The R2I_LIS Team Proposes Majority Vote for VarDial's MRC Task

Adrian-Gabriel Chifu
Aix-Marseille Université
Université de Toulon
CNRS, LIS, Marseille, France
adrian.chifu@univ-amu.fr

Abstract

This article presents the model that generated the runs submitted by the R2I_LIS team to the VarDial2019 evaluation campaign, more particularly, to the binary classification by dialect sub-task of the Moldavian vs. Romanian Cross-dialect Topic identification (MRC) task. The team proposed a majority vote-based model, between five supervised machine learning models, trained on forty manually-crafted features. One of the three submitted runs was ranked second at the binary classification sub-task, with a performance of 0.7963, in terms of macro-F1 measure. The other two runs were ranked third and fourth, respectively.

1 Introduction

The term "dialect" is used to capture two different types of linguistic phenomena: a variety of a language specific to a particular group of the language's speakers (Oxford Living Dictionaries) and a socially subordinated language with respect to a regional or national standard language, but not actually derived from the standard language (Maiden and Parry, 2006). In the case of the latter usage, the standard language it is not considered a "dialect", since it is the dominant language in state or a region.

The dynamics and the characteristics of the language variations are interesting for many research disciplines, Computer Science included. The Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial) represents a series of workshops focused on studying diatopic language variations from a computational perspective. The first workshop was in 2014, co-located with the COLING conference (Zampieri et al., 2014) and VarDial2019 is co-located with the NAACL2019 conference (Zampieri et al., 2019).

Since 2017, evaluation campaigns are proposed for the VarDial workshops. Four or five tasks are proposed every year. One of the VarDial2019 evaluation campaign (Zampieri et al., 2019) tasks is the Moldavian vs. Romanian Cross-dialect Topic identification (MRC) closed training shared task.

The proposed approach tackles the first sub-task of the MRC task (binary classification between dialects). We show how 40 simple features can be effective for a simple supervised machine learning architecture. The features are fed to five learning models and a majority vote between the decisions of the five classifiers is charged with the final decision.

The motivation behind this approach is to prove that simple features, thus faster to compute, are effective for the discrimination task. Another point is that the majority vote helps improving the performance and also stabilizes the model, making it more robust, with respect to various train data splits.

The rest of the article is structured as follows. Section 2 positions the article with respect to VarDial evaluation campaigns and presents the related work that provided the data set for the evaluation task. Section 3 describes the method, while Section 4 presents the implementation details and the experimental framework. In Section 5 the results are presented and discussed. Section 6 concludes the paper.

2 Related Work

Our research fits in the context of the VarDial evaluation campaigns. As for other evaluation campaigns, the tasks evolve from one edition to another. While some tasks are recurrent, others are not re-conducted, leaving place for new ones.

The 2017 campaign (Zampieri et al., 2017) had four tasks: Arabic Dialect Identification (ADI),

Proceedings of VarDial, pages 138–143

Cross-lingual Dependency Parcing (CLP), Discriminating between Similar Languages (DSL) and German Dialect Identification (GDI).

In 2018, the evaluation campaign (Zampieri et al., 2018) had five tasks, the continuation of the ADI and GDI tasks and the Morphosyntactic Tagging of Tweets (MTT), the Discriminating between Dutch and Flemish in Subtitles (DFS) and the Indo-Aryan Language Identification (ILI).

The latest evaluation campaign, VarDial2019 has also five shared tasks: the continuation of the German Dialect Identification (GDI) task, the Cross-lingual Morphological Analysis (CMA) task, the Discriminating between Mainland and Taiwan variation of Mandarin Chinese (DMT) task, the Moldavian vs. Romanian Cross-dialect Topic identification (MRC) task and the Cuneiform Language Identification (CLI) task.

