Adapting Grammatical Error Correction Based on the Native Language of Writers with Neural Network Joint Models

Shamil Chollampatt[1] and **Duc Tam Hoang**[2] and **Hwee Tou Ng**[1,2]
[1]NUS Graduate School for Integrative Sciences and Engineering
[2]Department of Computer Science
National University of Singapore
shamil@u.nus.edu, {hoangdt,nght}@comp.nus.edu.sg

Abstract

An important aspect for the task of grammatical error correction (GEC) that has not yet been adequately explored is adaptation based on the native language (L1) of writers, despite the marked influences of L1 on second language (L2) writing. In this paper, we adapt a neural network joint model (NNJM) using L1-specific learner text and integrate it into a statistical machine translation (SMT) based GEC system. Specifically, we train an NNJM on general learner text (not L1-specific) and subsequently train on L1-specific data using a Kullback-Leibler divergence regularized objective function in order to preserve generalization of the model. We incorporate this adapted NNJM as a feature in an SMT-based English GEC system and show that adaptation achieves significant $F_{0.5}$ score gains on English texts written by L1 Chinese, Russian, and Spanish writers.

1 Introduction

Grammatical error correction (GEC) deals with the automatic correction of errors (spelling, grammar, and collocation errors), particularly in non-native written text. The native language (L1) background of the writer has a noticeable influence on the errors made in second language (L2) writing, and considering this factor can potentially improve the performance of GEC systems. For example, consider the following sentence written by a Finnish writer (Jarvis and Odlin, 2000): *"When they had escaped **in** the police car they sat under the tree.*" The preposition *in* appears to be grammatically correct. However, in the given context, the preposition '*from*' is the correct choice in place of the preposition '*in*'. Finnish learners of English tend to overgeneralize the use of the preposition '*in*'. Knowledge of L1 makes the correction more probable whenever the preposition *in* appears in texts written by Finnish writers. Similarly, Chinese learners of English tend to make frequent verb tense and verb form errors, since Chinese lacks verb inflection (Shaughnessy, 1977). The cross-linguistic influence of L1 on L2 writing is a highly complex phenomenon, and the errors made by learners cannot be directly attributed to the similarities or differences between the two languages. As Ortega (2009) points out, learners seem to operate on two complementary principles: "what works in L1 may work in L2 because human languages are fundamentally alike; but if it sounds too L1-like, it will probably not work in L2". In this paper, we follow a data-driven approach to model these influences and adapt GEC systems using L2 texts written by writers of the same L1 background.

The two most popular approaches for grammatical error correction are the classification approach (Dahlmeier et al., 2012; Rozovskaya et al., 2014) and the statistical machine translation (SMT) approach (Chollampatt et al., 2016; Junczys-Dowmunt and Grundkiewicz, 2014). The SMT approach has emerged as a popular paradigm for GEC because of its ability to learn text transformations from ill-formed to well-formed text enabling it to correct a wide variety of errors including complex errors that are difficult to handle for the classification approach (Rozovskaya and Roth, 2016). The phrase-based SMT approach has been used in state-of-the-art GEC systems (Rozovskaya and Roth, 2016;

Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, pages 1901–1911,
Austin, Texas, November 1-5, 2016.

Chollampatt et al., 2016; Hoang et al., 2016). The SMT approach does not model error types specifically, nor does it require linguistic analysis like parsing and part-of-speech (POS) tagging. We adopt a phrase-based SMT approach to GEC in this paper. Additionally, we implement and incorporate a neural network joint model (NNJM) (Devlin et al., 2014) as a feature in our SMT-based GEC system. It is easy to integrate an NNJM into the SMT decoding framework as it uses a fixed-window context and it has shown to improve SMT-based GEC (Chollampatt et al., 2016). We adapt the NNJM to L1-specific data (i.e., English text written by writers of a particular L1) and obtain significant improvements over the baseline which uses an unadapted NNJM. Adaptation is done by using the unadapted NNJM trained on general domain data (i.e., not L1-specific) using a log likelihood objective function with self-normalization (Devlin et al., 2014) as the starting point, and training for subsequent iterations using the smaller L1-specific in-domain data with a modified objective function which includes a Kullback-Leibler (KL) divergence regularization term. This modified objective function prevents overfitting on the smaller in-domain data and preserves the generalization capability of the NNJM. We show that this method of adaptation works on very small and high-quality L1-specific data as well (50–100 essays).

In summary, the two major contributions of this paper are as follows. (1) This is the first work that performs L1-based adaptation for GEC using the SMT approach and covering all error types. (2) We introduce a *novel* method of NNJM adaptation and demonstrate that this method can work with in-domain data that are much smaller than the general domain data.

2 Related Work

In the past decade, there has been increasing attention on GEC in English, mainly due to the growing number of English as second language (ESL) learners around the world. The popularity of this problem grew further through Helping Our Own (HOO) (Dale and Kilgarriff, 2011; Dale et al., 2012) and CoNLL shared tasks (Ng et al., 2013; Ng et al., 2014). The majority of the published work on GEC aimed at building classifiers or rule-based systems for specific error types and combined them to build hybrid systems (Dahlmeier et al., 2012; Rozovskaya et al., 2014).

