Fewer features perform well at Native Language Identification task

Taraka Rama and **Çağrı Çöltekin**
Department of Linguistics
University of Tübingen, Germany
taraka-rama.kasicheyanula@uni-tuebingen.de
ccoltekin@sfs.uni-tuebingen.de

Abstract

This paper describes our results at the NLI shared task 2017. We participated in essays, speech, and fusion task that uses text, speech, and i-vectors for the task of identifying the native language of the given input. In the essay track, a linear SVM system using word bigrams and character 7-grams performed the best. In the speech track, an LDA classifier based only on i-vectors performed better than a combination system using text features from speech transcriptions and i-vectors. In the fusion task, we experimented with systems that used combination of i-vectors with higher order n-grams features, combination of i-vectors with word unigrams, a mean probability ensemble, and a stacked ensemble system. Our finding is that word unigrams in combination with i-vectors achieve higher score than systems trained with larger number of *n*-gram features. Our best-performing systems achieved F1-scores of 87.16 %, 83.33 % and 91.75 % on the essay track, the speech track and the fusion track respectively.

1 Introduction

In this paper, we describe our (team tubafs) efforts in three different tasks during our participation in NLI shared task 2017 (Malmasi et al., 2017). All the three tasks aim at identifying native language using essays (*essay track*), speech transcriptions along with i-vectors (*speech track*) and *fusion track* that allows the participants to use all the three data sources to design and test a system for the purpose of NLI.

The first NLI task employed only essays written in English for the identification of native language. To date, all NLI shared tasks have been based on L2 English data, but NLI research has been extended to at least six other non-English languages (Malmasi and Dras, 2015). In addition to using the written responses, a recent trend has been the use of speech transcriptions and audio features for dialect identification (Malmasi et al., 2016). The combination of transcriptions and acoustic features has also provided good results for dialect identification (Zampieri et al., 2017). Following this trend, the 2016 Computational Paralinguistics Challenge (Schuller et al., 2016) also included an NLI task based on the spoken response. The NLI 2017 shared task attempts to combine these approaches by including a written response (essay) and a spoken response (speech transcriptions and i-vector acoustic features) for each subject. The task also allows for the fusion of all features.

Recent years have seen a large amount of work on employing text based features for the purpose of native language identification. The winning system (Jarvis et al., 2013) of NLI shared task 2013 featured a single model SVM system that used n-grams of lemmas, words, and part-of-speech tags. The authors normalized each text to unit length and obtained an accuracy of 83.60 %. In another work, Ionescu et al. (2014) applied a union of character n-gram based string kernels and obtained an accuracy of 85.30 % on the dataset from NLI shared task 2013.

Using the data from NLI shared task 2013, Bykh and Meurers (2014) explored the use of phrase structure rules for the purpose of NLI. The authors obtained an accuracy of 84.82 % which is similar to the results reported by previous authors. In another paper, Goutte et al. (2013) employed an ensemble of SVM classifiers trained on character, word, part-of-speech n-grams, and syntactic dependencies and showed that the system achieves an accuracy of 81.82 % at NLI task. Recently,

Proceedings of the 12th Workshop on Innovative Use of NLP for Building Educational Applications, pages 255–260
Copenhagen, Denmark, September 8, 2017.

Malmasi and Dras (2017) explored ensemble related classifiers using word, character, lemma, and grammar based features and found that stacking the classifiers' ensemble achieves an accuracy of 87.10 %.

In this paper, we used the single SVM model of Çöltekin and Rama (2016) that combines character n-grams with word n-grams for the essay task. We explored different ensemble models such as hard majority ensemble, mean majority ensemble, and stacked ensemble for the fusion task. In the case of speech task, we found that a linear classifier trained on i-vectors (alone) achieves an accuracy greater than 80 % on the test data. We also found that i-vectors combined with word unigrams from essays and speech transcriptions achieve an accuracy of 90.64 % on the test data. The main result from our experiments is that i-vectors contribute towards improving the performance of NLI systems.

We also experimented with adding POS tags as features, and a number of neural network classifiers. However, within our efforts, neither options improve the results. As a result we only submitted results with the linear models noted above, and we only discuss these models in detail in this paper.

The remainder of the paper is organized as follows. In section 2, we describe the different tasks and systems. In section 3, we describe the results of our experiments. We conclude our paper in section 4.

