

R222 Assignment

Kaho Sato
King's College
ks789@cam.ac.uk

1 Introduction

There is an increasing interest in *Native Language Identification* (NLI). NLI is a task of detecting the native language (L1) of the author of the text. This may be treated as a sub-task of *author profiling*, where the goal is to predict traits of the author such as gender, age or psychometric traits (Estival et al., 2007). Author profiling is used, for instance, to narrow down the suspects of criminal activities (Abbasi and Chen, 2005) by filtering them with the attributes detected from the communication on online mediums. A possible commercial application is to collect customer information from the reviews for market intelligence (Glance et al., 2005), which could be especially effective when used in conjunction with sentiment analysis.

On the top of these applications, NLI can be used to create a better writing tutor system a non-native speaker of English. When using a non-native language (L2), one often applies the knowledge of their L1. This well-studied phenomenon is called *L1 transfer*. *CITEL1* transfer could result to correct expressions, but could also lead to errors which are specific to the particular L1. Based on this observation, there are several works (Chang et al., 2008; Rozovskaya and Roth, 2010, 2011; Dahlmeier and Ng, 2011) which investigate how the knowledge of the authors' first language could improve the grammatical error detection and correction in their texts. A writing tutor system can first used NLI, and use the detected L1 to have an improved error detection and correction.

There have been many works on NLI. Early works in NLI (Koppel et al., 2005; Tsur and Rappoport, 2007) showed the effectiveness of Support Vector Machine (SVM) in this task. SVM continues to be the popular choice, and 13 out of 24 participating teams, including the winning team (Jarvis et al., 2013) in the Native Language Identification Shared Task (Tetreault et al., 2013)

utilised SVM. The difference among these works comes down to the feature sets used, which range from lexical n-grams (Koppel et al., 2005; Tsur and Rappoport, 2007; Jarvis et al., 2013) to fragments of Tree Substitutional Grammar (Swanson and Charniak, 2012). In other words, feature engineering has been a large focus of NLI.

In this report, we perform NLI on the subset of Cambridge Learner Corpus (Nicholls; Yannakoudakis et al., 2011) using a character-based Convolutional Neural Network (CNN). Specifically, we use the character-based CNN designed by Zhang et al. (2015) to perform various classification tasks. We chose to use a character-based CNN as opposed a word-based CNN, as lexical features are in general considered to be more appropriate for the task of topic categorisation than of author profiling (Kochmar, 2011). CNNs have gained in popularity for its success in various practical applications, such as image recognition *cite*, natural language processing *cite* and playing Go *cite*. CNNs have a great advantage over traditional machine learning techniques such as SVM, which is the lack of feature engineering. The downside is that they typically require a large-scale dataset, and the dataset we have is fairly small. Indeed, applying the implementation by Zhang et al. (2015) with no modification to the dataset does not perform better than a simple baseline system which assigns the label arbitrarily. Another potential explanation for the poor performance is the length of each document in the dataset. Zhang et al. (2015) uses the first 1014 characters of each document as an input. However, many of the documents are shorter than 1014 characters and therefore require padding. This may have introduced some noise to the input. To partially mitigate these problems, we use a smaller window and sample a section of the document randomly, rather than always taking the first part. The results show that CNN performs better with a small window size. However, the per-

formance still does not match that of SVM.

The remaining of the report proceeds as follows; **FINISH**

2 Related Work

3 Background

4 Dataset

5 Architecture

We use the CNN described in (Zhang et al., 2015). The implementation is based on Torch (Collobert et al., 2011) and available online¹. In this section, we describe their design and the modifications we experimented with.

5.1 Input

We first need to encode documents to a fixed sized sequence of vectors, so it can be fed to the temporal convolutional module introduced in Section 3. Zhang et al. (2015) does this by lowercasing the document, and mapping each character to a one-hot vector. The alphabet used in our experiment is the same as in (Zhang et al., 2015) and is

```
abcdefghijklmnopqrstuvwxyz  
0123456789  
-,;.:!?"/\|_@#%&*+--=<>()[]{}
```

As the alphabet consists of 70 characters, each document is transformed into a sequence of vectors of size 70. If a character in a document is not present in the alphabet, it is encoded to an all-zero vector. Note that a whitespace is not part of the alphabet and therefore is also encoded to an all-zero vector.