The cross-linguistic influences between L1 and L2 have been mainly used for the task of native language identification (Massung and Zhai, 2016). It has also been used in typology prediction (Berzak et al., 2014) and predicting error distributions in ESL data (Berzak et al., 2015). L1-based adaptation has previously shown to improve GEC for specific error types using the classification approach. Rozovskaya and Roth (2010) used an approach to correct preposition errors by restricting the candidate corrections to those observed in L1-specific data. They further added artificial training data that mimic the error frequency in L1-specific text to improve accuracy. In their later work, Rozovskaya and Roth (2011) focused on L1-based adaptation for preposition and article correction, by modifying the prior probabilities in the naïve Bayes classifier during decision time based on L1-specific ESL learner text. Both approaches use native data for training, but rely on non-native L1-specific text to introduce artificial errors or to modify the prior probabilities. Dahlmeier and Ng (2011) implemented a system to correct collocation errors, by adding paraphrases derived from L1 into the confusion set. Specifically, they use a bilingual L1-L2 corpus, to obtain L2 paraphrases, which are likely to be translated to the same phrase in L1. There is no prior work on L1-based adaptation for GEC using the machine translation approach, which is a one of the most popular approaches for GEC.

With the availability of large-scale error corrected data (Mizumoto et al., 2011), the statistical machine translation (SMT) approach to GEC became popular and was employed in state-of-the-art GEC systems. Comparison of the classification approach and the machine translation approach can be found in (Rozovskaya and Roth, 2016) and (Susanto et al., 2014). Recently, an end-to-end neural machine translation framework was proposed for GEC (Yuan and Briscoe, 2016), which was shown to achieve competitive results. Neural network joint models have shown to be improve SMT-based GEC systems (Chollampatt et al., 2016) due to their ability to model words and phrases in a continuous space, access to larger contexts from source side, and abil-

ity to learn non-linear mappings from input to output. In this paper, we exploit the advantages of the SMT approach and neural network joint models (NNJMs) by adapting an NNJM based on the L1 background of the writers and integrating it into the SMT framework. We perform KL divergence regularized adaptation to prevent overfitting on the smaller in-domain data. KL divergence regularization was previously used by Yu et al. (2013) for speaker adaptation. Joty et al. (2015) proposed another NNJM adaptation method, which uses a regularized objective function that encourages a network trained on general-domain data to be closer to an in-domain NNJM. Other adaptation techniques used in SMT include mixture modeling (Foster and Kuhn, 2007; Moore and Lewis, 2010; Sennrich, 2012) and alternative decoding paths (Koehn and Schroeder, 2007).

3 A Machine Translation Framework for Grammatical Error Correction

We formulate GEC as a translation task from a possibly erroneous input sentence to a corrected sentence. We use the popular phrase-based SMT system, Moses (Koehn et al., 2007), which employs a log linear model to find the best correction hypothesis T^* given an input sentence S:

$$T^* = \underset{T}{\operatorname{argmax}}\, P(T|S) = \underset{T}{\operatorname{argmax}} \sum_{i=1}^{N} \mu_i f_i(T, S)$$

where μ_i and $f_i(T, S)$ are the i^{th} feature weight and feature function, respectively. We use the standard features in Moses, without any re-ordering models. The two main components of an SMT system are the translation model (TM) and the language model (LM). The TM (typically, a phrase table), responsible for generating hypotheses, is trained using parallel data, i.e., learner-written sentences (source data) and their corresponding corrected sentences (target data). It also scores the hypotheses using features like forward and inverse phrase translation probabilities and lexical weights. The LM is trained on well-formed text and ensures the fluency of the corrected output. The feature weights μ_i are computed by minimum error rate training (MERT), optimizing the $F_{0.5}$ measure (Junczys-Dowmunt and Grundkiewicz, 2014) using a development set. The $F_{0.5}$ measure computed using the MaxMatch scorer (Dahlmeier and Ng, 2012) is the standard evaluation metric for GEC used in the CoNLL-2014 shared task (Ng et al., 2014), weighting precision twice as much as recall.

Apart from the TM and the n-gram LM, we add a neural network joint model (NNJM) (Devlin et al., 2014) as a feature, following Chollampatt et al. (2016), who reported that NNJM improves the performance of a state-of-the-art SMT-based GEC system. Unlike Recurrent Neural Networks (RNNs) and Long Short Term Memory networks (LSTMs), NNJMs have a feed-forward architecture which relies on a fixed context. This makes it easy to integrate NNJMs into a machine translation decoder as a feature. The feature value is given by $\log P(T|S)$, which is the sum of the log probabilities of individual target words in the hypothesis T given the context:

$$\log P(T|S) \approx \sum_{i=1}^{|T|} \log P(t_i|h_i) \tag{1}$$

where $|T|$ is the number of words in the target hypothesis (corrected sentence), t_i is the i^{th} target word, and h_i is the context of t_i. The context h_i consists of $n-1$ previous target words and m source words surrounding the source word that is aligned to the target word t_i.

Each dimension in the output layer of the neural network (Chollampatt et al., 2016) gives the probability of a word t in the output vocabulary given its context h:

$$P(y = t|h) = \frac{\exp(U_t(h))}{Z(h)} = \frac{\exp(U_t(h))}{\sum_{t' \in V_o} \exp(U_{t'}(h))}$$

where $U_t(h)$ is the unnormalized output score before the softmax, and V_o is the output vocabulary.

