Joint English Spelling Error Correction and POS Tagging for Language Learners Writing

Keisuke Sakaguchi, Tomoya Mizumoto,
Mamoru Komachi, Yuji Matsumoto
Graduate School of Information Science
Nara Institute of Science and Technology
8916-5, Takayama, Ikoma, Nara 630-0192, Japan
{keisuke-sa, tomoya-m, komachi, matsu}@is.naist.jp

Abstract

We propose an approach to correcting spelling errors and assigning part-of-speech (POS) tags simultaneously for sentences written by learners of English as a second language (ESL). In ESL writing, there are several types of errors such as preposition, determiner, verb, noun, and spelling errors. Spelling errors often interfere with POS tagging and syntactic parsing, which makes other error detection and correction tasks very difficult. In studies of grammatical error detection and correction in ESL writing, spelling correction has been regarded as a preprocessing step in a pipeline. However, several types of spelling errors in ESL are difficult to correct in the preprocessing, for example, homophones (e.g. **hear/here*), confusion (**quiet/quite*), split (**now a day/nowadays*), merge (**swimingpool/swimming pool*), inflection (**please/pleased*) and derivation (**badly/bad*), where the incorrect word is actually in the vocabulary and grammatical information is needed to disambiguate.

In order to correct these spelling errors, and also typical typographical errors (**begginning/beginning*), we propose a joint analysis of POS tagging and spelling error correction with a CRF (Conditional Random Field)-based model. We present an approach that achieves significantly better accuracies for both POS tagging and spelling correction, compared to existing approaches using either individual or pipeline analysis. We also show that the joint model can deal with novel types of misspelling in ESL writing.

Keywords: Part-of-Speech Tagging, Spelling Error Correction.

Proceedings of COLING 2012: Technical Papers, pages 2357–2374,
COLING 2012, Mumbai, December 2012.

1 Introduction

Automated grammatical error detection and correction have been focused on natural language processing (NLP) over the past dozen years or so. Researchers have mainly studied English grammatical error detection and correction of areas such as determiners, prepositions and verbs (Izumi et al., 2003; Han et al., 2006; Felice and Pulman, 2008; Lee and Seneff, 2008; Gamon, 2010; Dahlmeier and Ng, 2011; Rozovskaya and Roth, 2011; Tajiri et al., 2012). In previous work on grammatical error detection and correction, spelling errors are usually corrected in a preprocessing step in a pipeline. These studies generally deal with **typographical** errors (e.g. **begginning/beginning*). In ESL writing, however, there exist many other types of spelling errors, which often occur in combination with, for example, **homophone** (**there/their*), **confusion** (**form/from*), **split** (**Now a day/Nowadays*), **merge** (**swimingpool/swimming pool*), **inflection** (**please/pleased*), and **derivation** (**badly/bad*) errors. Unlike typographical errors, these spelling errors are difficult to detect because the words to be corrected are possible words in English.

Previous studies in spelling correction for ESL writing depend mainly on edit distance between the words before and after correction. Some previous works for correcting misspelled words in native speaker misspellings focus on homophone, confusion, split, and merge errors (Golding and Roth, 1999; Bao et al., 2011), but no research has been done on inflection and derivation errors.

One of the biggest problems in grammatical error detection and correction studies is that ESL writing contains spelling errors, and they are often obstacles to POS tagging and syntactic parsing. For example, POS tagging fails for the following sentence[1]:

Input:
... it is **verey/very *convent/convenient* for the group.
without spelling error correction:
... it/PRP, is/VBZ, verey/PRP, convent/NN ...
with spelling error correction:
... it/PRP, is/VBZ, very/RB, convenient/JJ ...

Conversely, spelling correction requires POS information in some cases. For instance, the sentence below shows that the misspelled word **analysys/analyses* is corrected according to its POS (NNS), while it is difficult to select the best candidate based only on edit distance (analysis/NN or analyses/NNS).

Input:
... research and some **analysys/analyses*.
when assigning POS tags:
... and/CC, some/DT, analysys/NNS ...
candidates and their POS:
['analysis/NN', 'analyses/NNS']

In order to detect and correct errors in ESL writing, spelling correction is essential, because sentences with misspelled words cannot be parsed properly. However, the conventional pipeline for grammatical error detection and correction has a limitation due to the different types of spelling errors and the unavailability of contextual information, which results in failures in the subsequent POS tagging and syntactic parsing (Figure 1(1)).