2 Methodology and Data

2.1 Task description

In this subsection, we provide a description of the three subtasks in NLI shared task 2017 (Malmasi and Dras, 2017). The goal of the shared task is to produce a system that can identify the native language of the test giver based on written response (essays), speech transcriptions, and audio files (i-vectors). The native languages are known beforehand and are as follows: Arabic, Chinese, French, German, Hindi, Italian, Japanese, Korean, Spanish, Telugu, Turkish.

The *essays* task is limited to using (only) written response for identifying the native language of the individual. The *speech* task consists of using speech transcriptions and i-vectors (fixed-length vectors representing some acoustic properties of whole utterances) for NLI. In the *fusion* task, we use essays, speech transcriptions, and i-vectors for the purpose of NLI.

The organizers provided separate training and development datasets for each task. The training dataset consisted of 11 000 examples and the development dataset consisted of 1 100 examples.

2.2 NLI with a single classifier

In this paper, we extracted character n-grams, word n-grams, and word skip-grams from essays and speech transcriptions for training our classifiers. Specifically, we used the following features in our experiments. We used a simple regular expression based tokenizer for extracting words and did not apply any filtering (e.g., case normalization).

- Word n-grams: Unigrams and bigrams.
- Character n-grams: We extracted character substrings of length from 1–9.
- Word skip-grams: We extracted word bigrams by skipping a intermediary word for extracting 1-skip word bigram (Ionescu et al., 2014).

For each task, we extracted the following features:

- *Essays task*: Each document is represented as a combination of word and character n-grams which are weighted using sub-linear tf-idf scaling (Jurafsky and Martin, 2009, p.805).
- *Speech task*: We used a combination of i-vectors, word and character n-grams (extracted from speech transcriptions). The word/character n-grams are weighted separately using sublinear tf-idf scaling and then combined with the i-vectors.
- *Fusion task*: We extracted word and character n-grams from both essays and speech transcriptions and, then, applied sublinear tf-idf scaling to the combined word and character n-gram vectors. Finally, we combined the i-vectors with the sublinear tf-idf scaled speech & transcriptions n-grams.

In all the tracks, we normalize the combined document vectors to unit length. We also tuned the number of character and word n-grams, as well as the SVM margin parameter 'C' for each task separately. The SVMs were not very sensitive to the changes in 'C' parameter. All linear SVM models were implemented with scikit-learn (Pedregosa

et al., 2011) and trained and tested using Liblinear backend (Fan et al., 2008). All our multi-class classifiers are trained in a one-vs-many fashion.

2.3 Ensemble classifiers

In a recent paper, Malmasi and Dras (2017) showed that ensemble classifiers perform the best at NLI task. Specifically, Malmasi and Dras (2017) showed that ensemble of linear classifiers trained on multiple feature types performed better than a single classifier trained on a combination of feature types. We trained an SVM classifier on each of the above listed feature types extracted from essays and speech transcriptions. In the case of i-vectors, we trained an LDA classifier (Hastie et al., 2009, p.106) since it performed better than the SVM classifier on the development data. A classifier trained on a feature type predicts both the label and the probability score for each class. Based on this, we created two ensembles as follows:

- Majority Ensemble: In this system, each classifier labels an example and the class with the highest frequency is chosen as the label for the instance.
- Mean probability Ensemble: In this system, the probability estimates for each class are added and the class with the highest sum is chosen as the label for the instance.
- Meta Classifier: Following Malmasi (2016), we train a linear SVM classifier for each feature type through ten-fold cross-validation on the training data. This step results in 10 classifiers for each feature type. For each feature type, we average the class probability estimates of the ten classifiers and then train a linear SVM classifier with the probability estimates as features and the corresponding class label as target class.

2.4 Submitted systems

- *Essay task:* We trained SVM classifiers on combinations of word n-grams (ranging from 1 to 3) and character n-grams (ranging from 1-9) and found that the SVM system trained with word bigrams and character 7-grams performed the best at F1-score on the development data. We submitted the results of the trained model as **w2c7**.
- *Speech track:* We submitted the following two systems:
  - *only i-vectors:* In our experiments, we found that a Linear Discriminant Classifier (LDA) trained on i-vectors performed better than an SVM model on the development data. We submitted the system as **LDA (only i-vectors)**.
  - *Transcripts + i-vectors:* We submitted the results of the SVM model trained on a combination of i-vectors, word bigrams, and character 7-grams (extracted from speech transcriptions) as **SVM (i+t)**.
- *Fusion track:* We submitted four systems in this task. The first two systems are based on two SVM models trained on different combinations of word- and character-ngrams. The third system is a mean majority ensemble based on different feature types. The fourth system is a meta classifier model based on different feature types.