(Zhang et al., 2015) performs a data augmentation in order to avoid the generalisation error. Specifically, they use an English thesaurus which is based on WordNet^{cite} and replace words in the document to create a “new” example. The number of words to be replaced and the index of the synonym used is chosen probabilistically. This data augmentation technique is appropriate if the task is to classify the documents according to its semantics. Indeed, the text classification tasks that Zhang et al. (2015) uses this CNN for are sentiment analysis and topic classification. Though the dataset we have is small and a data augmentation would have been useful, it was expected that this

particular technique would introduce an undesirable noise in training for author profiling, as word choice is a crucial cue for the attributes of the author, especially their L1. (Yarowsky, 2013) shows that the frequency distribution of words used in a corpus of computational linguistics articles from the ACL Anthology varies greatly from L1 to L1 of the author. For instance, a word “claim” is much less often used by Chinese native speakers than native speakers of other languages, and “complementary” is more frequently used by French or Spanish native speakers. The latter can potentially be explained by the latin origin of the word. To preserve such distinguishing features in the data, we turned off the data augmentation.

In (Zhang et al., 2015), the first 1014 characters are taken from each document and encoded to be used as an input. This is potentially problematic with the dataset we have. As shown in **CREATE TABLE AND REFERENCE HERE**, majority of the documents in the training set is significantly shorter than 1014 characters. Though pooling should help the model to be invariant to the location of a feature, having majority of the training examples padded extensively might be harmful.

To partially mitigate the problem of small dataset and short documents, instead of taking the first 1014 characters, we take a window of smaller size at an arbitrary location. For instance, given a document of length 1000 and a window size $l_w = 100$, a CNN may sample 100 characters at 900 different locations, effectively creating 900 samples. In this report, we refer to this approach as *random window sampling*. We experiment with $l_w \in \{123, 339, 555, 771, 1014\}$. Note that these numbers are chosen as they are compatible with the architecture.

5.2 Architecture

Figure 1 gives an illustration of the model described in (Zhang et al., 2015). The CNN consists of 6 convolutional, and 3 fully-connected layers. The kernel described in Section ?? is applied with stride 1. The activation function used is a threshold function th defined as follows:

$$th(x) = \begin{cases} x & \text{if } x > 1e-6 \\ 0 & \text{otherwise} \end{cases}$$

Max-pooling is applied with stride 3 at first, second and sixth convolutional layer. Table 1 shows the configuration of the convolutional layers. The

¹<https://github.com/zhangxiangxiao/Crepe>

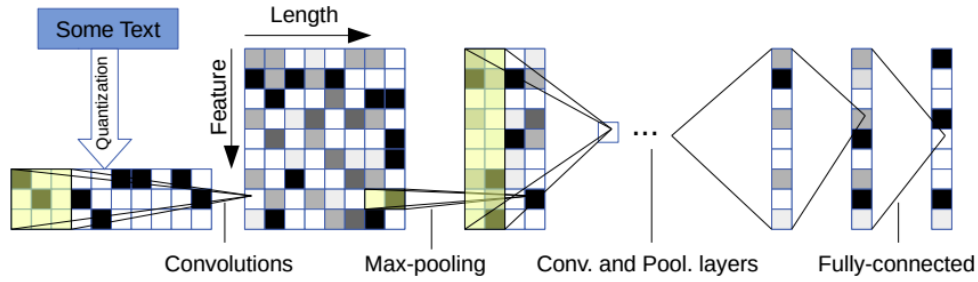


Figure 1: Overview of Architecture

Table 1: Convolutional Layers Used in (Zhang et al., 2015)

Layer	Frame Size	Kernel Width	Pooling Region Width
1	256	7	3
2	256	7	3
3	256	3	N/A
4	256	3	N/A
5	256	3	N/A
6	256	3	3

number of features of the input is 70, which is the size of the alphabet given in 5.1. As stated previously, in (Zhang et al., 2015) the length of the input is 1014. With our approach of random window sampling, the length of the input is equal to the window size l_w . The output frame length after the last convolutional layer is $(l_0 - 96)/27$, where l_0 is the length of the input. This defines a set of possible window sizes l_w .

