Neural Networks and Spelling Features for Native Language Identification

Johannes Bjerva
CLCG
University of Groningen
j.bjerva@rug.nl

Gintarė Grigonytė
Department of Linguistics
Stockholm University
gintare@ling.su.se

Robert Östling
Department of Linguistics
Stockholm University
robert@ling.su.se

Barbara Plank
CLCG
University of Groningen
b.plank@rug.nl

Abstract

We present the RUG-SU team's submission at the Native Language Identification Shared Task 2017. We combine several approaches into an ensemble, based on spelling error features, a simple neural network using word representations, a deep residual network using word and character features, and a system based on a recurrent neural network. Our best system is an ensemble of neural networks, reaching an F1 score of 0.8323. Although our system is not the highest ranking one, we do outperform the baseline by far.

1 Introduction

Native Language Identification (NLI) is the task of identifying the native language of, e.g., the writer of an English text. In this paper, we describe the University of Groningen / Stockholm University (team RUG-SU) submission to NLI Shared Task 2017 (Malmasi et al., 2017). Neural networks constitute one of the most popular methods in natural language processing these days (Manning, 2015), but appear not to have been previously used for NLI. Our goal in this paper is therefore twofold. On the one hand, we wish to investigate how well a neural system can perform the task. On the other hand, we wish to investigate the effect of using features based on spelling errors.

2 Related Work

NLI is an increasingly popular task, which has been the subject of several shared tasks in recent years (Tetreault et al., 2013; Schuller et al., 2016; Malmasi et al., 2017). Although earlier shared task editions have focussed on English, NLI has recently also turned to including non-English languages (Malmasi and Dras, 2015). Additionally, although the focus in the past has been on using written text, speech transcripts and audio features have also been included in recent editions, for instance in the 2016 Computational Paralinguistics Challenge (Schuller et al., 2016). Although these aspects are combined in the NLI Shared Task 2017, with both written and spoken responses available, we only utilise written responses in this work. For a further overview of NLI, we refer the reader to Malmasi (2016).

Previous approaches to NLI have used syntactic features (Bykh and Meurers, 2014), string kernels (Ionescu et al., 2014), and variations of ensemble models (Malmasi and Dras, 2017; Tetreault et al., 2013). No systems used neural networks in the 2013 shared task (Tetreault et al., 2013), hence ours is one of the first works using a neural approach for this task, along with concurrent submissions in this shared task (Malmasi et al., 2017).

3 External data

3.1 PoS-tagged sentences

We indirectly use the training data for the Stanford PoS tagger (Manning et al., 2014), and for initialising word embeddings we use GloVe embeddings from 840 billion tokens of web data.[1]

3.2 Spelling features

We investigate learner misspellings, which is mainly motivated by two assumptions. For one, spelling errors are quite prevalent in learners' written production (Kochmar, 2011). Additionally, spelling errors have been shown to be influenced by phonological L1 transfer (Grigonytė and Hammarberg, 2014). We use the Aspell spell checker to detect misspelled words.[2]

[1] https://nlp.stanford.edu/projects/glove/

[2] http://aspell.net

4 Systems

4.1 Deep Residual Networks

Deep residual networks, or *resnets*, are a class of convolutional neural networks, which consist of several convolutional blocks with skip connections in between (He et al., 2015, 2016). Such skip connections facilitate error propagation to earlier layers in the network, which allows for building deeper networks. Although their primary application is image recognition and related tasks, recent work has found deep residual networks to be useful for a range of NLP tasks. Examples of this include morphological re-inflection (Östling, 2016), semantic tagging (Bjerva et al., 2016), and other text classification tasks (Conneau et al., 2016).

We apply resnets with four residual blocks. Each residual block contains two successive one-dimensional convolutions, with a kernel size and stride of 2. Each such block is followed by an average pooling layer and dropout ($p = 0.5$, Srivastava et al. (2014)). The resnets are applied to several input representations: word unigrams, and character 4- to 6-grams. These input representations are first embedded into a 64-dimensional space, and trained together with the task. We do not use any pre-trained embeddings for this subsystem. The outputs of each resnet are concatenated before passing through two fully connected layers, with 1024 and 256 hidden units respectively. We use the rectified linear unit (ReLU, Glorot et al. (2011)) activation function. We train the resnet over 50 epochs with the Adam optimisation algorithm (Kingma and Ba, 2014), using the model with the lowest validation loss. In addition to dropout, we use weight decay for regularisation ($\epsilon = 10^{-4}$, Krogh and Hertz (1992)).

4.2 PoS-tagged sentences

In order to easier capture general syntactic patterns, we use a sentence-level bidirectional LSTM over tokens and their corresponding part of speech tags from the Stanford CoreNLP toolkit (Manning et al., 2014). PoS tags are represented by 64-dimensional embeddings, initialised randomly; word tokens by 300-dimensional embeddings, initialised with GloVe (Pennington et al., 2014) embeddings trained on 840 billion words of English web data from the Common Crawl project.[3]

To reduce overfitting, we perform training by choosing a random subset of 50% of the sentences in an essay, concatenating their PoS tag and token embeddings, and running the resulting vector sequence through a bidirectional LSTM layer with 256 units per direction. We then average the final output vector of the LSTM over all the selected sentences from the essay, pass it through a hidden layer with 1024 units and rectified linear activations, then make the final predictions through a linear layer with softmax activations. We apply dropout ($p = 0.5$) on the final hidden layer.

