Neural sequence modelling for learner error prediction

Zheng Yuan
The ALTA Institute
Department of Computer Science and Technology
University of Cambridge
zheng.yuan@cl.cam.ac.uk

Abstract

This paper describes our use of two recurrent neural network sequence models: sequence labelling and sequence-to-sequence models, for the prediction of future learner errors in our submission to the 2018 Duolingo Shared Task on Second Language Acquisition Modeling (SLAM). We show that these two models capture complementary information as combining them improves performance. Furthermore, the same network architecture and group of features can be used directly to build competitive prediction models in all three language tracks, demonstrating that our approach generalises well across languages.

1 Introduction

Most recent work on second language acquisition (SLA) has focused on intermediate-to-advanced learners in assessment settings driven by a series of shared tasks (Dale and Kilgarriff, 2011; Dale et al., 2012; Ng et al., 2013, 2014; Lee et al., 2015, 2016; Daudaravicius et al., 2016). The 2018 Duolingo Shared Task on Second Language Acquisition Modeling (SLAM) (Settles et al., 2018) targets early stage learners and aims to provide personalised learning instructions. Participating teams are provided with transcripts from exercises submitted by learners over their first 30 days of learning on Duolingo,[1] which are annotated for token (word) level errors. The task is to predict what errors each learner will make in the future based on their learning history. There are three language tracks in this shared task:

- *en_es*: native Spanish speakers learning English;
- *es_en*: native English speakers learning Spanish;
- *fr_en*: native English speakers learning French.

Teams can either focus on a particular language track, or explore generalised models and features across all three languages.

Inspired by the success of neural sequence models in grammatical error detection and correction (Yuan and Briscoe, 2016; Rei and Yannakoudakis, 2016; Yannakoudakis et al., 2017; Schmaltz et al., 2017), we propose two recurrent neural network sequence models for this problem: sequence labelling and sequence-to-sequence modelling. We demonstrate the utility of these two models for the future learner error prediction task. We also provide evidence of performance gains by using an ensemble of these two models, suggesting that they are complementary to each other.

For model development, we focus on the English track only and language-specific features are introduced and studied. When it comes to official evaluation, two new prediction systems, one for the *es_en* track and another for the *fr_en* track, are built using the same network architecture and the same (hyper-)parameter setting, without tuning for new datasets or languages. Competitive results on all three language tracks show that our approach generalises well and might be used as a generic solution across different languages.

The remainder of this paper is organised as follows: Section 2 describes our approach and two neural sequence models in detail, Section 3 discusses the feature types that we exploit in our models, Section 4 reports our experiments and results on the development set for the *en_es* track, Section 5 presents our official results on the test sets for all three language tracks. Finally, Section 6 provides conclusions and ideas for future work.

[1] https://www.duolingo.com

Proceedings of the Thirteenth Workshop on Innovative Use of NLP for Building Educational Applications, pages 381–388
New Orleans, Louisiana, June 5, 2018. ©2018 Association for Computational Linguistics

2 Approach

We introduce two models for the task of future learner error prediction: a sequence labelling model and a sequence-to-sequence model. The following sections describe these two models.

2.1 Neural sequence labelling

We treat error prediction as a sequence labelling problem. Similar to Yannakoudakis et al. (2017), we construct a bidirectional recurrent neural network for detecting future learner errors. Unlike their system, error-free and correct sequences are fed into our model, and the goal is to predict where a learner is likely to make token-level errors based on their learning history. The model receives a sequence of tokens $\mathbf{x} = (x_1, x_2, ..., x_T)$ as input, and assigns a label y to each input token x. A bidirectional long short-term memory (LSTM) (Hochreiter and Schmidhuber, 1997) is used to learn context-specific representations:

$$\overrightarrow{h_t} = \text{LSTM}(x_t, \overrightarrow{h_{t-1}}) \quad (1)$$

$$\overleftarrow{h_t} = \text{LSTM}(x_t, \overleftarrow{h_{t+1}}) \quad (2)$$

$$h_t = [\overrightarrow{h_t}; \overleftarrow{h_t}] \quad (3)$$

where $\overrightarrow{h_t}$ is the hidden state of the forward-moving LSTM at time t, that reads the input sequence from the first token to the last; $\overleftarrow{h_t}$ is the hidden state of the backward-moving LSTM at time t, which reads the input sequence in reverse order; and h_t is the concatenation of both hidden states, that captures both historical and future sequential information.

