A POS Tagging Model Designed for Learner English

Ryo Nagata[1,2], **Tomoya Mizumoto**[3], **Yuta Kikuchi**[4],
Yoshifumi Kawasaki[5], and **Kotaro Funakoshi**[6]

[1] Konan University / 8-9-1 Okamoto, Kobe, Hyogo 658-8501, Japan
[2] JST, PRESTO / 4-1-8 Honcho, Kawaguchi, Saitama 332-0012, Japan
[3] RIKEN AIP / 2-1 Hirosawa, Wako, Saitama 351-0198, Japan
[4] Preferred Networks, Inc. / 1-6-1 Otemachi, Chiyoda-ku, Tokyo 100-0004, Japan
[5] University of Tokyo / 3-8-1 Komaba, Meguro-ku, Tokyo 153-8902, Japan
[6] Kyoto University / Yoshida-honmachi, Sakyo-ku, Kyoto 606-8501, Japan

Abstract

There has been very limited work on the adaptation of Part-Of-Speech (POS) tagging to learner English despite the fact that POS tagging is widely used in related tasks. In this paper, we explore how we can adapt POS tagging to learner English efficiently and effectively. Based on the discussion of possible causes of POS tagging errors in learner English, we show that deep neural models are particularly suitable for this. Considering the previous findings and the discussion, we introduce the design of our model based on bidirectional Long Short-Term Memory. In addition, we describe how to adapt it to a wide variety of native languages (potentially, hundreds of them). In the evaluation section, we empirically show that it is effective for POS tagging in learner English, achieving an accuracy of 0.964, which significantly outperforms the state-of-the-art POS-tagger. We further investigate the tagging results in detail, revealing which part of the model design does or does not improve the performance.

1 Introduction

Although Part-Of-Speech (POS) tagging is widely used in Natural Language Processing (NLP), there has been little work on its adaptation to learner English[1]. It is often done by simply adding a manually-POS-tagged learner corpus to the training data (Nagata et al., 2011; Berzak et al., 2016). Probably only one exception is the work by Sakaguchi et al. (2012) who proposed to solve POS tagging and spelling error correction simultaneously. However, their method also requires a POS-annotated learner as training data. The availability of POS-labeled learner corpora is still very limited even after the efforts researchers (e.g., Díaz-Negrillo et al. (2009); van Rooy and Schäfer (2002); Foster (2007b,a); Nagata et al. (2011); Berzak et al. (2016)) have made. Because of this limitation, POS taggers designed for canonical English (i.e., native English) are normally used in related tasks including grammatical error correction (Leacock et al., 2010) and its automated evaluation (Bryant et al., 2017), automated essay scoring (Burstein et al., 1998), and analyses of learner English (Aarts and Granger, 1998; Tono, 2000), to name a few.

Unfortunately, however, the discrepancy between a POS tagger and its target text often results in POS-tagging errors, which in turn leads to performance degradation in related tasks as Nagata and Kawai (2011) and Bryant et al. (2017) show. Specifically, a wide variety of characteristic phenomena that potentially degrade POS tagging performance appear in learner English. Section 2 shows that there exist a wide variety of potential causes of POS-tagging errors. For the time being, let us consider the following erroneous sentence:

(1) ***Becose/NNP I/CD like/IN** reading/NN ,/, I/PRP want/VBP many/JJ **Books/NNPS** ./.[2]

where mistakenly-tagged tokens are written in bold type[3]. It reveals that several POS-tagging errors occur because of orthographic and grammatical errors. Besides, Nagata and Whittaker (2013) and Berzak et al. (2014) demonstrate that learner English exhibits characteristic POS sequence patterns depending on the writers' native languages. All these phenomena suggest that the adaptation of

[1] In this paper, *learner English* refers to English as a foreign language.

[2] In this paper, the asterisk * denotes that the following sentence is erroneous.

[3] Stanford CoreNLP 3.8.0 (Manning et al., 2014) was used to tag the sentence. It is only natural that a POS tagger for canonical English should make errors as in this example because they do not simply assume erroneous or unnatural inputs.

Proceedings of the 2018 EMNLP Workshop W-NUT: The 4th Workshop on Noisy User-generated Text, pages 39–48
Brussels, Belgium, Nov 1, 2018. ©2018 Association for Computational Linguistics

POS tagging to learner English will reduce their influence and thus contribute to achieving better performance in the related tasks.