In this work, we propose a joint model for spelling correction and POS tagging (Figure 1(2)). The model is based on morphological analysis, where each node in a lattice has both POS and

[1] We use Penn treebank-style part-of-speech tags.

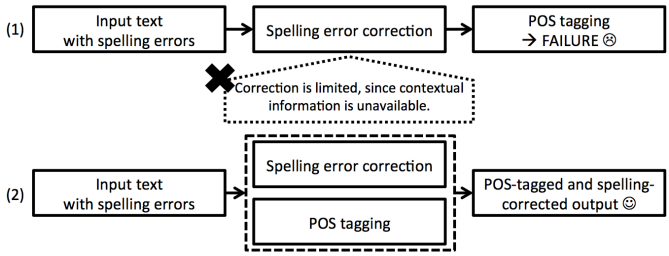

Figure 1: A limitation of pipeline analysis (1), and our proposed joint model (2).

spelling information as features. Because of these features, our method can deal with not only typographical errors but also homophones, confusion, split, merge, inflection and derivation errors. Also, higher accuracy with spelling correction improves POS tagging. We evaluated the joint model with two different ESL learners' error-annotated corpora, with the results showing 2.1% and 3.8% improvement in F-values of POS tagging for the corpora, and 5.0% in F-value of spelling errors. The results significantly outperform baseline and pipeline.

There are three main contributions described in this paper:

1. This is the first joint model for assigning POS tags and correcting misspelled words simultaneously.

2. Our work shows that the joint model improves the accuracy of both POS tagging and spelling correction for ESL writing compared to conventional pipeline methods.

3. This is the first model which is able to correct a wide range of misspelled words, including misspellings due to inflection and derivation errors.

In the following, we first present previous research done on grammatical error correction, spelling correction, and joint analysis (Section 2), and then describe our proposed method in detail (Section 3). The experimental setting and the results are presented in Section 4, and error analysis is given in Section 5. Finally, we conclude in Section 6.

2 Related works

In spelling error correction, the main concern is how to extract confusion pairs that consist of words before and after correction. A number of studies depend on such edit distance between written and corrected words as *Levenshtein Distance* (LD), *Longest Common Subsequence* (LCS) string matching, and pronunciation similarities (Kukich, 1992; Brill and Moore, 2000; Islam and Inkpen, 2009; Bao et al., 2011; Toutanova and Moore, 2002). In order to cover more misspelled words, many spelling errors were collected from web search queries and their results (Chen et al., 2007; Gao et al., 2010), click through logs (Sun et al., 2010), and users' keystroke logs (Baba and Suzuki, 2012). Note that previous studies for spelling correction described above focus on errors made by native speakers rather than second language learners, who show a wider range of misspellings with, for example, split, merge, inflection and derivation errors.

In most grammatical error detection and correction research, spelling error correction is performed before such linguistic analysis as POS tagging and syntactic parsing. Spelling correction as preprocessing generally uses existing spelling checkers such as GNU Aspell[2] and Jazzy[3], which depend on edit distance between words before and after correction. Then, candidate words are often re-ranked or filtered using a language model. In fact, in the Helping Our Own (HOO) 2012 (Dale et al., 2012), which is a shared task on preposition and determiner error correction, highly-ranked teams employ the strategy of spelling correction as preprocessing based on edit distance.

Some recent studies deal with spelling correction at the same time as whole grammatical error correction. For example, (Brockett et al., 2006) presents a method to correct whole sentences containing various errors, applying a statistical machine translation (SMT) technique where input sentences are translated into correct English. Although this approach can deal with any type of spelling errors, it suffers from a poverty of error-annotated resources and cannot correct misspelled words that have never appeared in a corpus. Similarly, (Park and Levy, 2011) propose a noisy channel model to correct errors, although they depend on a bigram language model and do not use syntactic information. A discriminative approach for whole grammatical error correction is also proposed in a recent study (Dahlmeier and Ng, 2012) where spelling errors are corrected simultaneously. In terms of spelling error types, however, typographical errors using GNU Aspell are dealt with, but not other misspelling types such as split and merge errors. Our proposed model uses POS features in order to correct spelling. As result, a wider range of spelling errors such as inflection and derivation errors can be corrected. Inflection and derivation errors are usually regarded as grammatical errors, not spelling errors. However, we include inflection and derivation error correction in our task, given the difficulty of determining whether they are grammatical or spelling errors, as will be explained in Section 4.1.