Between the fully-connected layers, dropout modules (Hinton et al., 2012) with dropout probability of 0.5 are used for regularisation. Seventh and eighth layer have 1024 output units, and ninth layer have three as this is applied to three-class classification.

6 Results

7 Conclusion

References

- Ahmed Abbasi and Hsinchun Chen. Applying authorship analysis to arabic web content. In *International Conference on Intelligence and Security Informatics*, pages 183–197. Springer, 2005.
- Yu-Chia Chang, Jason S Chang, Hao-Jan Chen, and Hsien-Chin Liou. An automatic collocation writing assistant for taiwanese efl learners: A case of corpus-based nlp technology. *Com-*

puter Assisted Language Learning, 21(3):283–299, 2008.

- R. Collobert, K. Kavukcuoglu, and C. Farabet. Torch7: A matlab-like environment for machine learning. In *BigLearn, NIPS Workshop*, 2011.

- Daniel Dahlmeier and Hwee Tou Ng. Correcting semantic collocation errors with 11-induced paraphrases. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 107–117. Association for Computational Linguistics, 2011.

- Sze-Meng Jojo Wong Mark Dras and Mark Johnson. Topic modeling for native language identification. In *Australasian Language Technology Association Workshop 2011*, page 115, 2011.

- Dominique Estival, Tanja Gaustad, Son Bao Pham, Will Radford, and Ben Hutchinson. Author profiling for english emails. In *Proceedings of the 10th Conference of the Pacific Association for Computational Linguistics (PACLING07)*, pages 263–272, 2007.

- Natalie Glance, Matthew Hurst, Kamal Nigam, Matthew Siegler, Robert Stockton, and Takashi Tomokiyo. Deriving marketing intelligence from online discussion. In *Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*, pages 419–428. ACM, 2005.

- Geoffrey E Hinton, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever, and Ruslan R Salakhutdinov. Improving neural networks by preventing co-adaptation of feature detectors. *arXiv preprint arXiv:1207.0580*, 2012.
- Scott Jarvis, Yves Bestgen, and Steve Pepper. Maximizing classification accuracy in native language identification. 2013.
- Ekaterina Kochmar. *Identification of a writers native language by error analysis*. PhD thesis, 2011.
- Moshe Koppel, Jonathan Schler, and Kfir Zigdon. Determining an author’s native language by mining a text for errors. In *Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*, pages 624–628. ACM, 2005.
- Diane Nicholls. The cambridge learner corpus: Error coding and analysis for lexicography and elt.
- Alla Rozovskaya and Dan Roth. Generating confusion sets for context-sensitive error correction. In *Proceedings of the 2010 conference on empirical methods in natural language processing*, pages 961–970. Association for Computational Linguistics, 2010.
- Alla Rozovskaya and Dan Roth. Algorithm selection and model adaptation for esl correction tasks. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pages 924–933. Association for Computational Linguistics, 2011.
- Ben Swanson and Eugene Charniak. Native language detection with tree substitution grammars. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Short Papers-Volume 2*, pages 193–197. Association for Computational Linguistics, 2012.
- Joel Tetreault, Daniel Blanchard, and Aoife Cahill. A report on the first native language identification shared task. 2013.
- Oren Tsur and Ari Rappoport. Using classifier features for studying the effect of native language on the choice of written second language words. In *Proceedings of the Workshop on Cognitive Aspects of Computational Language Acquisition*, pages 9–16. Association for Computational Linguistics, 2007.
- Sze-Meng Jojo Wong, Mark Dras, and Mark Johnson. Exploring adaptor grammars for native language identification. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 699–709. Association for Computational Linguistics, 2012.
- Helen Yannakoudakis, Ted Briscoe, and Ben Medlock. A new dataset and method for automatically grading esol texts. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pages 180–189. Association for Computational Linguistics, 2011.
- Shane BergsmaDavid Yarowsky. Learning domain-specific, ll-specific measures of word readability. 2013.
- Xiang Zhang, Junbo Zhao, and Yann LeCun. Character-level convolutional networks for text classification. In *Advances in neural information processing systems*, pages 649–657, 2015.