We participated at the MRC task, more specifically at the sub-task that focuses on the discrimination between Romanian and Moldavian news texts. To the best of our knowledge, there is no related work for these specific dialects, except for the MOROCO data set paper (Butnaru and Ionescu, 2019), in which the authors describe the collected corpus and present empirical studies on several classification tasks. Some experiments using a shallow string kernels-based approach and a deep approach, based on character-level CNNs with Squeeze-and-Excitation blocks are conducted. The authors also present and analyze the impact of the named entities. In the final data set, the named entities are removed.

3 Method

The proposed method is based on forty manually-crafted features that are fed to five supervised machine learning models for binary classification. The final output represents a majority vote that decides whether a text is written in Romanian or in Moldavian (”RO/MD?”). The architecture of the model is presented in Figure 1.

3.1 Features

The features we considered for this method are handcrafted and meant to be straightforward, simple to understand, thus easy and fast to compute. The forty features are of five types: token statistics-based, character-based, punctuation-based, word-based and named entity-based features, respectively. We believe that frequencies of some characters, or words may be discriminant for the classification between the two dialects.

3.1.1 Token Statistics Features

In order to compute these features, the text was pre-processed. Sentences have been extracted, based on the punctuation. In order to obtain tokens, the punctuation was removed and the remaining text was split by spaces. All tokens are transformed to lowercase.

For a given text, we considered three such features: the average number of tokens per sentence, the total number of tokens and the average number of characters of a token.

3.1.2 Character Features

The character-based features consider the text at the character level. A character feature represents the number of occurrences of the character in the text. We took into account fifteen such features. The considered characters are: all the vowels in Romanian (a, e, i, o u, ă, â and î) and some Romanian consonants (b, c, d, m, n and ş and ţ). Except for the ”ş” and ”ţ”, which are specific for the Romanian language, the choice of the other consonants was completely empirical.

Regarding the character ”î”, we have considered an extra feature that represents the number of occurrences of this character inside a token (not at the beginning). For instance, the character occurrence in the word ”dînsa” was counted, while the occurrence in the word ”început” was not. This is very specific to the Moldavian dialect, since in the Romanian dialect, the character ”î” does not generally appear (there are a few exceptions) in the interior of words, being replaced by the character ”â”.

3.1.3 Punctuation Features

The punctuation features concern some punctuation signs (space was also considered here). The number of occurrences of a punctuation sign represents a the feature value. Five such features were considered: space, dot, double quotes, exclamation points and question marks.

3.1.4 Word Features

The word features take into account some words that we considered as potentially discriminant, such as prepositions, or dialect/regional words. The number of occurrences of the selected words represent the feature values. There are fifteen such features: ”ci”, ”mai”, ”cu”, ”care”, ”la”, ”în”, ”o”