Joint learning and joint analysis have received much attention in recent studies for linguistic analysis. For example, the CoNLL-2008 Shared Task (Surdeanu et al., 2008) shows promising results in joint syntactic and semantic dependency parsing. There are also models that deal with joint morphological segmentation and syntactic parsing in Hebrew (Goldberg and Tsarfaty, 2008), joint word segmentation and POS tagging in Chinese (Zhang and Clark, 2010), and joint word segmentation, POS tagging and dependency parsing in Chinese (Hatori et al., 2012). These studies demonstrate that joint models outperform conventional pipelined systems. Our work applies for the first time a joint analysis to spelling correction and POS tagging for ESL writing in which input sentences contains multiple errors, whereas previous joint models deal only with canonical texts.

3 Joint analysis of POS tagging and spelling correction

In this section, we describe our proposed joint analysis of spelling error correction and POS tagging for ESL writing. Our method is based on Japanese morphological analysis (Kudo et al., 2004), which disambiguates word boundaries and assigns POS tags using re-defined Conditional Random Fields (CRFs) (Lafferty et al., 1999), while the original CRFs deal with sequential labeling for sentences with word boundaries fixed. We use the re-defined CRFs rather than the original CRFs because disambiguating word boundaries is necessary for split and merge error correction. In terms of decoding, our model has a similar approach to the decoder proposed by (Dahlmeier and Ng, 2012), though the decoder by Dahlmeier and Ng uses beam search. In (Kudo et al., 2004), they define CRFs as the conditional probability of an output path $\mathbf{y} = \left(\langle w_1, t_1 \rangle, ..., \langle w_{\#\mathbf{y}}, t_{\#\mathbf{y}} \rangle \right)$, given

[2] http://aspell.net/
[3] http://jazzy.sourceforge.net/

an input sentence $\mathbf{x}$ with words w and labels t:

$$P(\mathbf{y}|\mathbf{x}) = \frac{1}{Z_{\mathbf{x}}} \exp\left(\sum_{i=1}^{\#\mathbf{y}} \sum_{k} \lambda_k f_k\big(\langle w_{i-1}, t_{i-1} \rangle, \langle w_i, t_i \rangle \big) \right)$$

where $\#\mathbf{y}$ is the number of tokens according to the output sequence, and $Z_{\mathbf{x}}$ is a normalization factor for all candidate paths $\mathcal{Y}(\mathbf{x})$,

$$Z_{\mathbf{x}} = \sum_{\mathbf{y}' \in \mathcal{Y}(\mathbf{x})} \exp\left(\sum_{i=1}^{\#\mathbf{y}'} \sum_{k} \lambda_k f_k\big(\langle w'_{i-1}, t'_{i-1} \rangle, \langle w'_i, t'_i \rangle \big) \right)$$

Here, $f_k\big(\langle w_{i-1}, t_{i-1} \rangle, \langle w_i, t_i \rangle \big)$ is a feature function of the i-th token $\langle w_i, t_i \rangle$ and its previous token $\langle w_{i-1}, t_{i-1} \rangle$. λ_k is the weight for the feature function f_k. When decoding, the most probable path $\hat{\mathbf{y}}$ for an input sentence $\mathbf{x}$ is

$$\hat{\mathbf{y}} = \operatorname*{argmax}_{\mathbf{y} \in \mathcal{Y}(\mathbf{x})} P(\mathbf{y}|\mathbf{x})$$

which can be found with the Viterbi algorithm.

The lexicon consists of basic information: surface form, its base form, and its POS tag. In order to deal with misspelled words, we extend the format of the lexicon appending correctness of spelling and correct form in conjunction with the basic information. With the extended format, we prepare a misspelling dictionary in addition to the existing English dictionary. Here are examples of lexical entries in both dictionaries:

