.873	0.830	0.868	0.871	0.603	0.619	0.625
BiGRU	CoRoLa	0.922	0.941	0.941	**0.882**	**0.908**	**0.908**	0.690	0.690	0.694
BiGRU	NLPL	0.925	0.943	0.943	0.879	0.906	0.906	**0.701**	**0.692**	**0.699**
BiGRU	CC	**0.934**	0.945	0.945	**0.882**	0.903	0.903	0.649	0.652	0.658
HAN	CC	0.933	**0.959**	**0.959**	0.828	0.879	0.880	0.590	0.616	0.604

Model	Embeddings	Training F_1 Macro	Training F_1 Weighted	Training F_1 Micro	Evaluation F_1 Macro	Evaluation F_1 Weighted	Evaluation F_1 Micro	Test F_1 Macro	Test F_1 Weighted	Test F_1 Micro
BiGRU	CC	-	-	-	-	-	-	0.480	0.562	0.560
HAN	CC	-	-	-	-	-	-	0.138	0.196	0.224
LSTM	CoRoLa	0.779	0.768	0.767	0.761	0.751	0.750	0.739	0.800	0.803
LSTM	NLPL	0.775	0.762	0.763	0.764	0.751	0.751	0.787	0.834	0.834
LSTM	CC	0.743	0.740	0.742	0.738	0.734	0.735	0.790	0.843	0.844
BiGRU	CoRoLa	**0.854**	**0.840**	**0.840**	0.770	0.751	0.751	0.775	0.842	0.843
BiGRU	NLPL	0.833	0.821	0.821	0.765	0.748	0.748	**0.803**	**0.850**	**0.851**
BiGRU	CC	0.847	0.833	0.833	**0.776**	**0.757**	**0.756**	0.777	0.831	0.832
HAN	CC	0.804	0.818	0.823	0.687	0.711	0.717	0.687	0.772	0.783

Table 3: Results obtained for: subtask 1 (top), subtask 2 (middle) and subtask 3 (bottom). The best results are presented in bold. The first lines represent the official results, BiGRU represents the first run and HAN the second one. All results are grouped by model type as well as the embeddings used. We include both the results for training and evaluation datasets. Since the HAN model is computationally complex, we included only the embedding which provided the best results with the previous architectures, namely, FastText Common Crawl (CC) word vectors.

tween the word embeddings. Thus, it can create its own sentence and document models. For a consistent input, not dependent on different document and sentence sizes, the model requires two hyper parameters: maximum sentence length (number of words) and maximum document length (number of sentences). We choose these parameters by inspecting the statistics of the whole dataset to create an initial estimate which was later improved via a grid search. The best performance was achieved with a maximum sentence length of 150 words and a maximum document size of 20 sentences. Besides these parameters, we also used a grid search in order to choose a size of 200 neurons for the attention layer as well as for the BiGRU. Similarly to the previous architectures, the output consists of 2 or 6 neurons depending on the subtask.

Training configuration. We train the model using the Adam optimizer (Kingma and Ba, 2014) with the default hyper parameters. For the learning rate, we use $\alpha = 0.0005$ which was chosen using a grid search as well. We work with the training-validation split recommended by the organizers. Training is done using *tensorflow* (Abadi et al., 2016) as backend and *keras* (Chollet et al., 2015) as frontend, with a batch size of 50 for 30 epochs. During training we introduce an early stopping criterion, namely, if the cost function for the validation set does not improve for two consecutive epochs we stop.

Results. We test the proposed architecture in combination with the three presented word embeddings. The results include the official submission scores as well as the data after we corrected the input file issue. For each subtask, we present the extended results, expressed through the harmonic

	RO	MD
RO	**2517**	201
MD	207	**2998**

	CUL	FIN	POL	SCI	SPO	TEC
CUL	**152**	6	15	9	4	1
FIN	33	**562**	206	20	21	59
POL	36	67	**518**	24	29	28
SCI	10	7	6	**322**	3	13
SPO	2	12	9	0	**420**	13
TEC	19	100	20	164	27	**268**

	CUL	FIN	POL	SCI	SPO	TEC
CUL	**166**	16	15	1	13	7
FIN	7	**539**	43	0	0	16
POL	11	87	**806**	1	2	3
SCI	2	10	3	**96**	0	44
SPO	3	11	7	2	**572**	12
TEC	6	38	36	7	1	**135**

Table 4: Confusion matrices for: subtask 1 (top), subtask 2 (middle) and subtask 3 (bottom) constructed using the models which obtained the best results over the test dataset. Subtask 1 represents the classification between the Moldavian (MD) and the Romanian (RO) dialects. Subtask 2 and 3 are cross-dialect multi-class classification between: culture (CUL), finance (FIN), politics (POL), science (SCI), sports (SPO) and technology (TEC).

mean of precision and recall, F_1 score, in Table 3. Overall the HAN model outperforms the others for the first subtask and BiGRU with NLPL embeddings offers the best results for the second and third subtasks.

