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, 2003; 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-

Table 1: Distribution of L1 in the CLC FCE Dataset

L1	Count
Dutch	2
Swedish	15
Thai	63
Catalan	64
Chinese	66
Portuguese	68
German	69
Greek	74
Italian	76
Polish	76
Turkish	77
Japanese	80
Russian	82
Korean	86
French	146
Spanish	200

formance still does not match that of SVM.

The remaining of the report proceeds as follows; FINISH

2 Related Work

3 Background

4 Dataset

We use a subset of the Cambridge Learner Corpus (Nicholls, 2003) (CLC). CLC is a corpus which consists of scripts produced by learners with 86 different L1 for various Cambridge English Language Assessment examinations. Each document is annotated with basic information of the author, such as their L1 and age, making CLC appropriate corpus for author profiling.

We use a subset of the public available CLC FCE Dataset (Yannakoudakis et al., 2011) as a test set. This contains 1244 scripts, each containing two answers, produced by learners with 16 different L1 for the Cambridge ESOL First Certificate in English (FCE) examination in 2010 and 2011.

Table 1 shows the number of scripts for each L1. We extracted scripts produced by Japanese, Russian and Italian native speakers to create a test set of 476 examples. (Recall that each script contains two answers.) We chose these three languages for two reasons; Firstly, the number of the scripts available for each language is fairly even. Secondly, they belong to separate language families,

Table 2: Statistics of the Character Length of Documents in Training Set

Min	58
Max	3710
Median	376.5
Average	742.1
Standard Deviation	716.4

Table 3: Statistics of the Character Length of Documents in Test Set

Min	685
Max	1822
Median	1081
Average	1099
Standard Deviation	186.8

which should make the classification more manageable. We selected 3000 answers from the CLC, 1000 for each L1, to build a training set.

The level of proficiency is another variable that influences the writing of a learner, and therefore may introduce an undesirable bias. Each Cambridge ESOL examination expects a certain level of proficiency from the candidates. The reference levels are provided by Common European Framework of Reference for Languages (CEFR) (COUNCIL, 2001) and FCE is aimed for learners at CEFR level B (i.e. independent user). For the training set, for each language, we selected 250 answers by learners at CEFR level A and C (i.e. basic and proficient user, respectively), and 500 answers by learners at CEFR level B. We included more answers produced by learners at CEFR level B in order to assimilate the training set to the test set. Besides, it is reasonable to assume that there are more independent users than basic or proficient users.

Table 2 and 3 gives the statistic of the length of the documents in terms of character in training and test set respectively. As can be seen, there is a much larger variance in training set than in test set. This is due to the fact that the training set contains answers for 15 different exams, where the expected length of the answers are different from one another, whereas the test set only contains answers for FCE.

From the training set, we took 300 examples to create a validation set. This was used to see the number of epochs

5 Architecture

We use the CNN described in (Zhang et al., 2015). The implementation is based on Torch (Collobert et al., 2011) and available online[1]. 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 described in Section 3. Zhang et al. (2015) do 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) perform a data augmentation in order to avoid the generalisation error. Specifically, they use an English thesaurus which is based on WordNetciteand 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) use 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, during the training, instead of feeding the first 1014 characters of the document to the model, 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 different examples. In this report, we refer to this approach as *random window sampling*. For testing, we take the window of the same size at the beginning of the document for an easier comparison of the results.