A softmax output layer predicts the label distribution for each input token, given the whole input sequence $\mathbf{x}$:

$$p(y_t|\mathbf{x}) = \text{softmax}(W_o h_t) \quad (4)$$

where W_o is an output weight matrix.

We optimise the model by minimising categorical cross-entropy between the predicted label distributions and the gold labels:

$$E = -\sum_{t=1}^{T} \log p(y_t|\mathbf{x}) \quad (5)$$

2.2 Sequence-to-sequence modelling

We utilise a sequence-to-sequence model with a soft attention mechanism similar to that of Yuan and Briscoe (2016), which contains a bidirectional LSTM encoder and an attention-based LSTM decoder. An encoder first reads and encodes an input sequence $\mathbf{x} = (x_1, x_2, ..., x_T)$ into hidden state representations $\mathbf{h} = (h_1, h_2, ..., h_T)$, which is the same as the one used in our sequence labelling model (see Section 2.1, Equation 3). A decoder then generates an output sequence $\mathbf{y} = (y_1, y_2, ..., y_T)$[2] by predicting the next token y_t based on the input sequence $\mathbf{x}$ and all the previously generated tokens $\{y_1, y_2, ..., y_{t-1}\}$:

$$p(y_t|\{y_1, ..., y_{t-1}\}, \mathbf{x}) = \text{softmax}(W_o s_t) \quad (6)$$

where W_o is a decoder output weight matrix, and s_t is the hidden state of the LSTM decoder at decoding time t:

$$s_t = \text{LSTM}(s_{t-1}, y_{t-1}, c_t) \quad (7)$$

where c_t is the input sequence representation for predicting the output token y_t, and is calculated using a soft attention mechanism:

$$c_t = \sum_{j=1}^{T}(\alpha_{tj} h_j) \quad (8)$$

The weight α_{tj} is computed with a softmax function:

$$\alpha_{tj} = \frac{\exp(e_{tj})}{\sum_{k=1}^{T} \exp(e_{tk})} \quad (9)$$

A feedforward neural network is used to represent the energy function:

$$e_{tj} = \tanh(W_\alpha s_{t-1} + U_\alpha h_j) \quad (10)$$

where W_α and U_α are attention weight matrices.

3 Feature space

Besides original word tokens, new features (in the form of discrete labels) are introduced, which provide additional exercise and learner information. These features are described briefly below.

[2] For the error prediction task, the number of tokens generated in the output sequence $\mathbf{y}$ must equal the number of tokens in the input sequence $\mathbf{x}$.

3.1 Exercise-level feature set

user: a unique identifier for each learner;

format: the exercise format (*reverse_translate*, *reverse_tap*, or *listen*);[3]

session: the exercise session type (*lesson*, *practice*, or *test*);[4]

client: the learner's device platform (*android*, *ios*, or *web*);

country: the country from which the learner has done the exercise.

3.2 Token-level feature set

part of speech (POS): the POS tag of the word;

dependency edge label (DEP): the grammatical relation (GR) between the word and its head.

3.3 Language-specific feature set

CEFR word level: The Common European Framework of Reference (CEFR) (Council of Europe, 2011) describes what language learners can do at different stages of their learning and defines language proficiency in six levels: A1, A2, B1, B2, C1 and C2, with A1 being the lowest and C2 the highest. These six CEFR levels can be grouped into three broad levels: basic (A1 and A2), independent (B1 and B2) and proficient (C1 and C2).

The CEFR levels for all the English words appeared in the dataset are extracted from the English Vocabulary Profile (EVP),[5] which is based on the 50-million word Cambridge Learner Corpus (CLC) and the 1.2-billion word Cambridge English Corpus (CEC). The EVP is a free online vocabulary resource that contains information about which words and phrases are known and used by learners at each CEFR level (Capel, 2012).

Even though we only focus on English words here, it is worth noting that the CEFR levels were designed in a way that can be applied to all languages. Therefore, if resources for other languages similar to the EVP became available, we can then make use of this feature for other languages.

CLC error rate: We collect error rate information from the CLC, which is a large annotated corpus of learner English developed by Cambridge University Press and Cambridge English Language Assessment since 1993 (Nicholls, 2003). It comprises examination scripts written by learners of English who took Cambridge English examinations around the world with over 80 L1s and representing all six CEFR levels.