The neural network parameters which include the weights, biases, and embedding matrices are trained using back propagation with stochastic gradient descent (LeCun et al., 1998). Instead of using the noise contrastive estimation (NCE) loss as done in (Chollampatt et al., 2016), we use the log likelihood objective function with a self-normalization term similar to Devlin et al. (2014):

$$L = \frac{1}{N} \sum_{i=1}^{N} \left[\log p(y = t_i|h_i) - \alpha \log^2(Z(h_i))\right] \tag{2}$$

where N is the number of training instances, and t_i is the target word in the i^{th} training instance. Each training instance consists of a target word t and its context h. α is the self-normalization coefficient which we set to 0.1, following Devlin et al. (2014). The training can be done efficiently on GPUs. We adapt this NNJM using L1-specific learner text using a Kullback-Leibler divergence regularized objective function as described in Section 4.

4 KL Divergence Regularized Adaptation

We first train an NNJM with the general-domain data (the erroneous and corrected texts, not considering the L1 of the writers) as described in the previous section. Let $p^{GD}(y|h)$ be the probability distribution estimated by the general-domain NNJM. Starting from this NNJM, subsequent iterations of training are done using the L1-specific in-domain data alone. The in-domain data consists of the erroneous texts written by writers of a specific L1 and their corresponding corrected texts. This adaptive training is done using a modified objective function having a regularization term K, which is used to minimize the KL divergence between $p^{GD}(y|h)$ and the network's output probability distribution $p(y|h)$ (Yu et al., 2013):

$$K = \frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{V_o} p^{GD}(y = t_j|h_i)\log p(y = t_j|h_i)$$

The term K will prevent the estimated probability distribution from deviating too much from that of the general domain NNJM during training. Our final objective function for the adaptation step is a linear combination of the terms in L and K, with a regularization weight λ and a self-normalization coefficient α:

$$\begin{aligned} L' =& \lambda K + (1-\lambda)\frac{1}{N}\sum_{i=1}^{N}\log p(y = t_i|h_i) \\ & - \alpha\frac{1}{N}\sum_{i=1}^{N}\log^2(Z(h_i)) \end{aligned}$$

We integrate the unadapted NNJM and adapted NNJM independently into our SMT-based GEC system in order to compare the effect of adaptation.

5 Other Adaptation Methods

We compare our method against two other adaptation methods previously used in SMT.

Translation Model Interpolation: Following Sennrich (2012), we linearly interpolate the features in two phrase tables, one trained on in-domain data (L1-specific learner text) and the other on out-of-domain data. The interpolation weights are set by minimization of perplexity using phrase pairs extracted from our in-domain development set. The lexical weights are re-computed from the lexical counts and the interpolation weights are re-normalized if a phrase pair exists only in one of the phrase tables.

Neural Domain Adaptation Model: Joty et al. (2015) proposed an adaptation of NNJM for SMT. They first train an NNJM using in-domain data, and then perform regularized adaptation on the general domain data (concatenation of in-domain and out-of-domain data) which restricts the model from drifting away from the in-domain NNJM. Specifically, they add a regularization term J to the objective function in their adaptive training step:

$$J = \frac{1}{N}\sum_{i=1}^{N} p^{ID}(y = t_i|h_i)\log p(y = t_i|h_i)$$

where $p^{ID}(y|h)$ id probability distribution estimated by the in-domain NNJM.

NDAM has the following drawbacks compared to our method: (1) Regularization is done using probabilities of the target words alone and not on the entire probability distribution over all words, leading to a weak regularization. (2) If the in-domain data is too small, the probability distribution learnt by the in-domain NNJM will be overfitted. Therefore, encouraging adaptation to be closer to this probability distribution may not yield a good model. Our method, on the other hand, can utilize in-domain data of very small sizes to fine tune a general NNJM. (3) Their method requires retraining of the model on complete training data in order to adapt to each domain. On the contrary, our method can adapt a single general model to different domains using small in-domain data, leading to a considerable reduction in training time.

We re-implement their method by incorporating this regularization term into the log likelihood objec-

tive function with self-normalization, L (Equation 2), during adaptive training.

6 Data and Evaluation

The training data consist of two corpora: the NUS Corpus of Learner English (NUCLE) (Dahlmeier et al., 2013) and the Lang-8 Learner Corpora v2 (Mizumoto et al., 2011). We extract texts written by learners who learn only English from Lang-8. A language identification tool `langid.py`[1] (Lui and Baldwin, 2011) is then used to obtain purely English sentences. In addition, we remove noisy source-target sentence pairs in Lang8 where the ratio of the lengths of the source and target sentences is outside $[0.5, 2.0]$, or their word overlap ratio is less than 0.2. A sentence pair where the source or target sentence has more than 80 words is also removed from both NUCLE and Lang-8. The statistics of the data after pre-processing are shown in Table 1.