4.3 Spelling features

Essays are checked with the Aspell spell checker for any misspelled words. If misspellings occur, we simply consider the first suggestion of the spell checker to be the most likely correction. The features for NLI classification are derived entirely from misspelled words. We consider deletion, insertion, and replacement type of corrections. Features are represented as pairs of original and corrected character sequences (uni, bi, tri), for instance:

```
visiters visitors
{(e,o),(te,to),(ter,tor)}
travellers travelers
{(l,0),(ll,l0),(ole,l0e)}
```

These features are fed to a logistic regression classifier with builtin cross-validation, as implemented in the scikit-learn library.[4]

4.4 CBOW features

We complement the neural approaches with a simple neural network that uses word representations, namely a *continuous bag-of-words* (CBOW) model (Mikolov et al., 2013). It represents each essay simply as the average embedding of all words in the essay. The intuition is that this simple model provides complementary evidence to the models that use sequential information. Our CBOW model was tuned on the DEV data and consists of an input layer of 512 input nodes, followed by a dropout layer ($p = 0.1$) and a single softmax output layer. The model was trained for 20 epochs with Adam using a batch size of 50. No pre-trained embeddings were used in this model. We additionally experiment with a simple multiplayer perceptron (MLP). In contrast to CBOW it uses n-hot features (of the size of the vocabulary),

[3] https://nlp.stanford.edu/projects/glove/

[4] http://scikit-learn.org/

Table 1: Official results for the essay task, with and without external resources (ext. res.).

Setting	System	F1 (macro)	Accuracy
Baselines	Random Baseline	0.0909	0.0909
	Official Baseline	0.7100	0.7100
No ext. res.	01 – Resnet (w_1+c_5)	0.8016	0.8027
	02 – Resnet (w_1+c_5)	0.7776	0.7782
	03 – Ensemble (Resnet (w_1+c_5), Resnet (c_4))	0.7969	0.7964
	04 – Ensemble (Resnet (w_1+c_5), Resnet (c_6), Resnet (c_4), Resnet (c_3))	0.8023	0.8018
	05 – Ensemble (Resnet (w_1+c_5), Resnet (c_6), Resnet (c_4), CBOW)	0.8149	0.8145
	06 – Ensemble (Resnet (w_1+c_5), Resnet (c_6), MLP, CBOW)	**0.8323**	**0.8318**
With ext. res.	01 – Ensemble (LSTM, Resnet (w_1+c_5))	**0.8191**	**0.8186**
	02 – Ensemble (LSTM, Resnet (w_1+c_5), Resnet (c_4))	0.8191	0.8195
	03 – Ensemble (Spell, LSTM, Resnet (w_1+c_5), Resnet (c_6), CBOW)	0.8173	0.8175
	04 – Ensemble (Spell, Resnet (w_1+c_5), Resnet (c_6), CBOW)	0.8055	0.8051
	05 – Ensemble (Spell, Spell, Resnet (w_1+c_5), Resnet (c_6), Resnet (c_4), CBOW)	0.8045	0.8048
	06 – Ensemble (LSTM, Resnet (w_1+c_5), Resnet (c_6), Resnet (c_4), CBOW)	0.8009	0.8007

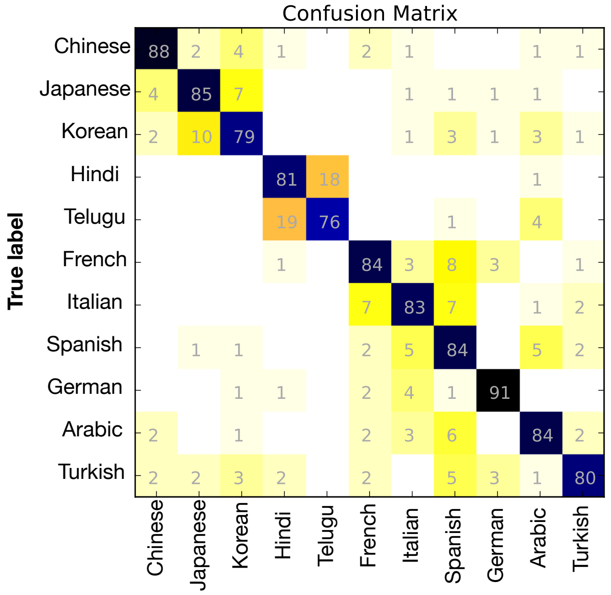

Figure 1: Confusion matrix for our best run (closed track, run 06)

a single layer with 512 nodes, sigmoid activation and dropout ($p = 0.1$). The remaining training parameters are the same as for CBOW. We see that this model adds complementary knowledge in the closed-track ensemble (run 06).