Two criteria are applied to create two sub corpora:

- CLC(KET): contains examination scripts for A2 Key, formerly known as Cambridge English: Key (KET)[6]; and A2 Key for Schools, formerly known as Cambridge English: Key for Schools (KETfS)[7].

  KET is the lowest level General English examination in the Cambridge English range, which targets at A2 level. KETfS is at the same level as KET, but its examination content is targeted at the interests and experiences of schoolchildren.

- CLC(ES): contains examination scripts written by native speakers of Spanish, which account for around 24.6% of the non-native speakers represented in the CLC.

For every word w, an error rate $E(w)$ is defined as:

$$E(w) = \frac{\text{count}(s \neq w, t = w)}{\text{count}(t = w)} \tag{11}$$

where count$(t = w)$ is the number of times the word w is seen in the target side (*i.e.* corrected version) of the corpus, and count$(s \neq w, t = w)$ is the number of times any word except w in the source side (*i.e.* original version) has been corrected to the word w in the target side.

We compute $E(w)$ from the CLC, CLC(KET) and CLC(ES); and then create two new features **CLC-KET** and **CLC-ES**:

[3] *reverse_translate*: learners are asked to read a sentence written in their L1, and then translate it into L2; *reverse_tap*: an easier version of *reverse_translate*, where learners are given a bank of words and distractors; *listen*: learners are asked to listen to an utterance in L2, and then transcribe it.

[4] The *lesson* sessions (about 77% of all the data) introduce new words; the *practice* sessions (22%) contain only previously-seen words; and the *test* sessions (1%) are quizzes that allow learners to "skip" a particular skill unit of the curriculum.

[5] http://www.englishprofile.org/wordlists

[6] http://www.cambridgeenglish.org/exams-and-tests/key

[7] http://www.cambridgeenglish.org/exams-and-tests/key-for-schools

$$\text{CLC-KET} = \begin{cases} 1 & \text{if } \frac{E_{\text{CLC(KET)}}}{E_{\text{CLC}}} > 1 \\ 0 & \text{otherwise} \end{cases} \tag{12}$$

$$\text{CLC-ES} = \begin{cases} 1 & \text{if } \frac{E_{\text{CLC(ES)}}}{E_{\text{CLC}}} > 1 \\ 0 & \text{otherwise} \end{cases} \tag{13}$$

All the exercise-level and token-level features are directly extracted from the metadata and pre-processed data provided by the shared task organisers. The language-specific features are only generated for the English data to be used in the *en_es* track.

4 Experiments and results

4.1 Dataset and evaluation

The shared task dataset comprises answers submitted by more than 6,000 Duolingo users over the course of their first 30 days. Token-level binary labels are provided:

Correct reference :	*She*	*is*	*my*	*mother*
Learner answer:	*She*	*is*		*mader*
Label:	0	0	1	1

Matched tokens are given the label '0'; and missing or misspelt tokens (ignoring capitalisation, punctuation and accents) are given the label '1' to indicate an error. Only correct references and label sequences are provided, not original learners' responses. Therefore, in our experiments, we map **correct** reference to its **label** sequence.

The dataset is partitioned sequentially into training, development and test sets, which all contain the same group of learners. The training set contains the first 80% of the sessions for each learner, followed by the next 10% for development and the final 10% for testing. Each learner's test items are subsequent to their development items, which in turn are all subsequent to their training items.

During development, we focus on learners of English. The training set provided for the *en_es* track contains approximately 2,622,958 tokens (however, only 13% are labelled with '1') in about 824,012 sentences. The development set includes additional 387,374 tokens in 115,770 sentences. All the data has been pre-processed using the Google SyntaxNet dependency parser[8] by the shared task organisers.

System performance is evaluated in terms of area under the ROC curve (AUROC) and F1 (with a threshold of 0.5).

4.2 Training

All our models are built using OpenNMT (Klein et al., 2017). For the sequence labelling model, our training procedure is similar to Yannakoudakis et al. (2017)); while for the sequence-to-sequence model, we follow Yuan and Briscoe (2016). Additionally, we set the source and target word embedding sizes to 750, as well as the LSTM hidden layer size. We no longer limit the vocabulary size or the maximum sentence length as both of them are small enough to train effectively. New features defined in Section 3 are added to the models incrementally and results are presented in the next section.