Corpus	#sents	#src tokens	#tgt tokens
NUCLE	57,063	1,156,460	1,151,278
LANG-8	2,048,654	24,649,768	25,912,707
CONCAT	2,105,717	25,806,228	27,063,985

Table 1: Statistics of training data

We obtain L1-specific in-domain data for adaptation based on the L1 information provided in Lang-8. Adaptation is performed on English texts written by learners of three different L1 backgrounds: Chinese, Russian, and Spanish. The statistics of the in-domain data from Lang-8 for each L1 are given in Table 2. For each L1, its out-of-domain data are obtained by excluding the L1-specific in-domain data (from Table 2) from the combined training data (CONCAT).

L1	#sents	#src tokens	#tgt tokens
Chinese	260,872	3,521,336	3,688,098
Russian	43,488	566,517	596,692
Spanish	19,357	292,257	309,236

Table 2: Statistics of L1-specific data in Lang-8

We use the publicly available CLC-FCE (Yannakoudakis et al., 2011) corpus to obtain the development and test data. The FCE corpus contains 1,244 scripts written by 1,244 distinct candidates sitting the Cambridge ESOL First Certificate in English (FCE) examination in 2000 and 2001. The corpus identifies the L1 of each writer. We extract the scripts written by Chinese, Russian, and Spanish native writers. We split the data for each L1 into two roughly equal parts based on the number of scripts, of which one part is used as the development data and other part is used as the test data. Splitting based on the number of scripts ensures that there is no overlap between the writers of the development and test data, as each script is written by a unique learner. The details of the FCE dataset corresponding to each L1 are given in Table 3.

	#scripts	#sents	#src tokens	#tgt tokens	#errors
L1: *Chinese*					
DEV	33	1,041	15,424	15,601	1,751
TEST	33	1,078	15,640	15,816	1,487
L1: *Russian*					
DEV	41	1,125	17,021	17,267	1,782
TEST	42	1,263	18,738	18,920	1,934
L1: *Spanish*					
DEV	100	2,281	41,375	41,681	4,133
TEST	100	2,431	41,557	42,035	4,237

Table 3: Statistics of the FCE dataset for each L1

For evaluation, we use the $F_{0.5}$ measure, computed by the M^2scorer v3.2 (Dahlmeier and Ng, 2012), as our evaluation metric. The error annotations in FCE are converted to the format required by the M^2scorer. The statistics of error annotations after converting to this format are given in Table 3. To deal with the instability of parameter tuning in SMT, we perform five runs of tuning and calculate the statistical significance by stratified approximate randomization test, as recommended by (Clark et al., 2011).

7 Experiments and Results

7.1 Baseline SMT-based GEC system

We use Moses (Version 3) to build all our SMT-based GEC systems. The phrase table of the baseline system (S_{CONCAT}) is trained using the complete

[1] https://github.com/saffsd/langid.py

	L1: *Chinese*			L1: *Russian*			L1: *Spanish*		
	P	**R**	**$F_{0.5}$**	**P**	**R**	**$F_{0.5}$**	**P**	**R**	**$F_{0.5}$**
S_{IN}	50.03	16.11	35.09	38.11	16.99	30.52	43.40	12.74	29.28
S_{OUT}	49.88	17.34	36.23	54.78	21.15	41.54	57.18	16.10	37.83
S_{CONCAT}	51.72	17.56	37.23	54.17	21.70	41.62	55.45	16.93	38.09
S_{CONCAT} + $NNJM_{BASELINE}$	50.47	18.75	37.63	55.22	21.73	42.15	58.30	16.42	38.60
NNJM adaptation using KL divergence regularization									
S_{CONCAT} + $NNJM_{ADAPTED}$	56.02	17.59	38.90	55.71	22.53	43.03	59.05	16.77	39.24
S_{CONCAT} + $NNJM_{ADAPTED\ (FCE)}$	53.82	18.60	**38.91**	56.03	22.46	**43.13**	58.88	16.95	**39.38**
Comparison to other adaptation techniques									
TM_{INT} + $NNJM_{BASELINE}$	55.70	17.18	38.38	54.97	21.90	42.21	58.32	16.44	38.59
S_{CONCAT} + NDAM	56.56	16.76	38.31	54.60	22.03	42.11	58.28	16.64	38.83
TM_{INT} + $NNJM_{ADAPTED}$	55.89	17.62	38.81	56.30	21.75	42.70	57.04	16.97	38.73
Using smaller general domain data									
S_{CONCAT} + $NNJM_{SMALL\text{-}BASELINE}$	53.29	17.47	37.75	55.34	20.87	41.55	58.05	16.46	38.55
S_{CONCAT} + $NDAM_{SMALL}$	53.89	17.36	37.87	55.29	21.09	41.70	56.82	16.69	38.36
S_{CONCAT} + $NNJM_{SMALL\text{-}ADAPTED}$	52.41	17.40	37.37	56.03	21.17	42.09	58.34	16.79	39.01

Table 4: Precision (P), recall (R), and $F_{0.5}$ of L1-based adaptation of GEC systems. All results are averaged over 5 runs of tuning and evaluation.

training data. We use two 5-gram language models (LMs) trained using KenLM (Heafield et al., 2013). One LM is trained on the English Wikipedia (about 1.78 billion tokens) and another on the target side of the complete training data. The system is tuned using MERT, optimizing the $F_{0.5}$ measure on the L1-specific development data in Table 2.