In view of this background, in this paper, we explore how we can adapt POS tagging to learner English effectively. We first discuss potential causes of POS-tagging errors in learner English. Based on this, we then describe how deep neural models, which have been successfully applied to sequence labeling (Huang et al., 2015; Ma and Hovy, 2016; Plank et al., 2016), are particularly suitable for our purpose. Considering the previous findings and our discussion on the possible causes of POS-tagging errors, we present the design of our model based on Long Short-Term Memory (Hochreiter and Schmidhuber, 1997) (LSTM). Our model is equipped with a word token-based and character-based bidirectional LSTMs (BLSTMs) whose inputs are respectively word embeddings and character embeddings obtained from learner corpora. In addition, we describe how to adapt it to a wide variety of native languages (potentially, hundreds of them) through native language vectors. In the evaluation section, we empirically show that it is effective in adapting POS tagging to learner English, achieving an accuracy of 0.964 on Treebank of Learner English (TLE; (Berzak et al., 2016)), which is significantly better than that of Turbo tagger (Martins et al., 2013), one of the state-of-the-art POS taggers for native English. We further investigate the tagging results in detail, revealing why the word token-based and character-based BLSTMs contribute to improving the performance while native language vectors do not.

2 Potential Causes of POS-Tagging Errors in Learner English

In general, a major cause of POS-tagging errors is *unknown words*. Here, unknown words refer to those that have not appeared in the training data (i.e., a POS-labeled corpus). It would often be difficult to recognize the POS label of an unknown word (Mizumoto and Nagata, 2017).

A frequent source of unknown words in learner English is spelling errors. They rarely (or never) appear in well-edited texts such as newspaper articles that are normally used to train a POS tagger. This means that they almost always become unknown words to a POS tagger trained on canonical English. For instance, the misspelt token *Becose* in Ex. (1) in Sect. 1 is mistakenly recognized as a proper noun. Interestingly, it causes further tagging errors in the following two tokens (i.e., *I/CD like/IN*). Similarly, errors in (upper/lower) cases affect POS tagging as can be seen in *Books/NNPS* in the same example. Considering this, the key to success in POS tagging for learner English is how to reduce the influence from these orthographic errors.

Note that most POS-tagging guidelines for learner English such as Ragheb and Dickinson (2012), Nagata et al. (2011), and Berzak et al. (2016) stipulate that a token with an orthographic error should receive the POS label that is given to the corresponding correct spelling. Accordingly, it is preferable that POS taggers for learner English should do the same.

Foreign words such as foreign proper names, which is another source of unknown words, often appear in learner English. They are sometimes not translated into English but transliterated as in *onigiri* meaning *rice ball*.

Grammatical errors also affect POS tagging. They are often classified into three types as shown in Izumi et al. (2004): insertion, omission, and replacement types. All of them may cause a POS tagging error as follows:

(2) Insertion:
You/PRP must/MD be/VB **feel/JJ** sad/JJ

(3) Omission:
_ **Flower/NNP** is/VBZ beautiful/JJ

(4) Replacement:
There/EX are/BP **differents/NNS** topics/NNS

Here, erroneous tokens are underlined whereas mistakenly-tagged tokens are written in bold type. These examples show that grammatical errors cause POS-tagging errors not only to the erroneous tokens themselves but also to their surroundings. In Ex. (2), the verb *be* is erroneously inserted after the word *must*, which causes the POS tagging error *feel/JJ*. In Ex. (3), a word is missing at the beginning (probably, *The*). This results in the upper case *Flower* and thus leads to its POS-tagging error as a proper noun. In Ex. (4), the word *differents* erroneously agrees with *topics* in number and should be replaced with the correct form *different*[4]. Because of the pseudo plural suffix in the adjective, it is mistakenly recognized as a plural noun.

Again, most existing POS-tagging guidelines for learner English state that a token with a grammatical error should primarily be tagged based on its superficial information[5]. For example, according to the guidelines Dickinson and Ragheb (2009), Nagata et al. (2011), and Berzak et al. (2016), the above three examples should be tagged as:

(5) *You/PRP must/MD be/VB feel/VB sad/JJ

(6) *Flower/NN is/VBZ beautiful/JJ

(7) *There/EX are/BP differents/JJ topics/NNS

While not errors, the differences in POS distribution might cause POS-tagging errors. For example, Chodorow and Leacock (2002) report that the word *concentrate* is mostly used as a noun (as in *orange juice concentrate*) in newspaper articles while as a verb (as in *He concentrated*) in learner English. The differences are reflected in the training data (native English) and the target text (learner English). This might cause POS-tagging errors even in correct sentences written by learners of English.