3 Results

In this section, we describe the results of the submitted systems in each track.

3.1 Essay task

In this track, the best performing model is a linear SVM model trained with word bigrams and character 7-grams (*w2c7* model). We explored the effect of using higher order word and character n-grams for this task by training a linear SVM model on the training data and testing the model on the development data. In the case of development data, with w2c7 model, we report an accuracy of $84.09\,\%$ and an F1 score of $84.04\,\%$. The results on the test data for the same model is given in table 1. The results suggest that the model performed better on the testing data than development data. We also explored the effect of tuning the SVM hyperparameter 'C' and found that the F1-score on the development data are not sensitive to the 'C' parameter.

The confusion matrix for the essay task is given in figure 1. The confusion matrix shows that model makes most of the mistakes occur at the classification of Telugu vs. Hindi and Japanese vs. Korean language pairs. More generally, the system makes mistakes between languages that have a history of long geographical contact (Chinese-Japanese–Korean; Hindi–Telugu) or belong to the same language subgroup (French–

System	F1 (macro)	Accuracy
w2c7	0.871 6	0.871 8
Official baseline	0.710 4	0.710 9
Random baseline	0.090 9	0.090 9

Table 1: The results for word bigrams and character 7-grams using Linear SVMs for essay task.

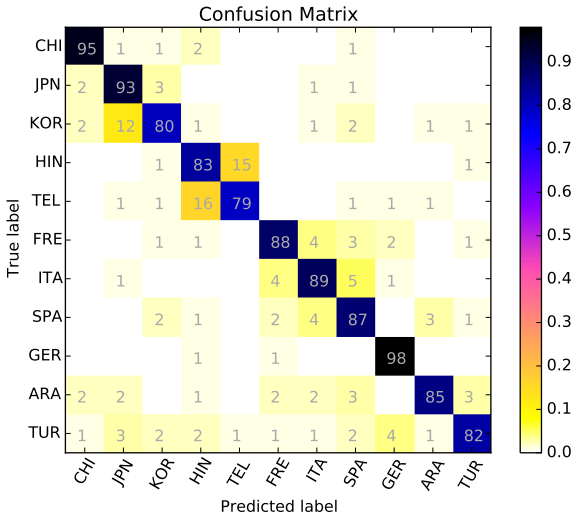

Figure 1: Confusion matrix for the *essay* track.

Italian–Spanish). In the case of Turkish, the model errs uniformly at classifying Turkish instances as instances of other classes.

3.2 Speech task

We submitted two systems in the case of speech task: an LDA classifier based on i-vectors and an SVM classifier based on the combined features of speech transcriptions and i-vectors. We expected that a combination of transcriptions and i-vectors might capture the acoustic features that would discriminate the highly confused language pairs such as Hindi–Telugu. However, the F1-scores in table 2 show that i-vectors alone perform better than a combination of transcriptions and i-vectors at NLI task. Although the combination model of transcriptions and i-vector features yield an F1-score of 81.57 % on the development data, the combined model performs poorly with test data. In contrast, the LDA model trained on i-vectors yielded an F1-score of 83.33 % on the test data.

The confusion matrix for the LDA model is presented in figure 2. The results suggest that the model makes most of its mistake at classifying Telugu–Hindi language pair. We hypothesized that i-vectors might be useful to discriminate Telugu–Hindi language pair since they might capture differences between languages that are in contact. However, the *LDA (only i-vector)* model errs more than the essay-based SVM model for the test dataset originating from the same set of individuals.

System	F1 (macro)	Accuracy
LDA (only i-vectors)	0.833 3	0.833 6
SVM (combined)	0.280 1	0.293 6
Official Baseline		
transcriptions	0.543 5	0.546 4
combined	0.798 0	0.798 2
Random Baseline	0.909 0	0.909 0

Table 2: Results of LDA classifier on i-vectors and the results on combined transcriptions and i-vectors.