For comparison, we show two other baselines S_{IN} and S_{OUT}, where the phrase tables for each L1 are trained on the in-domain data only (Table 2) and the out-of-domain data only, respectively. The results of the above baseline GEC systems on L1 Chinese, Russian, and Spanish FCE test data are summarized in Table 4. We enhance the baseline S_{CONCAT} with an NNJM feature, as described in following subsection.

7.2 NNJM Adaptation

We implement NNJM in Python using the deep learning library Theano[2] (Bergstra et al., 2010) in order to use the massively parallel processing power of GPUs for training. We first train an NNJM ($NNJM_{BASELINE}$) with complete training data for 10 epochs. The source context window size is set to 5 and the target context window size is set to 4, making it a (5+5)-gram joint model. Training is done using stochastic gradient descent with a mini-batch size of 128 and learning rate of 0.1. To speed up training and decoding, a single hidden layer neural network is used with an input embedding dimension of 192 and 512 hidden units. We use a self-normalization coefficient of 0.1. We pick 16,000 and 32,000 most frequent words on the source and target sides as our source context vocabulary and target context vocabulary, respectively. The output vocabulary is set to be the same as the target vocabulary. The vocabulary is selected from the complete training data, and not based on the L1-specific in-domain data. We add the self-normalized NNJM as a feature to our baseline GEC system, S_{CONCAT} to build a stronger baseline. This is referred to as S_{CONCAT} + $NNJM_{BASELINE}$ in Table 4.

We perform adaptation on $NNJM_{BASELINE}$ by training for 10 additional epochs using the in-domain training data alone. We use the same hyper-parameters, network structure, and vocabulary, but with the KL-divergence regularized objective function (regularization weight $\lambda = 0.5$). We train the adapted NNJM ($NNJM_{ADAPTED}$) specific to each L1. We integrate these to our baseline GEC system, and the adapted systems are referred to as S_{CONCAT} + $NNJM_{ADAPTED}$ in Table 4. The results are averaged over five runs of tuning and evaluation. Our evaluation shows that each adapted system S_{CONCAT}

[2]http://deeplearning.net/software/theano

+ $\text{NNJM}_{\text{ADAPTED}}$ outperforms every baseline system (S_{IN}, S_{OUT}, S_{CONCAT}, and S_{CONCAT} + $\text{NNJM}_{\text{BASELINE}}$) significantly on all three L1s ($p < 0.01$).

7.3 Comparison to Other Adaptation Techniques

We compare our method to two different adaptation techniques described in Section 5: Translation Model Interpolation (TM_{INT}) (Sennrich, 2012) and Neural Domain Adaptation Model (NDAM) (Joty et al., 2015)[3]. The optimization of interpolation weights for TM_{INT} is done using the L1-specific FCE development data. NDAM is trained on the complete training data (CONCAT) for 10 epochs by regularizing using an in-domain NNJM also trained for 10 epochs on L1-specific in-domain data from Lang-8. For NDAM, we use the same vocabulary and hyperparameters as our NNJMs.

The results are shown in the rows TM_{INT} + $\text{NNJM}_{\text{BASELINE}}$ and S_{CONCAT} + NDAM in Table 4. Our evaluation shows that for L1 Russian and L1 Spanish, our adapted system S_{CONCAT} + $\text{NNJM}_{\text{ADAPTED}}$ significantly outperforms both TM_{INT} + $\text{NNJM}_{\text{BASELINE}}$ and S_{CONCAT} + NDAM ($p < 0.01$), but the improvement is not statistically significant for L1 Chinese.

Our evaluation also shows that the combination of TM_{INT} and adapted NNJM is similar (for L1 Chinese and Russian) or worse (for Spanish) in performance compared to S_{CONCAT} + $\text{NNJM}_{\text{ADAPTED}}$. This is because $\text{NNJM}_{\text{ADAPTED}}$ by itself is a translation model adaptation (because it considers source and target side contexts) and hence using TM_{INT} along with it does not bring in any newer information and may even hurt the performance when the in-domain data is very small (in the case of Spanish).

7.4 Effect of Adaptation Data

We also perform adaptation on the L1-specific FCE development set in Table 3 (which is also our development data for the GEC systems), instead of the in-domain data from Lang-8 in Table 2. Our neural network overfits easily on the FCE development set due to its much smaller size. Hence, we perform adaptive training for only 2 epochs on top of $\text{NNJM}_{\text{BASELINE}}$. The results are shown in the row S_{CONCAT} + $\text{NNJM}_{\text{ADAPTED (FCE)}}$ in Table 4. Although the FCE development data is much smaller in size than the L1-specific in-domain data from Lang-8, we observe similar improvements on all three L1s. This is likely due to the similarity of the development and test sets, which are obtained from the same FCE corpus. These experiments show that smaller high-quality L1-specific error annotated data (1,000–2,000 sentences) similar to the target data can be used for adaptation to give competitive results compared to using much larger in-domain data (20,000–250,000 sentences) from other sources.