Related to this are the differences in POS sequence patterns in learner English. As already mentioned in Sect. 1, learner English exhibits characteristic POS sequence patterns depending on the writers' native languages. In other words, every group of a native language has its own POS sequence distribution, which might affect POS tagging just as the biased POS distributions do. Besides, similar native languages show similar POS patterns in English writing (Nagata and Whittaker, 2013; Berzak et al., 2014). This implies that the adaptation of POS tagging to the writers' native languages will likely contribute to extra improvement in tagging performance.

3 POS-Tagging Model for Learner English

3.1 Whole Architecture

Figure 1 shows the architecture of our POS-tagging model. Its input is the information about each word in the sentence in question and the writer's native language. It is transformed into vectors by the embedding layer. The resulting vectors are passed on to the BLSTM layer. Each LSTM block corresponds to each word vector. This enables the entire model to consider all surrounding words together with the target word to determine its POS label, which is effective in POS-tagging in general as shown in (Huang et al., 2015; Ma and Hovy, 2016). The outputs of the BLSTM layer are fed into the softmax layer to predict their corresponding POS labels[6].

The embedding layer is equipped with three modules (network layers) to handle linguistic phenomena particular to learner English. They are shown in the lower part of Fig. 1. The first and second ones encode the information from the word token itself and its characters, respectively; they will be referred to as word token-based and character-based modules, respectively. The third one is for the adaptation to the writer's native language, which will be referred to as the native language-based module.

The outputs of the three modules are put together as the input of the upstream BLSTM layer. Simply, all three vectors from the three modules are concatenated to form a vector. The resulting vector corresponds to a word in the target sentence in question. The concatenation of word and character embedding vectors represents the word by means of its word token and characters. Then, its concatenation to the native language embedding vector maps it onto another vector space that considers native languages. In other words, the resulting vectors represent native language specific words. They are in turn propagated through the BLSTM layer, which realizes the native language specific POS tagging.

[4] It is known that this kind of error occurs in the writing of learners whose native language has the adjective-noun agreement (e.g., Spanish) (Swan and Smith, 2001).

[5] POS labels based on distributional information can also be included by using the multiple layer scheme (Díaz-Negrillo et al., 2009; Dickinson and Ragheb, 2009; Nagata et al., 2011; Berzak et al., 2016). It depends on the user which layer to use. In either case, the presented model (and also most existing ones) can be trained with the given tagset.

[6] One could apply BLSTM-CRF as in (Huang et al., 2015) to this task. However, Huang et al. (2015) show that BLSTM performs equally to or even better than BLSTM-CRF in POS tagging. Considering this, we select BLSTM, which is simpler and thus faster to train.

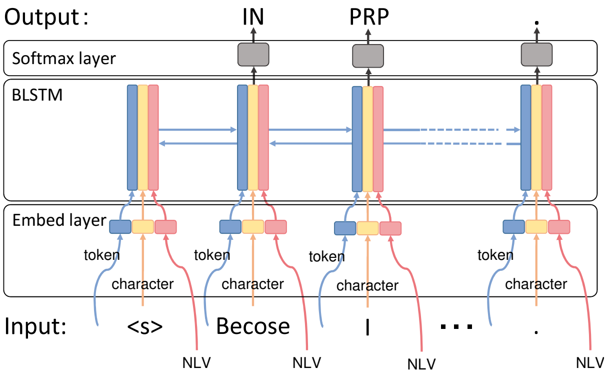

Figure 1: Structure of POS-Tagging Model: NLV stands for native language vector.

3.2 Word Token-based Module

The word token-based module consists of a word-embedding layer. Namely, it takes as input a word ID and returns its corresponding word-embedding vector; note that all words are converted into lowercase in order to reduce the vocabulary size. In this representation, (syntactically and semantically) similar words tend to have similar vectors. This property of word embeddings is particularly important to cope with unknown words including orthographic errors found in learner English. Alikaniotis et al. (2016) show that word embeddings place misspelt words and their correct spelling closer in the vector space; Figure 2 exemplifies this situation for the words *because* and *being* and their misspellings. As in this example, misspelt words can be treated as similar words through word embeddings. Consequently, they will likely be recognized to have the same POS label as the correct word does even when they do not appear in the training data.

Here, it should be emphasized that the word-embedding layer (or precisely, its weights) can be pre-trained without POS-labeled corpora. Because it simply requires an unlabeled corpus, even learner English corpora, which are now widely available[7], can be used to obtain word embeddings. Furthermore, the target learner texts themselves can also be used for the same purpose. This is especially beneficial for applications such as automated essay scoring for language proficiency tests or learner English analyses where a set of texts are available at a time and one can take some time to process them. Doing so, words that do not normally appear in native English such as orthographic errors and foreign proper names are naturally reflected in the POS-tagging model.