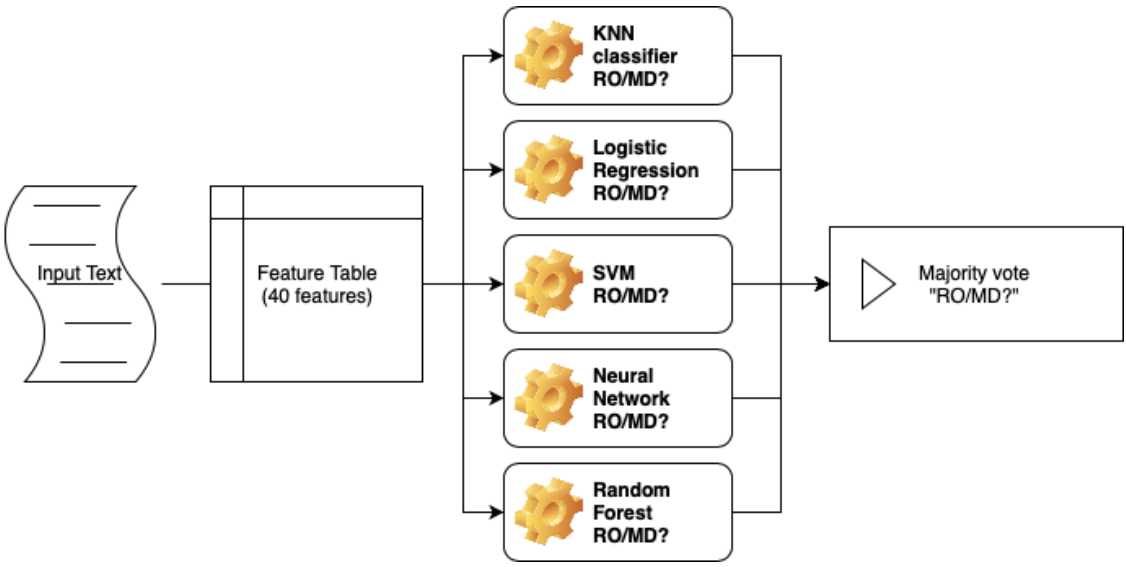

Figure 1: The architecture of the majority vote-based proposed model.

(the single character between two spaces), ”un”, ”de”, ”pe”, ”şi”, ”dînsa”, ”dînsul”, ”dînşii” and ”dînsele”.

3.1.5 Named Entity Feature

In the data set for the MRC task (Butnaru and Ionescu, 2019), the name entities are identified and replaced by ”NE”. We decided to take this information into account, thus the number of occurrences of ”NE” represent the feature value for our named entity-based feature.

3.2 Models

The features are fed to five supervised machine learning models. We have chosen the following models:

- a KNN classifier (called KNN here);
- a Logistic Regression classifier (called LR here);
- a SVM Classifier (called SVM here);
- a Neural Network classifier (called NN here);
- a Random Forest classifier (called RF here).

The hyperparameters of each model are presented later, in Section 4.2.

3.3 Majority Vote

The five models output their respective classification decisions. The final decision is made by a simple majority vote between the five aforementioned decisions. Having an odd number of votes will not yield any ex aequo final decisions. For instance, if three of the models decided in favor of Romanian and two models decided in favor of Moldavian, the final decision is Romanian.

4 Experiments

In this section we present the data set, the model parameters, as well as the submitted runs with their respective particularities.

4.1 Data Set

For the VarDial2019’s (Zampieri et al., 2019) MRC task, the MOROCO data set (Butnaru and Ionescu, 2019) is proposed. We focus on the data provided for the binary classification sub-task, that is to say the first sub-task, in which a classification model is required to discriminate between the Moldavian and the Romanian dialects.

The data set contains Moldavian (MD) and Romanian (RO) samples of text collected from the news domain. The training set (called ”train”) contains 21719, the development set (called ”dev”) contains 11845 samples and the test set (called ”test”) contains 5923 samples. A summary of the data set, containing the class distribution is presented in Table 1.

Since the training type for this task is a closed one, only subsets of provided train data have been used, without any external resources.

4.1.1 Environment

The proposed architecture was implemented in python (version 3.7.2) and the five supervised

# samples	Total	RO	MD
train	21719	11751	9968
dev	11845	6410	5435
test	5923	3205	2718

Table 1: MRC task data set summary.

machine learning models were implemented with the sklearn library (version 0.20.2). The implemented features have been scaled with a model from sklearn (StandardScaler), based on training features and then applied to both training and testing feature sets.

4.2 Model Parameters

We describe here the sklearn hyperparameter settings for each of the five supervised machine learning models used in the proposed architecture. We mention that the hyperparameter settings were chosen empirically. Most of the models are standard models with slim modifications. A more robust cross-validation is left as a perspective for the future work.

KNN. Besides the default configuration, the number of neighbors was set to five.

LR. Besides the default configuration, the random state was set to zero, the solver was "newton-cg", the maximum number of iterations was set to one thousand and the multi-class parameter was set to "auto".

SVM. Besides the default configuration, the gamma parameter was set to "scale".

NN. Besides the default configuration, the solver was set to "adam", the activation function was set to "tanh", the maximum number of iterations was set to one thousand, the alpha was set to $1e-5$, the size of the hidden layer was set to one hundrend, the random state was set to one and the warm start was set as "True".