4.5 Ensemble

The systems are combined into an ensemble, consisting of a linear SVM. We use the probability distributions over the labels, as output by each system, as features for the SVM, as in meta-classification (Malmasi and Dras, 2017). The ensemble is trained and tuned on a random subset of the development set (70/30 split). For the selection of systems to include in the ensemble, we use the combination of systems resulting in the highest mean accuracy over five such random splits.

5 Results

The results when using external resources are lower than when not using them (Table 1). Our best result without external resources is an F1 score of 83.23, whereas we obtain F1 score of 81.91 with such resources. Figure 1 shows the confusion matrix of our best system's predictions (run 06). Most confusions occur in three groups: *Hindi* and *Telugu* (South Asian), *Japanese* and *Korean* (East Asian), and *French*, *Italian* and *Spanish* (South European).

6 Discussion

In isolation, the ResNet system yields a relatively high F1 score of 80.16. This indicates that, although simpler methods yield better results for this task, deep neural networks are also applicable. However, further experimentation is needed before such a system can outperform the more traditional feature-based systems. This is in line with previous findings for the related task of language identification (Medvedeva et al., 2017; Zampieri et al., 2017). Combining all of our systems without external data yields an F1 score of 83.23, which places our system in the third best performing group of the NLI Shared Task 2017 (Malmasi et al., 2017).

When adding external data, the best performing systems are those including the spelling system predictions and/or the LSTM predictions. However, the highest F1 score obtained (81.91) is lower than our best score without external resources. This can attributed to overfitting of the ensemble on the development data. It is nonetheless interesting that adding spelling features does boost performance within the external resources setting.

The main confusions of our system were within three groups. We suggest two reasons for this bias. On the one hand, the South European group also encompasses only Romance languages, hence the confusion could be attributed to the learners making similar mistakes in the grammar. However, both the South Asian group and the East Asian group comprise languages which are not related to one another. Therefore, it is reasonable to assume that the confusion is also due to a cultural bias, such as South European learners using more vacation-related words, or South Asian learners using words related to India (in which both of the languages in question are spoken).

7 Conclusions

We describe our system for the NLI Shared Task 2017, which is one the first system to involve a neural approach to this task. Although deep neural networks are able to perform this task, traditional methods still appear to be better.

Acknowledgments

This work was partially funded by the NWO–VICI grant 'Lost in Translation – Found in Meaning' (288-89-003).

References

Johannes Bjerva, Barbara Plank, and Johan Bos. 2016. Semantic tagging with deep residual networks. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*. The COLING 2016 Organizing Committee, pages 3531–3541. http://aclweb.org/anthology/C16-1333.

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.

Alexis Conneau, Holger Schwenk, Loïc Barrault, and Yann Lecun. 2016. Very deep convolutional networks for natural language processing. *arXiv preprint arXiv:1606.01781* .

Xavier Glorot, Antoine Bordes, and Yoshua Bengio. 2011. Deep sparse rectifier neural networks. In *International Conference on Artificial Intelligence and Statistics*. pages 315–323.

Gintarė Grigonytė and Björn Hammarberg. 2014. Pronunciation and spelling: the case of misspellings in swedish l2 written essays. In *6th International Conference on Human Language Technologies-The Baltic Perspective (Baltic HLT), Kaunas, Lithuania, September 26-27, 2014*. IOS Press, pages 95–98.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2015. Deep residual learning for image recognition. *arXiv preprint arXiv:1512.03385* .

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Identity mappings in deep residual networks. *CoRR* abs/1603.05027.

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)*. Association for Computational Linguistics.

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

Ekaterina Kochmar. 2011. *Identification of a Writers Native Language by Error Analysis*. Master's thesis, University of Cambridge.

Anders Krogh and John A Hertz. 1992. A simple weight decay can improve generalization. In *Advances in neural information processing systems*. pages 950–957.

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*. Association for Computational Linguistics, Copenhagen, Denmark.

Christopher D. Manning. 2015. Computational linguistics and deep learning. *Computational Linguistics* 41(4):701–707.

Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. 2014. The Stanford CoreNLP natural language processing toolkit. In *Association for Computational Linguistics (ACL) System Demonstrations*. pages 55–60. http://www.aclweb.org/anthology/P/P14/P14-5010.

Maria Medvedeva, Martin Kroon, and Barbara Plank. 2017. When sparse traditional models outperform dense neural networks: the curious case of discriminating between similar languages. In *Proceedings of VarDial 2017*. page 156.

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

Robert Östling. 2016. Morphological reinflection with convolutional neural networks. In *Proceedings of the 2016 Meeting of SIGMORPHON*. Association for Computational Linguistics, Berlin, Germany.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*. pages 1532–1543. http://www.aclweb.org/anthology/D14-1162.

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.

Nitish Srivastava, Geoffrey E Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research* 15(1):1929–1958.

Joel Tetreault, Daniel Blanchard, and Aoife Cahill. 2013. A Report on the First Native Language Identification Shared Task. In *Proceedings of the Eighth Workshop on Building Educational Applications Using NLP*. Association for Computational Linguistics, Atlanta, GA, USA.

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.