German Dialect Identification in Interview Transcriptions

Shervin Malmasi
Harvard Medical School, USA
Macquarie University, Australia
shervin.malmasi@mq.edu.au

Marcos Zampieri
University of Cologne
Germany
mzampie2@uni-koeln.de

Abstract

This paper presents three systems submitted to the German Dialect Identification (GDI) task at the VarDial Evaluation Campaign 2017. The task consists of training models to identify the dialect of Swiss-German speech transcripts. The dialects included in the GDI dataset are Basel, Bern, Lucerne, and Zurich. The three systems we submitted are based on: a plurality ensemble, a mean probability ensemble, and a meta-classifier trained on character and word n-grams. The best results were obtained by the meta-classifier achieving 68.1% accuracy and 66.2% F1-score, ranking first among the 10 teams which participated in the GDI shared task.

1 Introduction

German is well-known for its intrinsic dialectal variation. Standard national varieties spoken in Germany, Austria, and Switzerland co-exist with a number of dialects spoken in everyday communication. The case of Switzerland is particular representative of this situation because of the multitude and importance of dialects which are widely spoken throughout the country.

The German Dialect Identification (GDI) task, part of the VarDial Evaluation Campaign 2017 (Zampieri et al., 2017), addressed the problem of German dialectal variation by providing a dataset of transcripts from interviews with speakers of Swiss German dialects from Basel, Bern, Lucern, and Zurich recorded within the scope of the ArchiMob[1] project (Samardžić et al., 2016). The goal of the GDI task is to evaluate how well computational methods can discriminate between these four Swiss German dialects.

[1] http://archimob.ch/

In this paper we present the entries submitted by the team MAZA to the GDI task 2017. We investigate different combinations of classifiers for the task, namely: a plurality ensemble method, a mean probability ensemble method, and a meta-classifier trained on character and word n-grams.

2 Related Work

Processing dialectal data is a challenge for NLP applications. When dealing with non-standard language, systems are trained to recognize spelling and syntactic variation for further processing in applications such as Machine Translation. In the case of German, a number of studies have been published on developing NLP tools and resources for processing non-standard language (Dipper et al., 2013), dealing with spelling variation on dialectal data and carrying out spelling normalization (Samardžić et al., 2015), and improving the performance of POS taggers for dialectal data (Hollenstein and Aepli, 2014).

The identification of Swiss German dialects, the topic of the GDI shared task, has been the focus of a few recent studies. Methods for German dialect identification have proved to be particularly important for the validation of methods applied to the compilation of German dialect corpora (Scherrer and Rambow, 2010a; Scherrer and Rambow, 2010b; Hollenstein and Aepli, 2015).

The work presented here also relates to studies on the discrimination between groups of similar languages, language varieties, and dialects such as South Slavic languages (Ljubešić et al., 2007), Portuguese varieties (Zampieri and Gebre, 2012), English varieties (Lui and Cook, 2013), Romanian dialects (Ciobanu and Dinu, 2016), Chinese varieties (Xu et al., 2016), and past editions of the DSL shared task (Zampieri et al., 2014; Zampieri et al., 2015; Malmasi et al., 2016c).

Proceedings of the Fourth Workshop on NLP for Similar Languages, Varieties and Dialects, pages 164–169,
Valencia, Spain, April 3, 2017.

3 Methods and Data

3.1 Data

The GDI training/test data was extracted from the aforementioned ArchiMob corpus (Samardžić et al., 2016) which contains transcriptions of 34 interviews with native speakers of various German dialects spoken in Switzerland. The subset used for GDI contains 18 interviews (14 for training and 4 for testing) from four Swiss German dialects: Basel, Bern, Lucerne, and Zurich. No acoustic data was released with the transcriptions.

According to the information provided by the task organizers, each interview was transcribed using the 'Schwyzertütschi Dialäktschrift' writing system (Dieth, 1986). The interviews were divided into utterances and each utterance was considered to be an instance to be classified by the systems. The training set contains a total of around 14,000 instances (114,000 tokens) and the test set contains a total of 3,638 instances (29,500 tokens).

We approach the text using ensemble classifiers and a meta-classifier. In the next sections we describe the features and algorithms used in the MAZA submissions in detail.

3.2 Features

We employ two lexical surface feature types for this task, as described below.