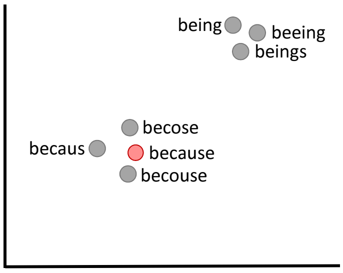

Figure 2: Words Mapped onto a Vector Space.

3.3 Character-based Module

The character-based module, which comprises a character-embedding layer and a BLSTM layer as shown in Fig. 3, augments the word token-based module. Although the latter is crucial for handling linguistic phenomena particular to learner English as just explained, it becomes less effective against low-frequency words; it is often difficult to obtain reliable word embeddings for them.

To overcome this, in the character-based module, each character in the word in question is passed on to the character-embedding layer and then to the BLSTM layer. With this module, any word can be encoded into a vector unless it includes unseen characters, which is normally not the case. Similar characters should receive similar character embedding vectors from the character-embedding layer. Likewise, words that have similar spellings are expected to receive similar vectors from the BLSTM layer. Thus, the resulting vectors are expected to absorb the influence from deletion (e.g., *Becuse*), insertion (e.g., *Beccause*), substitution (e.g., *Becouse*), and transposition (e.g., *Becuase*). Because of this property, low-frequency or even unseen orthographic errors will likely be recognized by the character-based module.

Note that the character-embedding layer can also be pre-trained with unlabeled native and learner English corpora. Also note that unlike the word embedding layer, all characters are NOT converted into lower case in the character-embedding layer in order to capture the differences in upper/lower cases.

3.4 Native Language-based Module

The final component is the native language-based module. Native languages can also be encoded into vectors by an embedding layer. Namely, a native language-embedding layer takes as input

[7] It is POS-labeled learner corpora that are still rare.

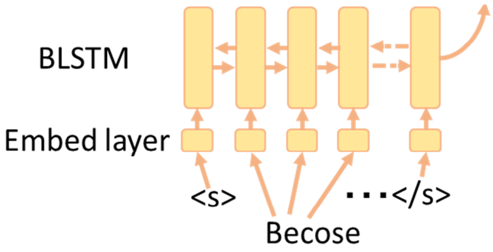

Figure 3: Character-based Module.

a native language ID and returns its corresponding native language embedding vector. Just as in word and character embeddings, similar native languages are expected to have similar vectors. Accordingly, even if there is no training data for a certain native language (say, Spanish), the POS-tagging model can benefit from other training data whose native language is similar to it (say, French or Italian). Fortunately, pre-trained language vectors are already available in other NLP tasks as in (Östling and Tiedemann, 2017; Malaviya et al., 2017). For example, Malaviya et al. (2017) propose a method for learning language vectors through word token-based neural machine translation (many source languages to English). Their resulting vectors covering over 1,000 languages are available on the web. We use these language vectors in the native language-based module. All weights in the native language-embedding layer are fixed during the training, and they are transformed by a linear layer to adjust their values so that they become informative for POS-tagging.

4 POS-Labeled Learner Corpora

A POS-labeled learner corpus is required to train and test the presented model. We use the following two learner corpora as our target: Treebank of Learner English (TLE; (Berzak et al., 2016)) and Konan-JIEM learner corpus (KJ; (Nagata et al., 2011)). Their statistics are shown in Table 1.

TLE is a learner corpus annotated with Universal POS, Penn Treebank POS, and dependency. It also contains information about the writers' native language which ranges over 10 languages (Chinese, French, German, Italian, Japanese, Korean, Portuguese, Spanish, Russian, and Turkish). It is already split into training, development, and test sets for evaluation. Besides, Berzak et al. (2016) report on accuracy of the Turbo tagger (Martins et al., 2013), which is one of the state-of-the-art POS taggers. All these properties are beneficial to POS-tagging evaluation. For consistency with the other learner corpus, only Penn Treebank POS is considered in this paper.

KJ is annotated with phrase structures (Nagata and Sakaguchi, 2016), which is based on the Penn Treebank POS and bracketing guidelines (Santorini, 1990; Bies et al., 1995). It consists of complete essays (whereas TLE only contains sampled sentences). The writers are Japanese learners of English. It provides both superficial and distributional POS tags[8] Its advantage is that spelling errors are manually annotated with their correct forms. This allows us to investigate how well POS-tagging models overcome the influence from spelling errors.