We perform experiments with smaller general domain data. In order to do this, we select a subset of CONCAT composed of the in-domain data of the three L1s along with 300,000 sentences randomly selected from the rest of CONCAT. This corpus is referred to as SMALL-CONCAT (623,717 sentences and 7,990,659 source tokens). We perform both KL-divergence regularized NNJM adaptation ($\text{NNJM}_{\text{SMALL-ADAPTED}}$) as well as Neural Domain Adaptation Model (Joty et al., 2015) ($\text{NDAM}_{\text{SMALL}}$) and compare them to NNJM trained with SMALL-CONCAT ($\text{NNJM}_{\text{SMALL-BASELINE}}$). We use these NNJMs with our S_{CONCAT} baseline. The results are summarized in Table 4. When the ratio between in-domain data and general domain data is high, both adaptation methods do not significantly improve over an unadapted NNJM. In the case of L1 Spanish, KL-divergence regularized adaptation significantly outperforms the unadapted NNJM and NDAM as the size of in-domain data is smaller.

7.5 Effect of Regularization

To analyze the effect of regularization when smaller data are used, we vary the regularization weight λ in our objective function (Section 4). The results are shown in Figure 1. $\lambda = 0$ corresponds to no regularization and training completely depends on the in-domain data apart from using the general NNJM as a starting point. On the other hand, setting $\lambda = 1$ forces the distribution learnt by the network to be similar to that of the unadapted model. We see that having no regularization ($\lambda = 0$) fails on all three L1s, due to overfitting on the smaller in-domain data. However, the effect of varying regularization is more significant on L1 Russian and Spanish, as the general NNJM has seen much smaller in-domain

[3] We use the NDAM_{vl} (Joty et al., 2015) trained using the log likelihood objective function with self-normalization.

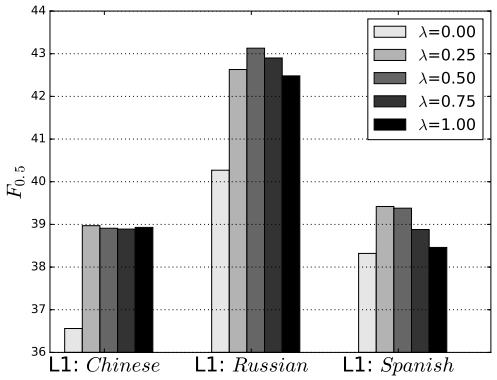

Figure 1: Effect of regularization for S_{CONCAT} + $NNJM_{ADAPTED\ (FCE)}$

data compared to L1 Chinese.

7.6 Evaluation on Benchmark Dataset

We also evaluate our system on the benchmark CoNLL-2014 shared task (Ng et al., 2014) test set for GEC in English. The CoNLL-2014 shared task consists of 1,312 sentences with two annotators. We also perform evaluation on the extension of CoNLL-2014 test set (Bryant and Ng, 2015), which contains eight additional sets of annotations over the two sets of annotations provided in the original test set. Following the settings of the CoNLL-2014 shared task, we tune our unadapted baseline system and the L1-adapted systems on the CoNLL-2013 shared task test set consisting of 1,381 test sentences. The results are summarized in Table 5.

We find that only the systems adapted based on L1 Chinese improves over the unadapted baseline system (S_{CONCAT} + $NNJM_{BASELINE}$). When the smaller-sized, high-quality FCE data is used for adaptation the margin of improvement is higher. This could be due to large proportion of Chinese learner written text in CoNLL-2014 test set, as the essays are written by the students of National University of Singapore comprising mostly of native Chinese speakers. Adaptation to L1 Russian and Spanish, does not help the system on CoNLL-2014 test set. We also compare our baseline SMT-based system with other state-of-the-art GEC systems. Our baseline system which is SMT-based, achieves the best $F_{0.5}$ score compared to other systems using the SMT approach alone, making it a competitive SMT-based GEC baseline. Overall, (Rozovskaya and Roth, 2016)

System	CoNLL-2014	
	ST	**10ANN**
S_{CONCAT} + $NNJM_{BASELINE}$	42.80	59.14
Adaptation based on L1 Chinese		
S_{CONCAT} + $NNJM_{ADAPTED}$	43.06	59.27
S_{CONCAT} + $NNJM_{ADAPTED\ (FCE)}$	44.27	60.36
Adaptation based on L1 Russian		
S_{CONCAT} + $NNJM_{ADAPTED}$	42.73	58.90
S_{CONCAT} + $NNJM_{ADAPTED\ (FCE)}$	42.12	58.30
Adaptation based on L1 Spanish		
S_{CONCAT} + $NNJM_{ADAPTED}$	41.88	58.32
S_{CONCAT} + $NNJM_{ADAPTED\ (FCE)}$	42.36	58.54
Best Published Results		
Rozovskaya and Roth (2016)		
(classifiers + spelling + SMT)	47.40	-
(SMT)	39.48	-
Chollampatt et al. (2016) (SMT)	41.75	57.19
Shared Task Teams		
CAMB (classifiers, rules, SMT)	37.33	54.26
CUUI (classifiers)	36.79	51.79
AMU (SMT)	35.01	50.17

Table 5: ST denotes $F_{0.5}$ scores on the shared task test set and **10ANN** denotes the $F_{0.5}$ scores on the extended test set with 10 sets of annotations.

achieves the best $F_{0.5}$ score (47.40) after adding classifier components, spelling checker, punctuation and capitalization correction components in a pipeline with their SMT-based system. However, their SMT-based system alone achieves an $F_{0.5}$ score of 39.48 only.