- **Character n-grams:** This is a sub-word feature that uses the constituent characters that make up the whole text. When used as n-grams, the features are n-character slices of the text. From a linguistic point of view, the substrings captured by this feature, depending on the length, can implicitly capture various sub-lexical features including single letters, phonemes, syllables, morphemes and suffixes. In this study we examine n-grams of order 1–6.

- **Word n-grams:** The surface forms of words can be used as a feature for classification. Each unique word may be used as a feature (i.e. unigrams), but the use of bigram distributions is also common. In this scenario, the n-grams are extracted along with their frequency distributions. For this study we evaluate unigram features.

We did not pre-process[2] the data prior to feature extraction. This was not needed as the data are human-generated transcripts.

3.3 Classifier

For our base classifier we use a linear Support Vector Machine (SVM). SVMs have proven to deliver very good performance in discriminating between language varieties and in other text classification problems,[3] SVMs achieved first place in both the 2015 (Malmasi and Dras, 2015a) and 2014 (Goutte et al., 2014) editions of the DSL shared task.[4]

3.4 Ensemble Classifiers

The best performing system in the 2015 edition of the DSL challenge (Malmasi and Dras, 2015a) used SVM ensembles evidencing the adequacy of this approach for the task of discriminating between similar languages and language varieties. In light of this, we decided to test two ensemble methods. Classifier ensembles have also proven to be an efficient and robust alternative in other text classification tasks such as grammatical error detection (Xiang et al., 2015), and complex word identification (Malmasi et al., 2016a).

We follow the methodology described by Malmasi and Dras (2015a): we extract a number of different feature types and train a single linear model using each feature type. Our ensemble was created using linear Support Vector Machine classifiers. We used the seven feature types listed in Section 3.2 to create our ensemble of classifiers.

Each classifier predicts every input and also assigns a continuous output to each of the possible labels. Using this information, we created the following two ensembles.

- **System 1 - Plurality Ensemble**
  In this system each classifier votes for a single class label. The votes are tallied and the label with the highest number[5] of votes wins. Ties are broken arbitrarily. This voting method is very simple and does not have any parameters to tune. An extensive analysis of this method and its theoretical underpinnings can be found in the work of (Kuncheva, 2004, p. 112). We submitted this system as run 1.

[2] For example, case folding or tokenization.

[3] For example, Native Language Identification is often performed using SVMs (Malmasi and Dras, 2015b)

[4] See Goutte et al. (2016) for a comprehensive evaluation.

[5] This differs with a *majority* voting combiner where a label must obtain over 50% of the votes to win. However, the names are sometimes used interchangeably.

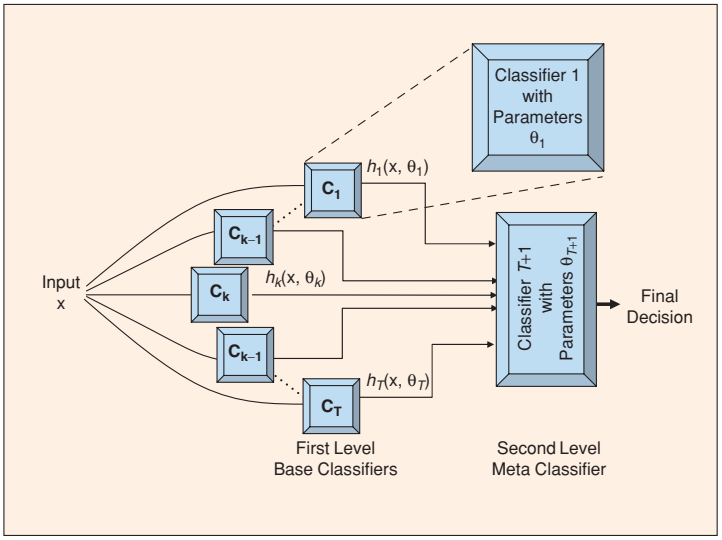

Figure 1: An illustration of a meta-classifier architecture. Image reproduced from Polikar (2006).

- **System 2 - Mean Probability Ensemble**
  The probability estimates for each class are added together and the class label with the highest average probability is the winner. An important aspect of using probability outputs in this way is that a classifier's support for the true class label is taken in to account, even when it is not the predicted label (*e.g.* it could have the second highest probability). This method has been shown to work well on a wide range of problems and, in general, it is considered to be simple, intuitive, stable (Kuncheva, 2014, p. 155) and resilient to estimation errors (Kittler et al., 1998) making it one of the most robust combiners discussed in the literature. We submitted this system as run 2.