5 Performance Evaluation

5.1 Evaluation Settings

The presented model was first tested on KJ to investigate how well it performs on learner English without a POS-labeled learner corpus (and therefore without the native language-based module). It was trained on sections 00–18 of Penn Treebank Wall Street Journal (WSJ[9]). Its hyperparameters, including the number of learning epochs, were determined using the development set of TLE[10]. The following native and learner corpora were used to obtain word and character embeddings: English Web Treebank (EWT[11]), an in-house English textbook corpus, International Corpus of Learner English (ICLE; (Granger et al., 2009)), ETS Corpus of Non-Native Written English (ETS[12]), Nagoya

[8] Two POS labels are sometimes given to a word in learner English, depending on its superficial information (word form) and distributional information (surrounding words surrounding). For example, in *I went swimming in the see*, the word *see* can be interpreted as a verb from its form and also as a noun from its context. The former and latter are referred to as superficial and distributional POS tags, respectively.

[9] Marcus, Mitchell, et al. Treebank-3. Linguistic Data Consortium, 1999.

[10] The maximum number of epochs was set to 20 and the one that maximized the performance on the development set was chosen. The other hyperparameters were determined as follows: The dimensions of word and character embeddings: 200 and 50. The native language module consisted of an embedding layer of dimension 512 and a linear layer of dimension 200. The dropout rate for BLSTM was 0.5; Adam was used for optimization (step size: 0.01, the first and second moment: 0.9 and 0.999, respectively).

[11] Bies, Ann, et al. English Web Treebank. Linguistic Data Consortium, 2012.

[12] `https://catalog.ldc.upenn.edu/LDC2014T06`

Corpus	# sentences	# tokens
TLE train	4,124	78,541
TLE development	500	9,549
TLE test	500	9,591
KJ	3,260	30,517

Table 1: Statistics on Target Learner Corpora.

Interlanguage Corpus of English (NICE[13]), Lang-8 corpus of Learner English[14]. The Word2vec software[15] was used to produce word and character embeddings. The hyperparameters were determined by using the TLE development set as follows: The dimensions of word and character embeddings were 200 and 50, respectively; the window size was set to five in both cases; the other hyperparameters were set as shown in the footnote. Performance was measured by accuracy.

For comparison, a Conditional Random Field (Lafferty et al., 2001) (CRF)-based method was implemented using the same training and development sets. The features are: superficial, lemma, prefix, and suffix of tokens and presence of specific characters (numbers, uppercase, and symbols) with a window size of five tokens to the left and right of the token in question. The first-order Markov model features were used to encode inter-label dependencies.

The presented model was then tested on TLE with the same evaluation settings as in Berzak et al. (2016) which reports on the performance of the Turbo Tagger. It was trained on the data consisting of the training portions of TLE and EWT. Its hyperparameters, including the number of learning epochs, were determined using the development set of TLE. Word and character embeddings were obtained from the same corpora above. The resulting model was tested on the test portion of TLE.

5.2 Results

Table 2 shows POS-tagging accuracy of our model and the CRF-based method on KJ both for superficial and distributional POS. It includes the results where all spelling errors were fully corrected to investigate the influence from spelling errors. It reveals that the presented model without the information about correct spellings outperforms even the CRF-based method fully exploiting the information; the differences between our model with original spellings and the CRF-based method with correct spellings are statistically significant both in surface (at the 99% confidence level) and distributional POS accuracy (at the 95% confidence level) (test for difference in population portion). This shows that the presented model successfully absorbs the influence from spelling errors and also other linguistic phenomena particular to learner English.

Table 3 shows the results on TLE including accuracy of the Turbo Tagger, which is cited from the work (Berzak et al., 2016). It shows that the presented model outperforms the Turbo Tagger; the difference is statistically significant at the 99% confidence level (test for difference in population proportion). Note that the Turbo Tagger was trained on the same POS-labeled native and learner corpus as in the presented model. Nevertheless it does not perform as well as the presented models.

Contrary to our expectation, the presented models with and without the native language-based module perform almost equally well. In other words, the adaptation to native languages by means of the native language-based module is not more effective than the simple addition of the learner data to the training data.

The evaluation results are summarized as follows: The presented model performs successfully on learner data even without a POS-labeled learner corpus. It seems to absorb the influence from spelling errors and other learner language-specific phenomena. By contrast, the direct adaptation to learner English through native language vectors is not effective. The next section will explore the reasons for these observations.

