NAIVE BAYES-BASED EXPERIMENTS IN ROMANIAN DIALECT IDENTIFICATION

Tommi Jauhiainen
Department of Digital Humanities, UH
Department of English, RIT

Heidi Jauhiainen
Department of Digital Humanities, UH

Krister Lindén
Department of Digital Humanities, UH

HELSINGIN YLIOPISTO
HELSINGFORS UNIVERSITET
UNIVERSITY OF HELSINKI
HUMANISTINEN TIEDEKUNTA
HUMANISTISKA FAKULTETEN
FACULTY OF ARTS

FIN-CLARIN

INTRODUCTION

This poster describes the experiments and systems developed by the SUKI team for the second edition of the Romanian Dialect Identification (RDI) shared task which was organized as part of the 2021 VarDial Evaluation Campaign. We submitted two runs to the shared task and our second submission was the overall best submission by a noticeable margin.

RDI SHARED TASK

RDI 2021 was the third shared task focusing on discriminating between the Moldavian and Romanian dialects of the Romanian language (ron). It appeared for the first time as one of the tracks of the Moldavian vs. Romanian Cross-dialect Topic identification shared task (MRC) in 2019 and as a separate task in 2020.

The participants were provided with two sets of data, one for training and one for development. An augmented MOROCO data set was used as the training data (39,487 samples). The development data provided were tweets which in the RDI 2020 were used as the target development data (215 tweets) and the test data (5,022 tweets).

The participants were informed that the test set would also consist of tweets making the development set in-domain and the training set out-of-domain in relation to the test set. The task was closed, so only the data provided by the organizers was to be used in preparation of the participating systems.

The test set was provided in the beginning of the general evaluation period of the VarDial Evaluation Campaign. The exact division of the samples between the two dialects in the test set was not a priori known to the participants. As in RDI 2020, the evaluation measure used was the macro F1 score. It gives both dialects equal value despite the possibility of there being an unequal number of samples in each dialect.

EXPERIMENTS

As the task was very similar to the RDI 2020 task, we continued our experiments from the conclusions we had arrived in 2020. We had experimented with three types of generative classifiers using character *n*-grams: the simple scoring, the sum of relative frequencies and the product of relative frequencies. As the results with the two former were clearly inferior to the product of relative frequencies, we begun our experiments using this naive Bayesian classifier.

From the beginning, we suspected that using the in-domain development set would most probably be far more beneficial than using the out-of-domain training set. The language identification methods we were using have a number of parameters which are set using a development set. In light of this, we divided the development set in two parts. We used the first 1,306 lines of Moldavian and the first 1,313 lines of Romanian for the *dev-dev* and the rest was used for *dev-test*. Table below lists the results of experiments using different combinations of the training and the development sets with and without blacklists and language model adaptation. We started with a classifier which modified the training and the test data so that all non-alphabetic characters were removed and the remaining alphabetic characters lowercased, but later noticed that this was not beneficial.

SUBMISSIONS

The results for the first run were produced by using a custom coded language identifier using the product of relative frequencies of character *n*-grams. Basically it is a **naive Bayes classifier using the relative frequencies as probabilities**. The lengths of the character *n*-grams used were from 2 to 5. Only the development data was used as training material in this run. All characters were used and not lowercased. In addition to the basic classifier, we used a **blacklist** of lowercase character *n*-grams generated from the training and the development data. We incorporated the blacklists as a preprocessing step on the product of relative frequencies classifier so that the blacklists would judge the mystery text if a blacklisted *n*-gram would be found from the mystery text. We first populated the blacklist for dialect 1 with those *n*-grams which were found from the *dev-dev* of dialect 2, but not of the dialect 1. Then, we pruned the list so, that only the *n*-grams found in the training data of dialect 2 but not in dialect 1 were kept.

Training data	Parameters optimized for	Method	Macro F_1 for dev-test
train	dev-dev	NB (lowercased alph. characters)	0.6678
train + dev-dev	dev-test	NB (lowercased alph. characters)	0.7203
dev-dev	dev-dev	NB (lowercased alph. characters)	0.7889
dev-dev	dev-test	NB (lowercased alph. characters)	0.8072
dev-dev	dev-test	NB (all characters)	0.8380
dev-dev	dev-test	NB + blacklist (all characters)	0.8411
dev-dev	dev-test	NB + adaptation (all characters)	0.8186

The results for the second run were also produced by a language identifier using the product of relative frequencies of character *n*-grams. In addition, we used a similar **language model adaptation technique** as we used when winning the GDI and ILI shared tasks in 2018.

In the adaptive version, the classifier keeps a separate record of the lines of the mystery text which have been finally identified. First, every line of the mystery text is identified and the resulting identifications are stored in a temporary table together with the probability differences between the best and the second best language. A larger probability difference corresponds to the identifier having a greater confidence in the identification and are thus called confidence scores. A fixed size fraction of the temporarily identified mystery lines with the highest confidence scores are then processed and their information added to the corresponding language models as well as their identification recorded as finally identified. Then all those lines not yet finally identified are re-identified with the newly adapted language models. This is repeated until all the fractions are processed.

We did not use a blacklist for the second run as combining the blacklist classifier with adaptive language models did not seem straightforward, so we left it for future work.

CONCLUSIONS

The results would seem to indicate that the tweets in the actual test data were clearly more out-of-domain when compared with the development data than our *dev-dev* set was from the *dev-test* set. The adaptive version, which gave lower scores between *dev-dev* and *dev-test* than the blacklist classifier was clearly superior with the actual test data.

Rank	Team	Run	Macro F_1
1	**SUKI**	**2**	**0.7772**
2	UPB	2	0.7325
3	UPB	1	0.7319
4	**SUKI**	**1**	**0.7266**
5	UPB	3	0.6743
6	Phlyers	1	0.6532
7	Phlyers	2	0.5133

ACKNOWLEGMENTS

The research presented in this article has been partly supported by The Finnish Research Impact Foundation Tandem Industry Academia funding, the Academy of Finland, and the University of Helsinki in cooperation with Lingsoft.

CONTACT INFORMATION

For each of the authors:
firstname.lastname@helsinki.fi