3.5 Meta-classifier System

In addition to classifier ensembles, meta-classifier systems have proven to be very competitive for text classification tasks (Malmasi and Zampieri, 2016) and we decided to include a meta-classifier in our entry. Also referred to as classifier stacking. A meta-classifier architecture is generally composed of an ensemble of base classifiers that each make predictions for all of the input data. Their individual predictions, along with the gold labels are used to train a second-level meta-classifier that learns to predict the label for an input, given the decisions of the individual classifiers. This setup is illustrated in Figure 1. This meta-classifier attempts to learn from the collective knowledge represented by the ensemble of local classifiers.

The first step in such an architecture is to create the set of base classifiers that form the first layer. For this we used the same seven base classifiers as our ensemble.

- **System 3 - Meta-classifier**
  In this system we combined the probability outputs of our seven individual classifiers and used them to train a meta-classifier using 10-fold cross-validation. Following Malmasi et al. (2016b), we used a Random Forest as our meta-classification algorithm. We submitted this system as run 3.

4 Results

In this section we present results in two steps. First we comment on the performance obtained using each feature type and the results obtained by cross-validation on the training set. Secondly, we present the official results obtained by our system on the test set and we discuss the performance of our best method in identifying each dialect.

4.1 Cross-validation Results

We first report our cross-validation results on the training data. We began by testing individual feature types, characters n-grams (2-6) and word unigrams. Results are presented in Figure 2.

As expected we observe that character n-grams outperform word features. Character 3-grams, 4-grams, and 5-grams obtained higher results than

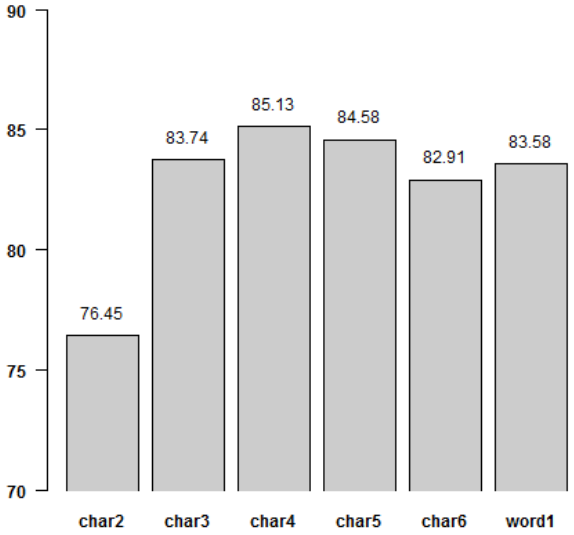

Figure 2: Cross-validation performance for each individual feature type (Y axis - Accuracy (%), X axis - Feature Type).

those obtained using word unigrams. The best results were obtained with character 4-grams achieving 85.13% accuracy. As transcriptions have been carried out using the same transcription method, character unigrams were not very informative features for the classifier achieving much lower performance than the other feature types, 52.02% accuracy. For this reason, character unigrams were not included in Figure 2

We next tested our ensemble and meta-classifier configurations on the training data. Accuracy results are shown in Table 1.

System	Accuracy
Majority Class Baseline	0.2738
Voting Ensemble (System 1)	0.8621
Probability Ensemble (System 2)	0.8674
Meta-Classifier (System 3)	**0.8725**

Table 1: Cross-validation results for the German training data.

We note that all of these methods outperform any individual feature type, with the meta-classifier achieving the best result of 87.2% accuracy and the two ensemble methods achieving comparable performance of 86.2% and 86.7% accuracy. With this information in hand we proceed to the test set evaluation.

4.2 Test Set Results

In this section we report the results of our three submissions generated from the unlabelled test data. The samples in the test set were slightly unbalanced with a majority class baseline of 25.8%.

The performance of all participants was evaluated by the shared task organizers and a more detailed description of the results is presented in the VarDial Evaluation Campaign report (Zampieri et al., 2017). Teams were ranked according to the weighted F1-score which provides a balance between precision and recall. We present the ranks with the best results for each team in Table 2.

MAZA achieved the best performance overall with 66.2% weighted F1-score. It is important to note that this rank is based on absolute scores. In the shared task report (Zampieri et al., 2017), organizers are likely to calculate ranks with statistical significance tests, which is a common practice in other shared tasks such as the DSL 2016 (Malmasi et al., 2016c) and the shared tasks from WMT (Bojar et al., 2016).