6 Discussion

To investigate where the presented model improved accuracy even without a POS-labeled learner corpus, we compared the POS-tagging results of the three methods (the presented model and two CRF-based methods).

Almost immediately, we found that spelling errors were one of the major reasons, as expected. Examples include*famouse*, *exsample*, *thier*, *waching*, and *exiting*, to name a few. All these appeared

[13] `http://sgr.gsid.nagoya-u.ac.jp/wordpress/?page_id=695`

[14] `http://cl.naist.jp/nldata/lang-8/`

[15] `https://github.com/dav/word2vec`. The options are: -negative 25 -sample 1e-4 -iter 15 -cbow 1 -min-count 5

	Superficial POS accuracy		Distributional POS accuracy	
Model	Original spelling	Correct spelling	Original spelling	Correct spelling
Our model	0.948	0.949	0.945	0.948
CRF	0.940	0.942	0.939	0.941

Table 2: Accuracy on KJ: All models are trained on sections 00-18 of WSJ.

Model	Accuracy
W/ annotated learner corpus (training data: EWT TLE train)	
Our model (with native language module)	0.964
Our model (without native language module)	0.963
Turbo Tagger (Berzak et al., 2016) (with annotated learner corpus)	0.958
W/o annotated learner corpus (training data: EWT)	
Our model	0.951
Turbo Tagger (Berzak et al., 2016)	0.943

Table 3: Accuracy on TLE Test Set.

in the unlabeled learner corpora and their word embeddings were available.

Looking into spelling errors revealed that there were cases where their word embeddings were not available. They often showed more severe spelling formations (in terms of edit distance and/or the probability of character replacements) and thus tend to be less frequent; note that words whose frequencies were less than the threshold (five in the evaluation) in the training data were excluded from word embeddings. For instance, *ranchi* (correctly, *lunch/NN*), *dilicuse* (*delicious/JJ*), and *beutifure* (*beautifull/JJ*) appeared less than five times in the training data and thus no word embeddings were available for them. Nevertheless, the presented model successfully predicted their POS labels. Quantitatively, the performance difference between the presented model with and without the character-based module is an accuracy of 1.0%. In contrast, because their affix gave less or zero information about their POS labels (or even wrong information in some cases), the CRF-based method failed with them[16]. These observations show that the character-based module is effective in analyzing misspelt words.

These analyses confirm that pre-training of word token-based and character-based modules is crucial to achieve better performance. With pre-training, the presented model can gain an accuracy of 0.3% to 0.6% depending on the training conditions. When a POS-annotated learner corpus is not available for training, the gain is relatively large. Even when it is available, their pre-training augment the presented model to some extent.

Our models were also robust against the influence from the differences in POS distributions between learner and native English. For example, the majority (82%) of the word *like* appeared as a preposition in the native corpus (WSJ) whereas only 5% were as a preposition and 94% were as a verb in KJ. The difference in the POS distribution often caused failures in the CRF-based method. To be precise, it only achieved an accuracy of 0.635 for the 304 instances of *like* in KJ. In contrast, the presented model achieved a much better accuracy of 0.927 for them, which suggests that the use as a verb was reflected in its word embedding vector by pre-training. We observed a similar tendency for the word *fight* as a verb and a noun.

To our surprise, the word token-based and character-based modules absorbed the influence from grammatical errors to some extent. In particular, they contributed to successfully predict POS labels for sentences containing missing determiners, especially at the beginning (as in Ex.(3) in Sect. 2). If the determiner at sentence beginning is missing, the following word begins with an uppercase letter as in $\textit{The flower is} \cdots \rightarrow \textit{*Flower is} \cdots$. In native English the word *flower* is normally used as a countable noun and thus rarely appears with a bare form (without determiner and in singular form), which makes the usage (i.e., *Flower*) unknown to methods trained on native English data. Accordingly, the CRF-based method tends to mis-

[16] At the same time, its overall performance decreases by 1% without information about the affix.

takenly interpret countable nouns appearing at the beginning of a sentence as proper nouns. In contrast, such bare forms often appear in learner English because of frequent missing determiner errors as in **Flower is* $\cdots$. Also, in learner English, a sentence sometimes begins with a lowercase letter as in **a flower is* $\cdots$ or words other than proper nouns begin with an uppercase letter in the middle as in **A Flower is* $\cdots$. These erroneous occurrences of uppercase and lowercase letters are reflected in the character-based modules, which make the entire model less sensitive to the uppercase/lowercase difference. Besides, the fact that countable nouns often appear in learner English with no determiner and in singular form, which is observed superficially as uncountable, is reflected in the word embeddings. The resulting word embeddings tend to treat countable nouns as more like uncountable nouns. Consequently, they tend to interpret a singular countable noun with no determiner at sentence beginning as a common noun; note that uncountable singular nouns can appear with no determiner as in *Water is abundant* even in canonical English. Similar cases such as **Mountain is beautiful.* and **Town has liblary.* are often found in the evaluation data.