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), which we use without modification. 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:

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

Max-pooling is applied with stride 3 at first, second and sixth convolutional layer. Table 4 shows the configuration of the convolutional layers. The 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

[1] https://github.com/zhangxiangxiao/Crepe

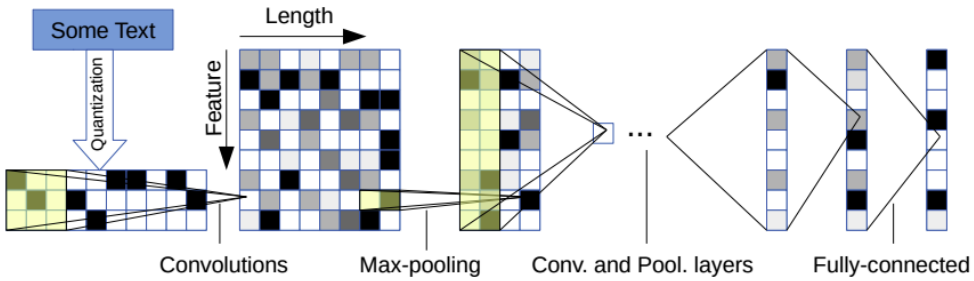

Figure 1: Overview of Architecture

Table 4: 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

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.

At each epoch, a minibatch of size 128, chosen randomly from the training set, is used to update the weight with stochastic gradient decent (Polyak, 1964) with momentum (Sutskever et al., 2013) 0.9 and step size 0.01.

6 Results

No Yes ?Table 5 shows the error rate of the CNN with the input size 1014. Model (A) is the unmodified implementation described in (Zhang et al., 2015). As described in Section 5.1, we modify the implementation in two ways; one is applying the random window sampling, and the other is switching off the data augmentation using a thesaurus.

All the models perform better than a simple baseline system which chooses the label arbitrary, though only by small margin. As can be seen in the table, (C) outperforms both (A) and (B), showing the effectiveness of the two modifications. As stated previously, this CNN is shown effective for sentiment analysis and topic categorisation (Zhang et al., 2015).

We argue that random window sampling is an effective method for this task. In (Zhang et al., 2015), the types of documents treated are reviews or articles, and the first part of these documents tend to be more informative about the overall content which is the objective of the network. Using only the start of the document therefore means less noise in the training data for the problems discussed in (Zhang et al., 2015). However, in the context of NLI, there is no clear relation between the informative features and the locations. Therefore it could be useful to augment the small dataset with random window sampling, rather than discarding a possibly useful information which may be contained in the later part of the training example.

A possible explanation of (A) performing better than (B) is that augmentation using thesaurus helps generalise the bias introduced by using only the first part of the document. The bias may be related to the topic of the document or to the format that the exam prompt asks for. There is an imbalance in the exams taken by native speakers of each language, and therefore there may be a prompt on a certain topic or asks for a certain format which was chosen more frequently by a learners of a cer-

Table 5: Error Rate of Models with Input Size 1014

	Random Window Sampling	Augmentation	Error Rate
(A)	No	Yes	0.628
(B)	No	No	0.664
(C)	Yes	No	0.638

Table 6: Error Rate of Models with Random Window Sampling

	Input Size	Error Rate
(D)	123	0.578
(E)	339	0.632
(F)	555	0.654
(G)	771	0.682
(C)	1014	0.638

tain L1. Clearly, if a network learns to classify based on the topic or the format, it would not generalise and perform poorly on the test set. As mentioned previously, the first part of the document may contain more information about the topic, and it possibly is more affected by the format. For instance, there are many documents which are in the format of a letter. These always start with a greeting, and if there are more of such documents are present for a certain L1, the network may associate the greeting with the L1, which would not generalise.

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. *Computer 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.

OF EUROPE COUNCIL. Common european framework of reference for languages: Learning, teaching, assessment (cefr). *Electronic version:¡ http://www. coe. int/t/dg4/linguistic/cadre1_en. asp*, 2001.

Daniel Dahlmeier and Hwee Tou Ng. Correcting semantic collocation errors with l1-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. 2003.

Boris T Polyak. Some methods of speeding up the convergence of iteration methods. *USSR Computational Mathematics and Mathematical Physics*, 4(5):1–17, 1964.

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.

Ilya Sutskever, James Martens, George Dahl, and Geoffrey Hinton. On the importance of initialization and momentum in deep learning. In *International conference on machine learning*, pages 1139–1147, 2013.

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, l1-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.