Rank	Team	F1 (weighted)
1	**MAZA**	**0.662**
2	CECL	0.661
3	CLUZH	0.653
4	qcri_mit	0.639
5	unibuckernel	0.637
6	tubasfs	0.626
7	ahaqst	0.614
8	Citius_Ixa_Imaxin	0.612
9	XAC_Bayesline	0.605
10	deepCybErNet	0.263

Table 2: GDI Closed Submission Results

Accuracy, along with macro- and micro-averaged F1-scores obtained by the three runs submitted by MAZA are presented in Table 3. We observe that the results follow the same relative pattern as the cross-validation results, with the meta-classifier achieving the best result and ranking first among the 10 teams that participated in the GDI task.

An important observation is that the test set results, for all teams, are much lower than the cross-validation results. It may have been the case that the test data was drawn from a different distribution as the training data, although this was not specified by the task organizers.

System	Accuracy	F1 (micro)	F1 (macro)	F1 (weighted)
Majority Class Baseline	0.258	—	—	—
Voting Ensemble (run1)	0.649	0.649	0.628	0.627
Probability Ensemble (run2)	0.669	0.669	0.648	0.647
Meta-classifier (run3)	**0.681**	0.681	0.663	**0.662**

Table 3: MAZA official results for the GDI task.

4.2.1 Accuracy per Dialect

Finally, we discuss the results obtained by our best method, the meta-classifier, in identifying each dialect in the test set. We present a confusion matrix with a column containing the total number of documents in each class and the performance for each dialect in Table 4.

The column 'Total' provides us an indication of the aforementioned imbalance between each dialect in the test set. The number of test instances varied from 939 instances from Basel to 877 from Zurich.

	be	bs	lu	zh	Total	Acc.
be	659	67	33	147	906	72.8%
bs	47	697	67	128	939	74.2%
lu	157	269	315	175	916	34.4%
zh	23	38	11	805	877	91.8%

Table 4: Confusion Matrix: Per Dialect Results

As expected, the four dialects are not equally difficult to be identified. The dialect from Lucern was the most difficult to be identified and the performance of the classifier was only slightly better than the 25.8% baseline.

An interesting outcome is that the dialect from Zurich, which was by far the easiest do be identified obtaining 91.8% accuracy, was also the one which generated most confusion with the other three dialects. This seems counter-intuitive on a first glance, but it might indicate that the algorithm achieves great performance for this dialect because it tries to label most of its predictions to Zurich to maximize performance. An error analysis of the misclassified instances can help understand this outcome.

5 Conclusion

In this paper we presented three systems submitted by the MAZA team to the GDI shared 2017. A meta-classifier system trained on word and character n-grams achieved 66.2% F1-score ranking first among the 10 teams that participated in the shared task. We showed that the meta-classifier outperforms two ensemble-based methods, namely plurality and mean probability, on both the training and test sets.

More than the NLP task itself, the GDI task provided participants with an interesting opportunity to study the differences between Swiss German dialects using computational methods. We observed that the dialect from Zurich is at the same time the easiest to be identified and also the one which causes the most confusion for the classifier. A linguistic analysis along with an error analysis of the misclassified instances is necessary to determine the reasons for this outcome.

Acknowledgement

We would like to thank the GDI task organizers, Noëmi Aepli and Yves Scherrer, for proposing and organizing this shared task.

References

Ondrej Bojar, Rajen Chatterjee, Christian Federmann, Yvette Graham, Barry Haddow, Matthias Huck, Antonio Jimeno Yepes, Philipp Koehn, Varvara Logacheva, Christof Monz, et al. 2016. Findings of the 2016 Conference on Machine Translation. In *Proceedings of WMT*.

Alina Maria Ciobanu and Liviu P. Dinu. 2016. A Computational Perspective on Romanian Dialects. In *Proceedings of LREC*.

Eugen Dieth. 1986. *Schwyzertütschi Dialäktschrift*. Sauerländer, Aarau, 2 edition.

Stefanie Dipper, Anke Lüdeling, and Marc Reznicek. 2013. NoSta-D: A corpus of German Non-standard Varieties. *Non-standard Data Sources in Corpus-based Research*.