4.3 Results

Evaluation results on the development set for the *en_es* track are reported. We also include a baseline model provided by the shared task organisers for comparison purposes. The baseline model uses L2-regularised logistic regression, trained with stochastic gradient descent (SGD) weighted by frequency (Settles et al., 2018).

Sequence labelling model Results for the sequence labelling models are presented in Table 1, and all our models outperform the baseline (Table 1 #0). We start by adding exercise-level features incrementally (Table 1 #1-5). Introducing new exercise-level features yields consistent improvements in overall performance. The one trained on all our exercise-level features gives the best AUROC and F1 scores (Table 1 #5).

Token-level features (Table 1 #6-7) and language-specific features (Table 1 #8-10) are then added to the current best model. However, none of them yields further gains. A closer inspection of the training data reveals a number of cases where **POS** and **DEP** tags provided by the shared task organises are not reliable, as in the following examples (incorrect tags are marked in red):

[8] https://github.com/tensorflow/models/tree/master/research/syntaxnet

#	Feature	AUROC	F1
0	baseline	0.776	0.173
1	token + user	0.784	0.421
2	token + user + format	0.809	0.453
3	token + user + format + session	0.825	0.470
4	token + user + format + session + client	0.834	0.476
5	token + user + format + session + client + country	**0.837**	**0.480**
6	token + user + format + session + client + country + POS	0.807	0.447
7	token + user + format + session + client + country + DEP	0.830	0.474
8	token + user + format + session + client + country + CEFR	0.823	0.469
9	token + user + format + session + client + country + CLC-KET	0.825	0.471
10	token + user + format + session + client + country + CLC-ES	0.825	0.470

Table 1: Results of our sequence labelling models on the *en_es* development set. The results of our best model are marked in **bold**.

Token	**POS**	**DEP**	**Label**
A	DET	det	0
man	NOUN	ROOT	0
a	PUNCT	punct	0
woman	DET	det	0

Token	**POS**	**DEP**	**Label**
The	DET	det	1
judge	ADJ	amod	1
returns	NOUN	ROOT	1

Since we use the tags in the dataset directly, without cleaning any noisy data or pre-processing the data again, it is not surprising that adding these features yields worse performance.

In terms of the language-specific features, we also notice that the **CEFR word level** feature is not very informative as not all the words in the dataset are also in the EVP; and for words that are, most of them turn out to be at either A1 or A2 level.

Sequence-to-sequence model We follow the same training procedure to build sequence-to-sequence models - see Table 2. Similar results are observed: all our models perform better than the baseline (Table 2 #0); exercise-level features contribute to the overall performance improvements (Table 2 #1-5); and token-level and language-specific features seem to be detrimental and bring performance down (Table 2 #6-10). The best sequence-to-sequence model uses all the exercise-level features, achieving an AUROC score of 0.837 and an F1 score of 0.464 - see Table 2 #5.

Combining two sequence models Our best sequence labelling model (`seqlabel`) and our best sequence-to-sequence model (`seq2seq`) achieve the same AUROC score of 0.837; while `seqlabel` yields a better F1 score of 0.480, compared to an F1 score of 0.464 for `seq2seq`.

We further combine these two best models using linear interpolation:

$$P_{combined} = (1 - \lambda)P_{seqlabel} + \lambda P_{seq2seq} \quad (14)$$

where $P_{seqlabel}$ represents the score from the sequence labelling model, $P_{seq2seq}$ represents the score from the sequence-to-sequence model, and λ is a parameter that controls the impact the sequence-to-sequence model has on the final score. After tuning λ on the development set, we set it to 0.5.

Results of our best individual models and the final combined model are reported in Table 3. We can see that the combined model yields the overall best results, which suggests that our two individual neural sequence models capture complementary information even though they are both trained on the same group of features.

5 Official evaluation results

Our submissions to the shared task are the results of our best systems. As each participating team is allowed to submit up to 10 runs, we first run our best sequence labelling, sequence-to-sequence and combined systems from the previous section on the *en_es* test set.