Examples of correct lexicon:
writing,-40,VB,write,VBG,CORR,*
English,152,NN,English,NNP,CORR,*
Examples of lexicon of spelling errors:
absoletuly,-18,RB,absolutely,RB,INCO,absolutely
difficultly,36,JJ,difficult,JJ,INCO,difficult

where each entry consists of a surface form, followed by cost of the word, POS group[4], base form, POS, CORR (correct) / INCO (incorrect) spelling error flag, and correct spelling form. If the flag is CORR, the correct spelling form is written as '*'. In the above examples for the lexicon of spelling errors, **absoletuly/absolutely* is a typographical error and **difficultly/difficult* is a derivation error. The unigram costs in the correct lexicon and POS bigram costs are calculated as a result of learnt weights in the CRFs, and the detail of weights learning of the CRFs is found in Kudo et al.(2004). The cost in the lexicon of spelling errors is obtained based on the corresponding correct form. In other words, the model is able to decode unseen spelling errors, if correct candidates for the misspelled word exist in the correct lexicon. The way to construct a lexicon of spelling errors is described in detail in Section 4. With the additional lexicon, where the cost for each entry is determined, we can decode sentences including spelling errors, with simultaneous spelling correction and POS tagging. Algorithm 1 shows a brief overview of our proposed model for decoding. Figure 2 shows examples of the decoding process, where **beggining/beginning*, **Auguest/August*, and **swimingpool/swimming pool* are misspelled. Without a misspelling dictionary, we fail to decode spelling error words and to assign POS tags (as shown in dotted lines in Figure 2). Because we prepare a misspelling dictionary as explained above, we can decode **begginning* as *beginning*,

[4]POS groups are a coarse version of Penn Treebank POS tags. For example, JJ, JJR and JJS are merged into JJ.

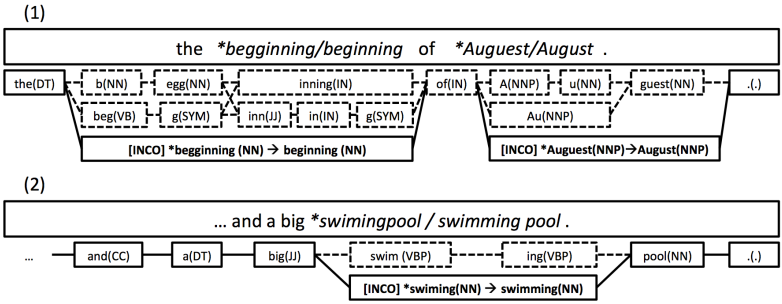

Figure 2: Samples of decoding process with proposed method. "[INCO]" is a misspelling flag.

Algorithm 1 Proposed joint POS tagging and spelling correction algorithm

Input: *Lexicon, Sentence* // *Sentence* ignores blanks between words.
Output: *Best path for the input sentence*
  Lattice $= \phi$
  $i = 0$ // i is letter index of a *Sentence*.
  repeat
    for each node ending with Sentence[i] **do**
      right_nodes = Lexicon.commonPrefixSearch(Sentence[i+1:])
      for for each right_node in right_nodes **do**
        Append right_node with unigram cost into *Lattice*
        Append the edge between node and right_node with POS bigram cost into *Lattice*
      end for
    end for
    i++
  until The end of the input sentence
  Best_Path = **Decode_Viterbi**(*Lattice*)
  return Best_Path

**Auguest* as *August* in Figure 2(1) (shown in solid lines). Furthermore, since the re-defined CRFs deal with word boundary ambiguity, this model is suitable for split and merge spelling error detection and correction as shown in Figure 2(2). In Figure 2(2), where **swimingpool* is a merge error, the misspelled word is split into **swiming/swimming* and *pool*, and corrected from **swiming* to *swimming*.

4 Experiment

4.1 Data

For our experiments, we use two different ESL learners' corpora: the Cambridge Learners Corpus First Certificate in English (CLC FCE) dataset (Yannakoudakis et al., 2011) and the Konan-JIEM learner corpus (KJ corpus) (Nagata et al., 2011). Table 1 shows the statistics of the two corpora. The CLC FCE dataset, which is one of the largest and most commonly used ESL learners' corpora, consists of 1,244 files, and each file consists of two essays with gold-standard error annotation.

	CLC FCE dataset	KJ corpus
# Essays	2,488	233
# Sentences	28,033	3,199
# Tokens	423,850	25,537
1st language	16 languages	Japanese
Error Tagged	Yes	Yes* (Spelling errors are not tagged.)
POS Tagged	No	Yes

Table 1: Statistical overview of the datasets: CLC FCE dataset and KJ corpus.