The effect on word order errors was also observable. The presented model successfully POS-tagged sentences containing word order errors as in **My hobby is abroad/RB travel* and **I very/RB enjoyed*. Part of the reason is that the model abstractly encodes information about word sequences through the main BLSTM; it is expected that this property of the presented model makes it robust against unusual word order to some extent. In contrast, completely different features are given to a sequence of two words and its reversal in the CRF-based method, which makes it more sensitive to word order.

Finally, let us discuss the native language-based module. Unlike the other two modules, it has little or no effect on POS tagging for learner English.

To reconfirm this, we conducted an additional experiment where we held out the data of one native language in TLE at a time, trained the presented model on the rest of the data (together with the native data), and tested it on the held-out data (i.e., leave-one-native-language-out cross validation). The results reconfirmed that the native-language module had almost no effect. To be precise, the presented model with and without the native language module achieved an accuracy of 0.966 and 0.965, respectively; if the native language-based module had been really effective and had been able to exploit training data of similar native languages, the performance difference would have been larger in this setting.

Possible reasons for this are: (i) the size of training and/or test data is not large enough to show its effectiveness; (ii) phenomena common to all or most native languages are dominant (in other words, phenomena specific to a certain group of native languages are infrequent); (iii) even the presented model without the native-language module automatically and abstractly recognizes the writer's native language in the course of POS tagging (for example, by some of the units in the main BLSTM). Further investigation is required to determine which explanation is correct.

7 Conclusions

In this paper, we have explored the adaptation of POS tagging to learner English. First, we discussed possible causes of POS-tagging failures. Then, we proposed a POS-tagging model consisting of BLSTMs based on word, character, and native language embeddings. We showed that it achieved an accuracy of 0.948 and 0.964 on KJ and TLE, respectively, which significantly outperformed methods for comparison. Finally, we empirically showed where and why the word token-based and character-based modules were effective in POS tagging for learner English.

For future work, we will investigate why the native language-based module does not perform well, which is contrary to our expectation. We will also investigate how we can effectively adapt POS-tagging models to native languages.

References

Jan Aarts and Sylviane Granger. 1998. *Tag sequences in learner corpora: a key to interlanguage grammar and discourse*. Longman, New York.

Dimitrios Alikaniotis, Helen Yannakoudakis, and Marek Rei. 2016. Automatic text scoring using neural networks. In *Proc. of 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 715–725.

Yevgeni Berzak, Jessica Kenney, Carolyn Spadine, Jing Xian Wang, Lucia Lam, Keiko Sophie Mori, Sebastian Garza, and Boris Katz. 2016. Universal dependencies for learner English. In *Proc. of*

54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 737–746.

Yevgeni Berzak, Roi Reichart, and Boris Katz. 2014. Reconstructing native language typology from foreign language usage. In *Proc. of 18th Conference on Computational Natural Language Learning*, pages 21–29.

Ann Bies, Mark Ferguson, Karen Katz, and Robert MacIntyre. 1995. Bracketing guidelines for Treebank II-style Penn treebank project.

Christopher Bryant, Mariano Felice, and Ted Briscoe. 2017. Automatic annotation and evaluation of error types for grammatical error correction. In *Proceedings of 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 793–805.

Jill Burstein, Karen Kukich, Susanne Wolff, Chi Lu, Martin Chodorow, Lisa Braden-Harder, and Mary D. Harris. 1998. Automated scoring using a hybrid feature identification technique. In *Proc. of 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics*, pages 206–210.

Martin Chodorow and Claudia Leacock. 2002. Techniques for detecting syntactic errors in text. In *IEICE Technical Report (TL2002-39)*, pages 37–41.

Ana Díaz-Negrillo, Detmar Meurers, Salvador Valera, and Holger Wunsch. 2009. Towards interlanguage POS annotation for effective learner corpora in SLA and FLT. *Language Forum*, 36(1–2):139–154.

Markus Dickinson and Marwa Ragheb. 2009. Dependency annotation for learner corpora. In *Proc. of 8th Workshop on Treebanks and Linguistic Theories*, pages 59–70.