8 Discussion and Error Analysis

Our results show that L1-based adaptation of the NNJM using L1-specific in-domain data from Lang-8 significantly improves the $F_{0.5}$ score of the GEC system on the three L1s by 1.27 (Chinese), 0.88 (Russian), and 0.64 (Spanish). We observe similar gains when smaller in-domain development data from FCE is used for adaptation. These results show that adaptation based on L1 is beneficial for targeted error correction based on the native language of the writers. Our results also show that the proposed method of NNJM adaptation is scalable to different sizes of in-domain and general domain data and outperforms other methods of adaptation like phrase table interpolation (Sennrich, 2012) and Neural Domain Adaptation Model (NDAM) (Joty et al., 2015).

We perform error analysis on four error types

Error type	Δ $F_{0.5}$		
	Chinese	*Russian*	*Spanish*
verb form/tense	+0.394	+0.298	-0.124
determiner	**+2.892**	**+2.440**	+1.648
preposition	+0.084	+2.010	**+1.806**
noun number	+0.130	-0.706	+0.822
all	+0.400	+1.068	+0.586

Table 6: Differences between per error type $F_{0.5}$ scores of *system* and *baseline* for the three L1s

which are difficult for non-native learners of English.

We compare the outputs produced by our adapted *system*: S_{CONCAT} + $NNJM_{ADAPTED}$ and the *baseline*: S_{CONCAT} + $NNJM_{BASELINE}$. We perform per error type quantitative analysis of our results by observing the difference in the per error type $F_{0.5}$ scores averaged over five runs of tuning and evaluation of *baseline* and *system*. Computing per error type $F_{0.5}$ scores is difficult for SMT-based GEC systems, as the error types for corrections proposed by the SMT-based GEC system cannot be determined trivially. To overcome this difficulty, we attempt to determine the error type of the proposed corrections by matching them to the available human annotations (the source/target phrase without the surrounding context) in the complete FCE dataset (1,244 scripts). We select those sentences from the test data where the error type of every correction proposed by the *baseline* and the *system* can be determined. This process selects 928, 1102, and 2179 sentences for L1 Chinese, Russian, and Spanish, respectively. The differences in per error type $F_{0.5}$ scores between *system* and *baseline* are shown in Table 6. For Chinese, the largest gain in $F_{0.5}$ is observed for determiner errors. Determiner errors are frequent in our L1 Chinese FCE test set (10.02%) . Moreover, we see that adaptation improves verb form errors by approximately 0.4% $F_{0.5}$. Verb form errors are the most frequent error type in our L1 Chinese test set (14.46%). Also, the highest gain for L1 Russian comes from determiner errors which is the most frequent error type in our FCE test data for L1 Russian (13.55%). Similarly, the highest gain for L1 Spanish comes from preposition errors which is the most frequent error type for L1 Spanish (12.51%).

From a practical standpoint, the adapted system can be used as an educational aid in English classes of local students in non-English-speaking countries, where all the students share the same L1 and their L1 is known in advance. The adapted system can give focused feedback to learners by correcting mistakes frequently made by learners having the same L1. Also, NNJM adaptation proposed in this paper can be done using a small number of essays (50–100 essays) in a relatively short time (20–30 minutes), making it easy to adapt GEC systems in practice.

9 Conclusion

In this paper, we perform NNJM adaptation using L1-specific learner text with a KL divergence regularized objective function. We integrate adaptation into an SMT-based GEC system. The systems with adapted NNJMs outperform unadapted baselines significantly. We also demonstrate the necessity for regularization when adapting on smaller in-domain data. Our method of adaptation performs better compared to other adaptation methods, especially when smaller in-domain data is used. Our results show that adapting GEC systems for learners of similar L1 background gives significant improvement and can be adopted in practice to improve GEC systems.

Acknowledgments

We thank Kaveh Taghipour for insightful comments and discussions throughout this work. We are also grateful to the anonymous reviewers for their feedback which helped in revising and improving the paper. This research is supported by Singapore Ministry of Education Academic Research Fund Tier 2 grant MOE2013-T2-1-150.

References

James Bergstra, Olivier Breuleux, Frédéric Bastien, Pascal Lamblin, Razvan Pascanu, Guillaume Desjardins, Joseph Turian, David Warde-Farley, and Yoshua Bengio. 2010. Theano: a CPU and GPU math expression compiler. In *Proceedings of the Python for Scientific Computing Conference*.

Yevgeni Berzak, Roi Reichart, and Boris Katz. 2014. Reconstructing native language typology from foreign language usage. In *Proceedings of the 19th Conference on Computational Natural Language Learning*.

Yevgeni Berzak, Roi Reichart, and Boris Katz. 2015. Contrastive analysis with predictive power: Typology

driven estimation of grammatical error distributions in ESL. In *Proceedings of the 19th Conference on Computational Natural Language Learning*.

Christopher Bryant and Hwee Tou Ng. 2015. How far are we from fully automatic high quality grammatical error correction? In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*.

Shamil Chollampatt, Kaveh Taghipour, and Hwee Tou Ng. 2016. Neural network translation models for grammatical error correction. In *Proceedings of the 25th International Joint Conference on Artificial Intelligence*.