CLC FCE dataset		KJ corpus	
Error Types	%	Error Types	%
Verb	20.8	Noun	27.6
Punctuation	14.2	Verb	23.9
Spelling	10.7	Article	18.4
Preposition	10.5	Preposition	13.0
Determiner	9.5	Adjective	4.1
Noun	9.3	Adverb	3.4

Table 2: The top 6 error types in CLC FCE dataset and KJ corpus.

The KJ corpus consists of more than 200 essays written by Japanese ESL learners. This is the only dataset where POS tags are assigned for ESL writing. Table 2 shows the proportion of error types for the two datasets. Note that the KJ corpus does not contain error tags for spelling errors and other ungrammatical errors such as punctuation errors.

In terms of spelling errors in the CLC FCE dataset, there are 'S'(spelling) and 'SX'(spelling confusion) error tags. The number of 'S' and 'SX' are 4,922 and 789 respectively. Under the definition of spelling error types in our work, *homophone*, *confusion*, *split*, and *merge* errors are included in 'S' and 'SX' error annotations. There are also 760 'I' (inflection) and 1,913 'D' (derivation) error tags that contain spelling errors such as **usefull/useful* and **suppost/supposed*, in addition to clear examples of inflection/derivation errors (e.g. **badly/bad*). In total, there are 8,349 spelling errors in the CLC FCE dataset, which accounts for 1.9% of the spelling errors in the whole corpus. The distribution of spelling error types is shown in Table 3. A confusion pair is excluded when the original word length is less than 3 letters or when the word is a pronoun, in order to avoid highly frequent words being corrected. We also exclude a confusion pair when the pair derives from semantic confusion (e.g. **dead/killed* and **although/however*).

4.2 Methodology

For training and decoding, we use the MeCab[5] toolkit, a CRF-based POS and morphological analyzer. Table 4 shows the feature template for MeCab (i.e. CRF) training. As mentioned in Section 3, we also use the POS bigrams as the cost of a sequential edge.

The CLC FCE dataset is used for training, development and test sets, where files are randomly divided into 1,000 for training, 100 for development and 100 for test sets. For statistical analysis,

[5]MeCab 0.98 `http://mecab.googlecode.com/svn/trunk/mecab/doc/index.html`

Spelling Error Types	Numbers	%
Typographical	4,859	58.2
Homophone or Confusion	789	9.5
Split	17	0.2
Merge	11	0.1
Inflection	760	9.1
Derivation	1,913	22.9
Total	8,349	

Table 3: A distribution of spelling error types in the CLC FCE dataset.

we take five different samples from the CLC FCE dataset. We use the development set for deciding a hyper-parameter c during MeCab training. We use the KJ corpus only as a test set for POS tagging, because it does not have a gold standard for spelling errors. For evaluating the KJ corpus, we use the same training and development sets of the CLC FCE dataset explained above.

Since the CLC FCE dataset does not contain POS tags, we need to assign POS tags for the corrected sentences in the CLC FCE corpus. We use a MeCab trained on Penn Treebank[6] with the NAIST English dictionary (NAIST edic)[7] (hereafter referred to as MeCab-PTB). The accuracy of MeCab-PTB is 0.974 of precision, 0.980 of recall, and 0.977 of F-value as a result of a preliminary experiment[8]. We assign POS tags by MeCab-PTB for the CLC FCE training set in which all sentences are corrected, and the output is used as the POS-tagged CLC FCE training set (CLC-POS-Train). Then, we train again MeCab-PTB on both Penn Treebank and CLC-POS-Train (referred to as MeCab-CLC).

In order to analyse spelling errors, we extract the pairs of misspelled and corrected words in the CLC FCE training set, so as to develop a lexicon of spelling errors (hereafter LexTrain) as shown in Section 3. The cost for each misspelled entry is extracted from cost-learnt NAIST edic in MeCab-CLC. For example, when the misspelled and corrected word is **bok/book*, the word '*book*' is found in learnt NAIST edic as *book/NN* and *book/VB*. Since the costs for these two candidates are determined, we can construct a lexicon of spelling errors with the flag "INCO". For each of the five training sets, 4,656 entries on average are extracted for the lexicon of spelling errors, including all spelling error types.