After determining that our language-specific features are not helpful, we train new models for

#	Feature	AUROC	F1
0	baseline	0.776	0.173
1	token + user	0.787	0.353
2	token + user + format	0.800	0.431
3	token + user + format + session	0.811	0.441
4	token + user + format + session + client	0.825	0.448
5	token + user + format + session + client + country	**0.837**	**0.464**
6	token + user + format + session + client + country + POS	0.829	0.460
7	token + user + format + session + client + country + DEP	0.823	0.448
8	token + user + format + session + client + country + CEFR	0.805	0.433
9	token + user + format + session + client + country + CLC-KET	0.804	0.433
10	token + user + format + session + client + country + CLC-ES	0.805	0.433

Table 2: Results of our sequence-to-sequence models on the *en_es* development set. The results of our best model are marked in **bold**.

Model	AUROC	F1
seqlabel	0.837	0.480
seq2seq	0.837	0.464
combined	**0.843**	**0.481**

Table 3: Results of our best models on the *en_es* development set. The best results are marked in **bold**.

the *es_en* and *fr_en* tracks using the same network architecture and the same group of features as for *en_es*. No tuning of (hyper-)parameters is performed for new datasets or languages.

The official results of our submissions for all three language tracks are reported in Table 4. Results on the *en_es* test set are similar to those on the *en_es* development set (see Table 3) - no significant drop is observed. The `combined` model produces the best overall performance, and the `seqlabel` model outperforms the `seq2seq` model. In the *fr_en* track, the `combined` model again yields the highest AUROC and F1 scores, followed by the `seq2seq` model and the `seqlabel` model. Our *es_en* `seq2seq` model had not finished training by the shared task submission deadline, therefore, we only submit the *es_en* `seqlabel` model. Based on the results for the other two language tracks, we expect our *es_en* results might be further improved by combining a `seqlabel` model and a `seq2seq` model.

6 Conclusions and future work

In this paper, we have described the use of recurrent neural sequence labelling and sequence-to-sequence models for future learner error prediction. We have provided evidence of further performance gains by combining them together, showing that these two types of sequence models are complementary. We have also explored different types of features, which capture exercise-level, token-level and language-specific information. Furthermore, we have demonstrated that the same network architecture and group of features can be applied directly to build competitive prediction systems across all three languages, without the need for language-specific parameter tuning.

Results of our best systems on the official test sets yield: AUROC=0.841 (ranked sixth out of the fifteen participating teams) and F1=0.479 (ranked third) for the *en_es* track; AUROC=0.835 (ranked sixth) and F1=0.508 (ranked third) for *fr_en*; and AUROC=0.807 (ranked sixth) and F1=0.435 (ranked fifth) for *es_en*.

Plans for future work include combining the training and development sets to train new models, using better quality token-level features, and exploring other exercise-level features like the amount of **time** it took for the learner to construct and submit their answer and the number of **days** since the learner started using Duolingo. We would also like to test our approach as well as our language-specific features on a broader scale (*i.e.* using corpora which cover language learners at different levels, ideally ranging from basic to proficient).

Acknowledgments

We would like to thank Ted Briscoe and Meng Zhang for their valuable comments and suggestions. We are also grateful to Christopher Bryant

	en_es		*fr_en*		*es_en*	
Model	**AUROC**	**F1**	**AUROC**	**F1**	**AUROC**	**F1**
seqlabel	0.836	0.467	0.825	0.498	0.807	0.435
seq2seq	0.830	0.465	0.830	0.500	-	-
combined	0.841	0.479	0.835	0.508	-	-
baseline	0.774	0.190	0.771	0.281	0.746	0.175
top-performing	0.861	0.561	0.857	0.573	0.838	0.530

Table 4: Official results of our submitted systems on the test sets for all three tracks: `seqlabel` is our best sequence labelling model, `seq2seq` is our best sequence-to-sequence model, and `combined` is the combination of these two models. For comparison, we also include the `baseline` results provided by the shared task organisers and the results from the `top-performing` systems.

and Ahmed Zaidi for providing us with the CLC and EVP resources. Our gratitude goes also to the shared task organisers for coordinating the task. We acknowledge NVIDIA for an Academic Hardware Grant.

References

Annette Capel. 2012. Completing the English Vocabulary Profile: C1 and C2 vocabulary. *English Profile Journal*, 3(e1).

Council of Europe. 2011. *Common European Framework of Reference for Languages: Learning, Teaching, Assessment*. Cambridge University Press.