Jennifer Foster. 2007a. Treebanks gone bad: generating a treebank of ungrammatical English. In *2007 Workshop on Analytics for Noisy Unstructured Data*, pages 39–46.

Jennifer Foster. 2007b. Treebanks gone bad: Parser evaluation and retraining using a treebank of ungrammatical sentences. *International Journal on Document Analysis and Recognition*, 10(3):129–145.

Sylviane Granger, Estelle Dagneaux, Fanny Meunier, and Magali Paquot. 2009. *International Corpus of Learner English v2*. Presses universitaires de Louvain, Louvain.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Comput.*, 9(8):1735–1780.

Zhiheng Huang, Wei Xu, and Kai Yu. 2015. Bidirectional LSTM-CRF models for sequence tagging.

Emi Izumi, Toyomi Saiga, Thepchai Supnithi, Kiyotaka Uchimoto, and Hitoshi Isahara. 2004. The NICT JLE Corpus: Exploiting the language learners' speech database for research and education. *International Journal of The Computer, the Internet and Management*, 12(2):119–125.

John D. Lafferty, Andrew McCallum, and Fernando C.Ñ. Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proc. of 18th International Conference on Machine Learning*, pages 282–289.

Claudia Leacock, Martin Chodorow, Michael Gamon, and Joel Tetreault. 2010. *Automated Grammatical Error Detection for Language Learners*. Morgan & Claypool, San Rafael.

Xuezhe Ma and Eduard Hovy. 2016. End-to-end sequence labeling via bi-directional LSTM-CNNs-CRF. In *Proc. of 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1064–1074.

Chaitanya Malaviya, Graham Neubig, and Patrick Littell. 2017. Learning language representations for typology prediction. In *Proc. of 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2529–2535.

Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. 2014. The Stanford CoreNLP natural language processing toolkit. In *Proc. of 52nd Association for Computational Linguistics (ACL) System Demonstrations*, pages 55–60.

Andre Martins, Miguel Almeida, and Noah A. Smith. 2013. Turning on the Turbo: Fast third-order non-projective Turbo parsers. In *Proc. of 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 617–622.

Tomoya Mizumoto and Ryo Nagata. 2017. Analyzing the impact of spelling errors on POS-tagging and chunking in learner English. In *Proc. of 4th Workshop on Natural Language Processing Techniques for Educational Applications*, pages 54–58.

Ryo Nagata and Atsuo Kawai. 2011. Exploiting learners' tendencies for detecting English determiner errors. In *Lecture Notes in Computer Science*, volume 6882/2011, pages 144–153.

Ryo Nagata and Keisuke Sakaguchi. 2016. Phrase structure annotation and parsing for learner English. In *Proc. of 54th Annual Meeting of the Association for Computational Linguistics*, pages 1837–1847.

Ryo Nagata and Edward Whittaker. 2013. Reconstructing an Indo-European family tree from nonnative English texts. In *Proc. of 51st Annual Meeting of the Association for Computational Linguistics*, pages 1137–1147.

Ryo Nagata, Edward Whittaker, and Vera Sheinman. 2011. Creating a manually error-tagged and shallow-parsed learner corpus. In *Proc. of 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 1210–1219.

Robert Östling and Jörg Tiedemann. 2017. Continuous multilinguality with language vectors. In *Proc. of 15th Conference of the European Chapter of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 644–649.

Barbara Plank, Anders Søgaard, and Yoav Goldberg. 2016. Multilingual part-of-speech tagging with bidirectional long short-term memory models and auxiliary loss. In *Proc. of 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 412–418.

Marwa Ragheb and Markus Dickinson. 2012. Defining syntax for learner language annotation. In *Proc. of 24th International Conference on Computational Linguistics*, pages 965–974.

Bertus van Rooy and Lande Schäfer. 2002. The effect of learner errors on POS tag errors during automatic POS tagging. *Southern African Linguistics and Applied Language Studies*, 20(4):325–335.

Keisuke Sakaguchi, Tomoya Mizumoto, Mamoru Komachi, and Yuji Matsumoto. 2012. Joint English spelling error correction and POS tagging for language learners writing. In *Proc. of 24th International Conference on Computational Linguistics*, pages 2357–2374.

Beatrice Santorini. 1990. Part-of-speech tagging guidelines for the Penn Treebank project.

Michael Swan and Bernard Smith. 2001. *Learner English (2nd Ed.)*. Cambridge University Press, Cambridge.

Yukio Tono. 2000. A corpus-based analysis of interlanguage development: analysing POS tag sequences of EFL learner corpora. In *Practical Applications in Language Corpora*, pages 123–132.