For the CLC FCE test set, since we cannot add the gold-standard pairs of misspelled word and its correction directly into the lexicon, we obtain candidates for misspelled words in a test set using GNU Aspell[9]. If the pair of misspelled word and its candidate does not exist in LexTrain, we add the pair into a new lexical dictionary (LexTest), where the cost of learning, POS group, and POS are extracted from learnt NAIST edic in MeCab-CLC. As is the case with LexTrain, all possible entries are added into LexTest for the words that have several POS tags (e.g. NN and VB for *book*). If a candidate word does not exist in learnt NAIST edic, we do not add its pair because no

[6]The Penn Treebank Project Release 2 `http://www.cis.upenn.edu/treebank/`

[7]NAIST-edic-0.2.0 `http://sites.google.com/site/masayua/p/naist-edic`

[8]We use the sections 0-18 of the Penn Treebank for training and sections 22-24 for evaluation. However, it is difficult to make a fair comparison of the result with other PTB POS taggers, since we use *.pos files for training and test sets instead of the *.mrg files that are generally used. We use *.pos files because they have more data.

[9]Because the CLC FCE contains some words, such as proper nouns, that GNU Aspell does not recognize, we add all words in correct sentences of the CLC FCE training set into the GNU Aspell dictionary.

Feature description (Unigram)	Feature description (Bigram)
WORD[i]	WORD[i-1] + WORD[i]
WORD[i] + POS[i]	WORD[i-1] + WORD[i] + POS[i-1]
WORD[i] + POS_group[i]	WORD[i-1] + WORD[i] + POS[i]

Table 4: Feature template of *i*-th token used for training CRF.

information about the cost for the candidate is available. We develop MeCab-CLC+Lex by adding the indices of LexTrain and LexTest into MeCab-CLC.

In our experiment, we analyse test set sentences, where all but spelling errors are corrected beforehand. We compare three conditions for POS tagging: POS-BASELINE, POS-PIPELINE, and POS-JOINT. For POS-BASELINE and POS-PIPELINE, we use MeCab-CLC to analyse test set sentences. In the case of POS-PIPELINE, unknown words in the test set are replaced by the best candidate suggested by GNU Aspell and re-ranked by a 5-gram language model built on the Google 1T Web corpus (Brants and Franz, 2006) with IRSTLM toolkit[10]. In POS-JOINT, we use MeCab-CLC+Lex to analyse the test set.

For spelling correction, we compare two conditions: SP-BASELINE and SP-JOINT. We use GNU Aspell as SP-BASELINE, and the output from POS-JOINT is used for SP-JOINT. With respect to gold standard POS and spelling correction, we analyse the error-free test set with MeCab-PTB.

4.3 Evaluation

We evaluated the performance of POS tagging and spelling correction by computing precision, recall, and F-value. In POS tagging, for each sentence, we count the number of words in the gold standard (*REF-POS*) as N_{REF} and the number of words in system output (*SYS-POS*) as N_{SYS}. In addition, we count the the number of words when the word tokenization and POS tagging match exactly between gold standard and system output (*CORR-POS*) as N_{CORR}. For example, when an input sentence is "*Are you *studing a lot?*" and its reference and output are as follows,

REF-POS: {Are/VBP, you/PRP, studying/VBG, a/DT, lot/NN, ?/? }
SYS-POS: {Are/VBP, you/PRP, stud/JJ, ing/NN, a/DT, lot/NN, ?/?}
CORR-POS: {Are/VBP, you/PRP, a/DT, lot/NN, ?/?}

then N_{REF} is 6, N_{SYS} is 7, and N_{CORR} is 5.

With respect to spelling correction, along with the POS tagging, we count N_{REF}, N_{SYS}, and N_{CORR}, looking at the spelling of tokenized words. N_{REF} is the number of gold-standard spelling correction pairs in (*REF-SP*), N_{SYS} is the number of corrected pairs in the system output (*SYS-SP*), and N_{CORR} is the number of pairs in the system output that correctly identifies the gold standard (*CORR-SP*). For instance, when an input is "*There aren't *convinent *appliaces in their houses yet.*" and the output is "*There aren't **convenient places** in **there** houses yet.*", the result is as follows:

REF-SP: {*appliaces/appliances, *convinent/convenient}
SYS-SP: {*appliaces/places, *convinent/convenient, *their/there}
CORR-SP: {*convinent/convenient}