RF. Besides the default configuration, the number of estimators was set to three hundred, the maximum depth was set to two and the random state was set to zero.

4.3 Runs

The MRC task allows three runs per sub-task. We describe below the particularities of the three runs that we submitted to the first sub-task.

Run1. For this run, the forty features are computed as described in Section 3.1 and the training data was represented by the "train" and the "dev" texts, concatenated.

Run2. For this run, the forty features are computed as described in Section 3.1 and the training data was represented only by the "train" texts.

Run3. For this run, the forty features are computed as described in Section 3.1, with one modification: the character features, the punctuation features and the word features were normalized by dividing them by the total number of characters in the corresponding text. The training data was represented by the "train" and the "dev" texts, concatenated.

5 Results and Discussion

We present here the F1-score results and the confusion matrices of the submitted runs. We discuss the results both with respect with train and test data. Finally, we present the relative performance of our runs that were ranked second, third and fourth, with respect to the other participants to the first sub-task of the MRC task.

5.1 F1-scores

The macro-averaged F1-score was the ranking criterion for the MRC task. The values obtained by the three submitted runs are presented in Table 2. We present the performance both on train and on test and with respect to the five models of the architecture. The final decision majority vote (called "Majority" here) performances are displayed on the last line of the table. One can clearly notice that the best performing run, Run3, obtains the best performance in the case of the most models (except for RF and test data of NN). Run1 only gets the best performance on test data for the NN model. Run2 has the best performance for the RF model. However, overall, Run1 has a slightly better performance (Majority on test: 0.7781) than Run2 (Majority on test: 0.7762).

Run3 has the normalized features. Thus, as expected, the normalized features are performing better than the unnormalized features.

Run1 and Run3 are trained on the concatenated "train" and "dev" texts. Thus, as expected, the best performances are achieved when training on the most data possible.

Since there are not many differences in terms of performance between train and test, we may hypothesise that overfitting is not present. The only exception is for the NN models, for the three runs, where the absolute difference between train and

Runs	Run1		Run2		Run3	
Model/Split	**train**	**test**	**train**	**test**	**train**	**test**
KNN	0.8367	0.7551	0.8318	0.7467	**0.8476**	**0.7732**
LR	0.7248	0.7204	0.7272	0.7218	**0.7327**	**0.7315**
SVM	0.7884	0.7765	0.7857	0.7718	**0.8305**	**0.8039**
NN	0.8502	**0.7889**	0.8727	0.7646	**0.8933**	0.7821
RF	0.6896	0.6973	**0.7078**	**0.7151**	0.6994	0.7049
Majority	0.8092	0.7781	0.8117	0.7762	**0.8379**	**0.7964**

Table 2: The macro-averaged F1 score for the three runs, by split (train/test) and by model. The best values per line, for train and test, respectively, are displayed in bold.

Model/Split			train		test	
			predicted		predicted	
			MD	***RO***	***MD***	***RO***
KNN	**true**	***MD***	12861	2542	2072	646
		RO	2537	15624	690	2515
LR	**true**	***MD***	10070	5333	1805	913
		RO	3467	14694	652	2553
SVM	**true**	***MD***	11717	3686	1991	727
		RO	1889	16262	413	2792
NN	**true**	***MD***	13113	2290	2016	702
		RO	1247	16914	575	2630
RF	**true**	***MD***	7738	7665	1383	1335
		RO	1751	16410	297	2908
Majority	**true**	***MD***	11570	3833	1928	790
		RO	1484	16677	389	2816

Table 3: Confusion matrices for Run3, by method and by split (train/test).

test performance is of about 0.1.

5.2 Confusion Matrices

To focus on the best submitted run, we present the confusion matrices for Run3 in Table 3. In this table, we present the detailed confusion matrices for each of the five models, as well as for the Majority, both for train and test.