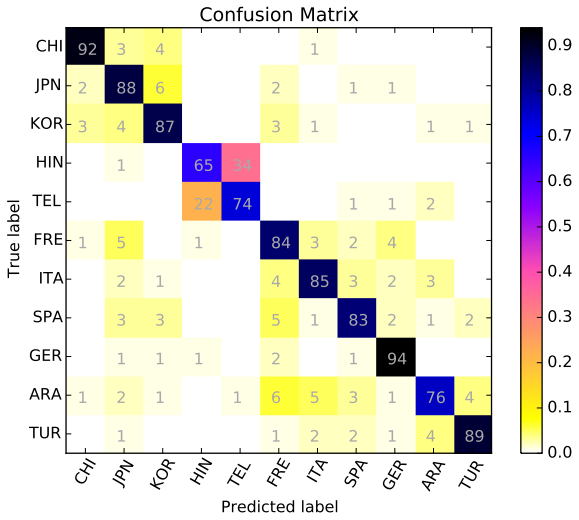

Figure 2: Confusion matrix for the *speech* task (i-vectors only).

3.3 Fusion task

We submitted four systems in this task.

The first system is a *Combined* feature system is a combination of the following features:

- Word bigrams and character 7-grams from essays (*w2c7* model)
- Word bigrams from transcriptions
- i-vectors

The combined feature system achieved an F1-score of 85.24 % on the development data and an F1-score of 88.71 % on the test data. The difference between the performance on the development

and test data is similar to that of the SVM model trained on essays data. We attribute the improvement from essay model SVM mainly to i-vector based features.

Due to the poor performance of the combination of transcriptions and i-vectors, we also explored if reducing the features would improve the performance of the model. After exploring different combinations of n-grams in essays and transcriptions, we found that the following feature combination (a 66 881 dimension vector) yielded an F1-score of 88.20 % on the development data and 90.65 % on the test data.

- Both Essays and transcriptions: Word unigrams and *no* character n-grams
- i-vectors

The third system is a mean probability ensemble trained on the following features:

- Essays: char ngrams (n ranging from 2–5), word ngrams (n ranging from 1–2), 1-skip word bigram
- Transcripts: word 1gram, 1-skip word bigram
- i-vectors

The mean probability ensemble yielded an F1-score of 89.93 % on the development data and a score of 91.75 % on the test data. The mean probability ensemble made the most number of mistakes in classifying Telugu–Hindi language pair but erred less than the essay based SVM model at other language pairs.

The meta classifier described in section 2.3 was trained on the following feature types and yielded an F1-score of 90.54 % on the development data.

- essays: character ngrams 2–7, word ngrams 1–2
- transcriptions: word 1-gram
- i-vectors: LDA

The meta classifier performed better than the mean probability ensemble on the development data. This result is in line with the previously reported results of Malmasi and Dras (2017). Surprisingly, the meta classifier performs worse on the test data.

4 Discussion

In this paper, we described our systems participating in the NLI shared task 2017. We participated in all the three tasks offered during this shared task campaign. We find that word unigram features in conjunction with i-vectors perform better than a combination of word or character based higher order n-gram features. We also find that transcription-based features do not improve the performance on the test data as is in the case of the combination system. All the systems make errors when discriminating between Hindi vs. Telugu. Another surprising result from experiments is that the Meta Classifier approach does not perform better than the mean probability ensemble which is not in line with the result of Malmasi and Dras (2017).

Besides the models we describe above, we also experimented with additional linguistic features (POS tags) and neural network classifiers. The POS tag n-gram features used together with our best-performing models did not improve the results. Furthermore, the best performing neural network architectures performed a few percentage scores worse than the linear models described in this paper in all of our experiments. Although this

System	F1 (macro)	Accuracy
Combined system	0.887 1	0.887 3
Simple system	0.906 5	0.906 4
Mean probability ensemble	0.917 5	0.917 3
Meta Classifier	0.848 1	0.848 2
Official Baseline		
essays and trans.	0.778 6	0.779 1
all	0.790 1	0.790 9
Random Baseline	0.909 0	0.909 0

Table 3: Results of different submissions for Fusion track.