[10] irstlm 5.70 `http://sourceforge.net/projects/irstlm/files/irstlm/`

		Precision	Recall	F-value
CLC FCE	POS-BASELINE	0.950	0.971	0.960
	POS-PIPELINE	0.961 (+1.1%)	0.976 (+0.5%)	0.968 (+0.8%)
	POS-JOINT	**0.982 (+3.2%)***†	**0.979 (+0.8%)**	**0.981 (+2.1%)***†
KJ corpus	POS-BASELINE	0.794	0.857	0.824
	POS-PIPELINE	0.827 (+3.3%)*	0.857 (± 0.0%)	0.842 (+1.7%)*
	POS-JOINT	**0.853 (+5.9%)***†	**0.871 (+1.4%)***†	**0.862 (+3.8%)***†

Table 5: Experimental result on POS tagging. Statistically significant improvements over the baseline are marked with an asterisk (*), and those over the pipeline are marked with a dagger (†), where $p < 0.05$.

		Precision	Recall	F-value
CLC FCE	SP-BASELINE	**0.519**	0.427	0.468
	SP-JOINT	0.445 (-7.3%)*	**0.622 (+19.5%)***	**0.519 (+5.0%)***

Table 6: Experimental result on spelling error correction. Statistically significant improvements over the baseline are marked with an asterisk (*), where $p < 0.05$.

and therefore, in this case, N_{REF} is 2, N_{SYS} is 3, and N_{CORR} is 1.

Precision, Recall, and F-value are computed by N_{REF}, N_{SYS}, and N_{CORR} as the following equations.

$$Precision = \frac{N_{CORR}}{N_{SYS}}, \quad Recall = \frac{N_{CORR}}{N_{REF}}, \quad F\text{-}value = \frac{2 \times Precision \times Recall}{Precision + Recall}$$

4.4 Result and Analysis

The experimental results of POS tagging is shown in Table 5. From the table, we make the following observations.

First, the joint model and the pipeline perform better than the baseline both in the CLC FCE dataset and the KJ corpus. For the two corpora, the joint model achieves 2.1% and 3.8% improvements and the pipeline achieves 0.8% and 1.7% in F-value, although only POS-JOINT shows statistical significance. Second, the result of the KJ corpus is lower than that of the CLC FCE dataset. This may be because there is a slight difference in segmentation and POS format in the KJ corpus. For example, some words are assigned multiple POS tags such as *everyday/DT-NN* and *logout/VBN-RP*. Furthermore, in the KJ corpus, there are a lot of Japanese words written in Roman letters (e.g. *Onigiri* (*rice ball* in English), *himawari* (*sunflower*)), which make it difficult to segment words and assign POS tags in this corpus. Third, the result shows that our joint analysis performs better in POS tagging than the pipeline in all three metrics for the two ESL learners' corpora. This is because our proposed model assigns POS tags and corrects spelling errors simultaneously, and the joint model can correct not only typographical spelling errors but also homophone, split, merge, inflection, derivation, and confusion errors. Finally, the overall results in the CLC FCE dataset show relatively high values for POS tagging. This may be because the topics in the CLC FCE dataset are limited and there are categorical overlaps between training and test sets.

In terms of spelling error correction, Table 6 presents our experimental results. Overall, the joint model performs better in recall (+19.5%) and F-value (+5.0%), whereas the precision decreases

from 0.519 to 0.445. The result of higher recall and less precision is not surprising, since the joint model can deal with all types of spelling errors whereas only typographical errors are corrected in the baseline.

5 Discussion

In this section, we look at our experimental results in detail and discuss the contribution of our work.

First, looking at the cases when POS tagging and spelling error correction are successfully analysed, we find that the joint model (POS-JOINT) works well for all 7 types of spelling errors we defined. Figure 3 shows successful examples of the 7 error types. For instance, (1) in Figure 3 shows that the word **surprice* is misspelled and split into two words *sur* and *price* in the baseline (POS-BASELINE), whereas the joint model corrects the spelling error and assigns a POS tag successfully. Of course, these typographical errors can be corrected using conventional ways such as edit distance, and in fact these errors are also corrected in the pipeline (POS-PIPELINE), where misspelled words are corrected using edit distance before POS tagging.