Cyril Goutte, Serge Léger, and Marine Carpuat. 2014. The NRC System for Discriminating Similar Languages. In *Proceedings of the VarDial Workshop*.

Cyril Goutte, Serge Léger, Shervin Malmasi, and Marcos Zampieri. 2016. Discriminating Similar Languages: Evaluations and Explorations. In *Proceedings of LREC*.

Nora Hollenstein and Noëmi Aepli. 2014. Compilation of a Swiss German Dialect Corpus and its Application to POS Tagging. In *Proceedings of the VarDial Workshop*.

Nora Hollenstein and Noëmi Aepli. 2015. A Resource for Natural Language Processing of Swiss German Dialects. In *Proceedings of GSCL*.

Josef Kittler, Mohamad Hatef, Robert PW Duin, and Jiri Matas. 1998. On combining classifiers. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(3):226–239.

Ludmila I. Kuncheva. 2004. *Combining Pattern Classifiers: Methods and Algorithms*. John Wiley & Sons.

Ludmila I. Kuncheva. 2014. *Combining Pattern Classifiers: Methods and Algorithms*. Wiley, second edition.

Nikola Ljubešić, Nives Mikelic, and Damir Boras. 2007. Language Identification: How to Distinguish Similar Languages? In *Proceedings of ITI*.

Marco Lui and Paul Cook. 2013. Classifying English Documents by National Dialect. In *Proceedings of ALTW*.

Shervin Malmasi and Mark Dras. 2015a. Language Identification using Classifier Ensembles. In *Proceedings of the LT4VarDial Workshop*.

Shervin Malmasi and Mark Dras. 2015b. Multilingual Native Language Identification. In *Natural Language Engineering*.

Shervin Malmasi and Marcos Zampieri. 2016. Arabic Dialect Identification in Speech Transcripts. In *Proceedings of the VarDial Workshop*.

Shervin Malmasi, Mark Dras, and Marcos Zampieri. 2016a. LTG at SemEval-2016 Task 11: Complex Word Identification with Classifier Ensembles. In *Proceedings of SemEval*.

Shervin Malmasi, Marcos Zampieri, and Mark Dras. 2016b. Predicting Post Severity in Mental Health Forums. In *Proceedings of the CLPsych Workshop*.

Shervin Malmasi, Marcos Zampieri, Nikola Ljubešić, Preslav Nakov, Ahmed Ali, and Jörg Tiedemann. 2016c. Discriminating between Similar Languages and Arabic Dialect Identification: A Report on the Third DSL Shared Task. In *Proceedings of the VarDial Workshop*.

Robi Polikar. 2006. Ensemble based systems in decision making. *Circuits and Systems Magazine, IEEE*, 6(3):21–45.

Tanja Samardžić, Yves Scherrer, and Elvira Glaser. 2015. Normalising Orthographic and Dialectal Variants for the Automatic Processing of Swiss German. In *Proceedings of LTC*.

Tanja Samardžić, Yves Scherrer, and Elvira Glaser. 2016. ArchiMob–A corpus of spoken Swiss German. In *Proceedings of LREC*.

Yves Scherrer and Owen Rambow. 2010a. Natural Language Processing for the Swiss German Dialect Area. In *Proceedings of KONVENS*.

Yves Scherrer and Owen Rambow. 2010b. Word-based Dialect Identification with Georeferenced Rules. In *Proceedings of EMNLP*.

Yang Xiang, Xiaolong Wang, Wenying Han, and Qinghua Hong. 2015. Chinese Grammatical Error Diagnosis Using Ensemble Learning. In *Proceedings of the NLP-TEA Workshop*.

Fan Xu, Mingwen Wang, and Maoxi Li. 2016. Sentence-level Dialects Identification in the Greater China Region. *International Journal on Natural Language Computing (IJNLC)*, 5(6).

Marcos Zampieri and Binyam Gebrekidan Gebre. 2012. Automatic Identification of Language Varieties: The Case of Portuguese. In *Proceedings of KONVENS*.

Marcos Zampieri, Liling Tan, Nikola Ljubešić, and Jörg Tiedemann. 2014. A Report on the DSL Shared Task 2014. In *Proceedings of the VarDial Workshop*.

Marcos Zampieri, Liling Tan, Nikola Ljubešić, Jörg Tiedemann, and Preslav Nakov. 2015. Overview of the DSL Shared Task 2015. In *Proceedings of the LT4VarDial Workshop*.

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 VarDial Workshop*.