One can notice that the most balanced in terms of false positives is KNN, while the most unbalanced seems to be RF. Even though the examples in the data set are quite balanced with respect to the number of samples per class, our model has a tendency to predict much more texts for the label "RO". For instance, the false positives for the "RO" class are more than twice as many as for the "MD" class. This occurs for Majority, both for train and test.

In Figure 2 we display the confusion matrices for the test data, for the three runs. One can notice that Run2 has the most false positives for the class "MD". On the other hand, Run1 has the most false positives for the class "RO".

5.3 Other Participants

The ranking of all participants at the first sub-task of MRC are presented in Table 4. One can notice that the runner-up, our Run3 is at a significant distance from the winner (about 0.1 in absolute difference between the macro-averaged F1-scores).

6 Conclusion

We presented here the approach that generated the three runs we submitted at VarDial's MRC task, most specifically at the first sub-task that aims to discriminate news texts written in Moldavian and Romanian dialects. The architecture is based on forty features and a majority vote between five supervised machine learning models. The submitted runs ranked second, third and fourth, respectively. We thus showed that a simple architecture, based on features simple to compute can still be effective and competitive.

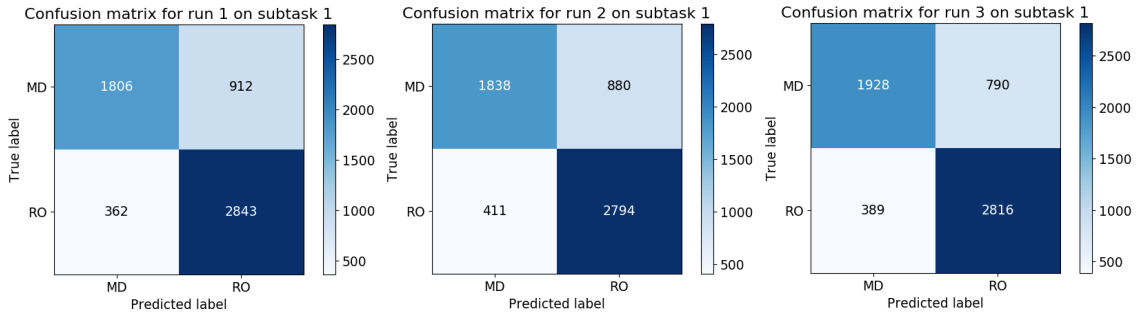

Figure 2: The confusion matrices for the three submitted runs.

Rank	Team	Run	Macro-F1	Weighted-F1	Micro-F1 (accuracy)
1	DTeam	1	0.8950	0.8960	0.8965
2	**R2I_LIS (Run3)**	3	**0.7964**	**0.7989**	**0.8009**
3	**R2I_LIS (Run1)**	1	**0.7781**	**0.7813**	**0.7849**
4	**R2I_LIS (Run2)**	2	**0.7762**	**0.7792**	**0.7820**
5	tearsofjoy	1	0.7573	0.7592	0.7596
6	lonewolf	2	0.7354	0.7332	0.7381
7	SC-UPB	1	0.7088	0.7114	0.7121
8	lonewolf	1	0.6560	0.6646	0.6877
9	lonewolf	3	0.6077	0.5997	0.6319
10	SC-UPB	2	0.5081	0.5131	0.5156

Table 4: The ranking of all participants at the MRC's first sub-task. The runs from this paper are shown in bold.

As future work, we plan to set up a more rigorous cross-validation protocol for the hyperparameter setup, in order to obtain more robust models. Another lead is to apply a feature selection method in order to identify the most useful features.

References

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

Martin Maiden and Mair Parry. 2006. *The dialects of Italy*. Routledge.

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, 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.

Marcos Zampieri, Liling Tan, Nikola Ljubešić, and Jörg Tiedemann. 2014. A report on the dsl shared task 2014. In *Proceedings of the First Workshop on Applying NLP Tools to Similar Languages, Varieties and Dialects*, pages 58–67, Dublin, Ireland. Association for Computational Linguistics and Dublin City University.