Jonathan H Clark, Chris Dyer, Alon Lavie, and Noah A Smith. 2011. Better hypothesis testing for statistical machine translation: Controlling for optimizer instability. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*.

Daniel Dahlmeier and Hwee Tou Ng. 2011. Correcting semantic collocation errors with L1-induced paraphrases. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*.

Daniel Dahlmeier and Hwee Tou Ng. 2012. Better evaluation for grammatical error correction. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.

Daniel Dahlmeier, Hwee Tou Ng, and Eric Jun Feng Ng. 2012. NUS at the HOO 2012 shared task. In *Proceedings of the 7th Workshop on the Innovative Use of NLP for Building Educational Applications*.

Daniel Dahlmeier, Hwee Tou Ng, and Siew Mei Wu. 2013. Building a large annotated corpus of learner English: The NUS corpus of learner English. In *Proceedings of the Eighth Workshop on Innovative Use of NLP for Building Educational Applications*.

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*.

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 Seventh Workshop on Building Educational Applications Using NLP*.

Jacob Devlin, Rabih Zbib, Zhongqiang Huang, Thomas Lamar, Richard Schwartz, and John Makhoul. 2014. Fast and robust neural network joint models for statistical machine translation. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*.

George Foster and Roland Kuhn. 2007. Mixture-model adaptation for SMT. In *Proceedings of the Second Workshop on Statistical Machine Translation*.

Kenneth Heafield, Ivan Pouzyrevsky, Jonathan H. Clark, and Philipp Koehn. 2013. Scalable modified Kneser-Ney language model estimation. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*.

Duc Tam Hoang, Shamil Chollampatt, and Hwee Tou Ng. 2016. Exploiting n-best hypotheses to improve an SMT approach to grammatical error correction. In *Proceedings of the 25th International Joint Conference on Artificial Intelligence*.

Scott Jarvis and Terence Odlin. 2000. Morphological type, spatial reference, and language transfer. *Studies in Second Language Acquisition*, 22:535–556.

Shafiq Joty, Hassan Sajjad, Nadir Durrani, Kamla Al-Mannai, Ahmed Abdelali, and Stephan Vogel. 2015. How to avoid unwanted pregnancies: Domain adaptation using neural network models. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*.

Marcin Junczys-Dowmunt and Roman Grundkiewicz. 2014. The AMU system in the CoNLL-2014 shared task: Grammatical error correction by data-intensive and feature-rich statistical machine translation. In *Proceedings of the Eighteenth Conference on Computational Natural Language Learning: Shared Task*.

Philipp Koehn and Josh Schroeder. 2007. Experiments in domain adaptation for statistical machine translation. In *Proceedings of the Second Workshop on Statistical Machine Translation*.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondrej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics (Interactive Poster and Demonstration Sessions)*.

Yann LeCun, Leon Bottou, Genevieve Orr, and Klaus Müller. 1998. Efficient backprop. *Neural Networks: Tricks of the Trade*, pages 9–50.

Marco Lui and Timothy Baldwin. 2011. Cross-domain feature selection for language identification. In *Proceedings of the 5th International Joint Conference on Natural Language Processing*.

Sean Massung and Chengxiang Zhai. 2016. Non-native text analysis: A survey. *Natural Language Engineering*, 22:163–186.

Tomoya Mizumoto, Mamoru Komachi, Masaaki Nagata, and Yuji Matsumoto. 2011. Mining revision log of

language learning SNS for automated Japanese error correction of second language learners. In *Proceedings of the Fifth International Joint Conference on Natural Language Processing*.

Robert C. Moore and William Lewis. 2010. Intelligent selection of language model training data. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*.

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*.

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*.

Lourdes Ortega. 2009. *Understanding Second Language Acquisition*. Hodder Education.

Alla Rozovskaya and Dan Roth. 2010. Generating confusion sets for context-sensitive error correction. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*.

Alla Rozovskaya and Dan Roth. 2011. Algorithm selection and model adaptation for ESL correction tasks. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics*.

Alla Rozovskaya and Dan Roth. 2016. Grammatical error correction: Machine translation and classifiers. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*.

Alla Rozovskaya, Kai-Wei Chang, Mark Sammons, Dan Roth, and Nizar Habash. 2014. The Illinois-Columbia system in the CoNLL-2014 shared task. In *Proceedings of the Eighteenth Conference on Computational Natural Language Learning: Shared Task*.

Rico Sennrich. 2012. Perplexity minimization for translation model domain adaptation in statistical machine translation. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*.

Mina Shaughnessy. 1977. *Errors and Expectations*. New York: Oxford University Press.

Raymond Hendy Susanto, Peter Phandi, and Hwee Tou Ng. 2014. System combination for grammatical error correction. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*.

Helen Yannakoudakis, Ted Briscoe, and Ben Medlock. 2011. 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*.

Dong Yu, Kaisheng Yao, Hang Su, Gang Li, and Frank Seide. 2013. KL-divergence regularized deep neural network adaptation for improved large vocabulary speech recognition. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*.

Zheng Yuan and Ted Briscoe. 2016. Grammatical error correction using neural machine translation. In *Proceedings of the 15th Annual Conference of the North American Chapter of the Association for Computational Linguistics*.