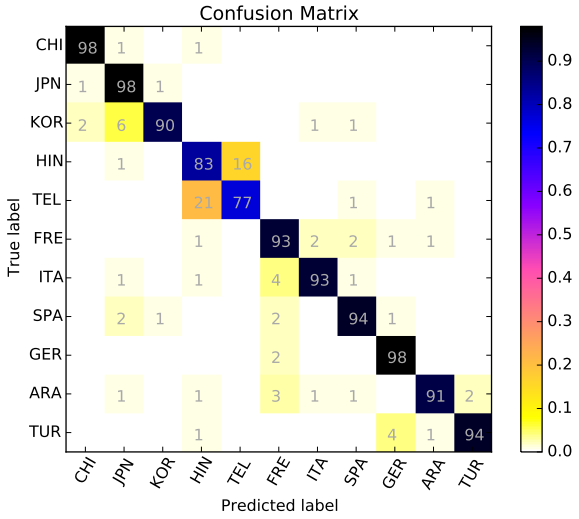

Figure 3: Confusion matrix of Mean probability ensemble for the *fusion* track.

is in line with our earlier experiments (Çöltekin and Rama, 2016, 2017) in a similar task, discriminating between similar languages and dialects (Malmasi et al., 2016; Zampieri et al., 2017), our experiments were not exhaustive and it is likely that one can get better results with neural networks with different architectures, and/or more data.

Acknowledgements

The first author is supported by the ERC Advanced Grant 324246 EVOLAEMP, which is gratefully acknowledged.

References

Serhiy Bykh and Detmar Meurers. 2014. Exploring Syntactic Features for Native Language Identification: A Variationist Perspective on Feature Encoding and Ensemble Optimization. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*. Dublin, Ireland, pages 1962–1973.

Çağrı Çöltekin and Taraka Rama. 2016. Discriminating Similar Languages with Linear SVMs and Neural Networks. In *Proceedings of the Third Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial3)*. Osaka, Japan, pages 15–24.

Çağrı Çöltekin and Taraka Rama. 2017. Tübingen system in VarDial 2017 shared task: experiments with language identification and cross-lingual parsing. In *Proceedings of the Fourth Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial)*. Valencia, Spain, pages 146–155. http://www.aclweb.org/anthology/W17-1218.

Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin. 2008. LIBLINEAR: A library for large linear classification. *Journal of Machine Learning Research* 9:1871–1874.

Cyril Goutte, Serge Léger, and Marine Carpuat. 2013. Feature space selection and combination for native language identification. In *BEA@ NAACL-HLT*. pages 96–100.

Trevor Hastie, Robert Tibshirani, and Jerome Friedman. 2009. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer series in statistics. Springer-Verlag New York, second edition.

Radu Tudor Ionescu, Marius Popescu, and Aoife Cahill. 2014. Can characters reveal your native language? A language-independent approach to native language identification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Scott Jarvis, Yves Bestgen, and Steve Pepper. 2013. Maximizing Classification Accuracy in Native Language Identification. In *Proceedings of the Eighth Workshop on Innovative Use of NLP for Building Educational Applications*. Atlanta, Georgia, pages 111–118.

Daniel Jurafsky and James H. Martin. 2009. *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*. Pearson Prentice Hall, second edition.

Shervin Malmasi. 2016. *Native Language Identification: Explorations and Applications*. Ph.D. thesis. http://hdl.handle.net/1959.14/1110919.

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

Shervin Malmasi and Mark Dras. 2017. Native Language Identification using Stacked Generalization. *arXiv preprint arXiv:1703.06541* .

Shervin Malmasi, Keelan Evanini, Aoife Cahill, Joel Tetreault, Robert Pugh, Christopher Hamill, Diane Napolitano, and Yao Qian. 2017. A Report on the 2017 Native Language Identification Shared Task. In *Proceedings of the 12th Workshop on Building Educational Applications Using NLP*. Copenhagen, Denmark.

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

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.

Björn Schuller, Stefan Steidl, Anton Batliner, Julia Hirschberg, Judee K. Burgoon, Alice Baird, Aaron Elkins, Yue Zhang, Eduardo Coutinho, and Keelan Evanini. 2016. The INTERSPEECH 2016 Computational Paralinguistics Challenge: Deception, Sincerity & Native Language. In *Interspeech 2016*. pages 2001–2005. https://doi.org/10.21437/Interspeech.2016-129.

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, pages 1–15.