Robert Dale, Ilya Anisimoff, and George Narroway. 2012. HOO 2012: A Report on the Preposition and Determiner Error Correction Shared Task. In *Proceedings of the 7th Workshop on Innovative Use of NLP for Building Educational Applications*, pages 54–62, Montréal, Canada. Association for Computational Linguistics.

Robert Dale and Adam Kilgarriff. 2011. Helping Our Own: The HOO 2011 Pilot Shared Task. In *Proceedings of the 13th European Workshop on Natural Language Generation*, pages 242–249, Nancy, France. Association for Computational Linguistics.

Vidas Daudaravicius, Rafael E. Banchs, Elena Volodina, and Courtney Napoles. 2016. A Report on the Automatic Evaluation of Scientific Writing Shared Task. In *Proceedings of the 11th Workshop on Innovative Use of NLP for Building Educational Applications*, pages 53–62, San Diego, CA. Association for Computational Linguistics.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long Short-Term Memory. *Neural Computation*, 9(8):1735–1780.

Guillaume Klein, Yoon Kim, Yuntian Deng, Jean Senellart, and Alexander Rush. 2017. OpenNMT: Open-Source Toolkit for Neural Machine Translation. In *Proceedings of ACL 2017, System Demonstrations*, pages 67–72, Vancouver, Canada. Association for Computational Linguistics.

Lung-Hao Lee, Gaoqi RAO, Liang-Chih Yu, Endong XUN, Baolin Zhang, and Li-Ping Chang. 2016. Overview of NLP-TEA 2016 Shared Task for Chinese Grammatical Error Diagnosis. In *Proceedings of the 3rd Workshop on Natural Language Processing Techniques for Educational Applications*, pages 40–48, Osaka, Japan. The COLING 2016 Organizing Committee.

Lung-Hao Lee, Liang-Chih Yu, and Li-Ping Chang. 2015. Overview of the NLP-TEA 2015 Shared Task for Chinese Grammatical Error Diagnosis. In *Proceedings of the 2nd Workshop on Natural Language Processing Techniques for Educational Applications*, pages 1–6, Beijing, China. Association for Computational Linguistics.

Hwee Tou Ng, Siew Mei Wu, Ted Briscoe, Christian Hadiwinoto, Raymond Hendy Susanto, and Christopher Bryant. 2014. The CoNLL-2014 Shared Task on Grammatical Error Correction. In *Proceedings of the Eighteenth Conference on Computational Natural Language Learning: Shared Task*, pages 1–14, Baltimore, Maryland. Association for Computational Linguistics.

Hwee Tou Ng, Siew Mei Wu, Yuanbin Wu, Christian Hadiwinoto, and Joel Tetreault. 2013. The CoNLL-2013 Shared Task on Grammatical Error Correction. In *Proceedings of the Seventeenth Conference on Computational Natural Language Learning: Shared Task*, pages 1–12, Sofia, Bulgaria. Association for Computational Linguistics.

Diane Nicholls. 2003. The Cambridge Learner Corpus - error coding and analysis for lexicography and ELT. In *Proceedings of the Corpus Linguistics 2003 Conference*, pages 572–581.

Marek Rei and Helen Yannakoudakis. 2016. Compositional Sequence Labeling Models for Error Detection in Learner Writing. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1181–1191, Berlin, Germany. Association for Computational Linguistics.

Allen Schmaltz, Yoon Kim, Alexander Rush, and Stuart Shieber. 2017. Adapting Sequence Models for Sentence Correction. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2807–2813, Copenhagen, Denmark. Association for Computational Linguistics.

Burr Settles, Chris Brust, Erin Gustafson, Masato Hagiwara, and Nitin Madnani. 2018. Second Language Acquisition Modeling. In *Proceedings of the 13th Workshop on Innovative Use of NLP for Building Educational Applications*. Association for Computational Linguistics.

Helen Yannakoudakis, Marek Rei, Øistein E. Andersen, and Zheng Yuan. 2017. Neural Sequence-Labelling Models for Grammatical Error Correction. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2795–2806, Copenhagen, Denmark. Association for Computational Linguistics.

Zheng Yuan and Ted Briscoe. 2016. Grammatical error correction using neural machine translation. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 380–386, San Diego, California. Association for Computational Linguistics.