For the official results, the best model, BiGRU with CC embeddings, obtained macro-averaged F_1 scores as follows: 0.708 for subtask 1, 0.481 for subtask 2 and 0.480 for the third one. After the correction, the HAN with CC embedding model achieved 0.930 for subtask 1 while BiGRU with NLPL obtained 0.701 for subtask 2 and 0.803 for subtask 3. Additionally, unlike the CC embeddings, for subtasks 2 and 3 the model that obtained the best results uses embeddings without diacritics.

To better visualize and understand the misclassification behaviour we present the confusion matrices for the three subtasks in Table 4. The matrices are created using the models which achieved the best results over the test dataset. For the first subtask the error is consistent across both classes, RO and MD. Next, for the second subtask we observe high misclassification between the following classes: finance (FIN) – politics (POL), technology (TEC) – FIN and TEC – science (SCI). For the last subtask we notice that the misclassification errors from subtask 2 hold, as well as the addition of a high error between the TEC – POL classes.

5 Conclusions

In this paper we tackled the task of Moldavian vs. Romanian cross-topic identification which is part of the VarDial 2019 evaluation campaign. We proposed deep learning solutions for all three of the competition subtasks: binary classification between the two dialects and two cross-dialect six category classification from one of the dialects to the other. The proposed architectures use state of the art recurrent neural network layers as well as hierarchical attention networks. To model the languages we used the following pretrained word embeddings: Romanian Language Corpus (CoRoLa), Nordic Language Processing Laboratory (NLPL) word embedding repository and Common Crawl (CC) word vectors. We present the official competition results together with additional tests since the official submissions suffered from an input parsing issue that corrupted 50% of the results. All extra tests are evaluated with the official script. The new results confirm the superior classification performance of the HAN model with CC embeddings for subtask 1 and BiGRU with NLPL embeddings for the other subtasks.

Acknowledgments

This work was supported by the 2008-212578 LT-fLL FP7 project.

References

Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. 2016. Tensorflow: a system for large-scale machine learning. In *OSDI*, volume 16, pages 265–283.

Mohamed Ali. 2018. Character level convolutional neural network for arabic dialect identification. In *Proceedings of the Fifth Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial 2018)*, pages 122–127.

Andrei M. Butnaru and Radu Tudor Ionescu. 2019. MOROCO: The Moldavian and Romanian Dialectal Corpus. *ArXiv*.

Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using RNN encoder-decoder

for statistical machine translation. *arXiv preprint arXiv:1406.1078*.

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

Alina Maria Ciobanu and Liviu P Dinu. 2016. A computational perspective on the romanian dialects. In *LREC*.

Felix A. Gers, Jürgen A. Schmidhuber, and Fred A. Cummins. 2000. Learning to forget: Continual prediction with LSTM. *Neural Comput.*, 12(10):2451–2471.

Edouard Grave, Piotr Bojanowski, Prakhar Gupta, Armand Joulin, and Tomas Mikolov. 2018. Learning word vectors for 157 languages. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC 2018)*.

Alex Graves and Jürgen Schmidhuber. 2005. Framewise phoneme classification with bidirectional LSTM and other neural network architectures. *Neural Networks*, 18(5-6):602–610.

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

Jie Hu, Li Shen, and Gang Sun. 2018. Squeeze-and-excitation networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7132–7141.

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.

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

Andrei Kutuzov, Murhaf Fares, Stephan Oepen, and Erik Velldal. 2017. Word vectors, reuse, and replicability: Towards a community repository of large-text resources. In *Proceedings of the 58th Conference on Simulation and Modelling*, pages 271–276. Linköping University Electronic Press.

Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. 2015. Deep learning. *nature*, 521(7553):436.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Verginica Barbu Mititelu, Dan Tufis, and Elena Irimia. 2018. The reference corpus of the contemporary romanian language (CoRoLa). In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC-2018)*.

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.

Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. 2016. Hierarchical attention networks for document classification. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1480–1489.

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)*, pages 1–15, Valencia, Spain. Association for Computational Linguistics.

Marcos Zampieri, Shervin Malmasi, Preslav Nakov, Ahmed Ali, Suwon Shon, James Glass, Yves Scherrer, Tanja Samardžić, Nikola Ljubešić, Jörg Tiedemann, et al. 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*. Association for Computational Linguistics.

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.