The rest of the examples, (2) to (7), in Figure 3 are harder to correct if we depend only on edit distance. However, as pointed out above, the joint model can correct these different kinds of spelling errors. In (2) in Figure 3, the homophone error **hear/here* is corrected in the joint analysis, since the model compares the path costs between the POS sequences of "**... am(VBP)-hear(VB)-to(TO) ...**" and "**... am(VBP)-here(RB)-to(TO) ...**", while the baseline and pipeline cannot figure out the homophone spelling error. The confusion and split errors such as the examples in Figures 3(3) and 3(4) are corrected successfully with the joint model, as is the case of homophone errors. When it comes to merge errors as shown in Figure 3(5), a misspelled word **swimingpool* should be rewritten from **swiming* to *swimming* and also split into *swimming pool*. The joint analysis corrects the error successfully, while the baseline fails to split and the pipeline fails to correct the spelling error. Previous studies, as mentioned in Section 2, deal with the spelling error types shown in examples (1) to (5) in Figure 3, but our work widens the scope of spelling error types to *inflection* and *derivation* errors as shown in the examples in (6) and (7) in Figure 3, since ESL writing contains a number of inflection and derivation errors, as shown in Table 3. In addition, hyphenated words (e.g. **fourty-five/forty-five*) are also corrected by the joint model.

Second, we find several errors, where POS tagging and spelling correction fail. In many error cases, incorrect POS tagging is due to a failure in spelling error correction. In other words, when misspelled words are corrected successfully, the result of POS tagging is also correct. Therefore, we analyse errors in cases of failed spelling correction.

With regard to false positives, when our model could not correct spelling errors in the experiment, we found two main patterns. First, the joint model (SP-JOINT) suggests different words for typographical errors, while the baseline (SP-BASELINE) also tends to fail to correct spelling errors. For example, Figures 4(1) and 4(2) show the failures in typographical error correction. In (1), the misspelled word **beginers* is corrected to *beginner* instead of *beginners*. In the same manner, **concer* in 4(2) is changed to *cancer*. For this pattern, both the baseline and the joint model are able to detect typographical spelling errors, although they fail to suggest correct words. These errors are difficult to correct, because we need information about the broader context or semantics information that sometimes goes beyond the sentence level. Second, our joint model changed correct words into different words. The examples seen in Figures 4(3) and 4(4) show that the proposed model rewrites correct words into different words. In Figure 4(3), the correct word *fell* is rewritten

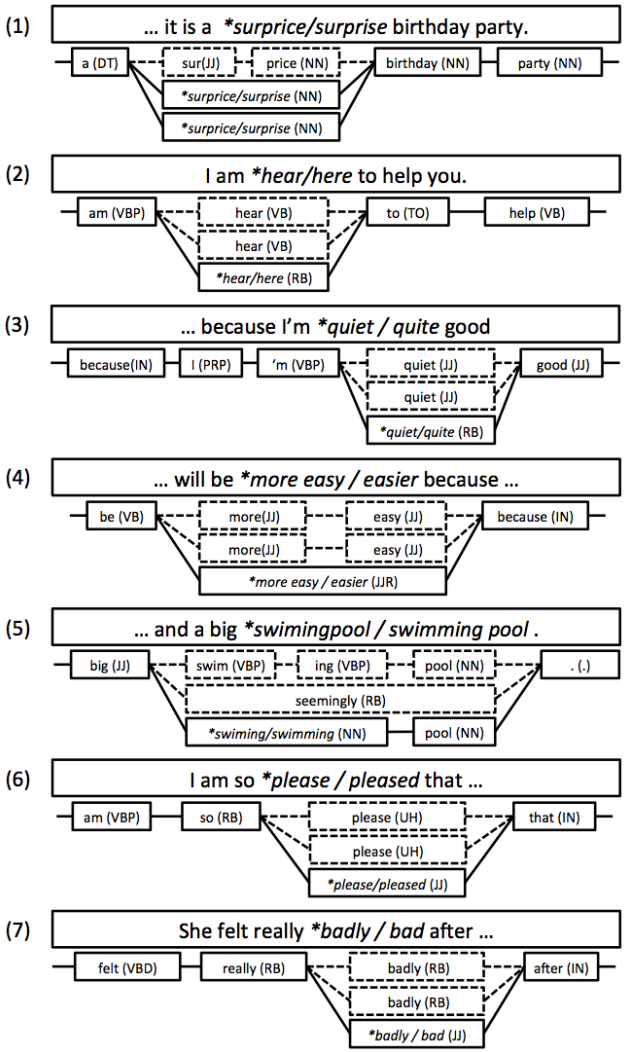

Figure 3: Examples of true positives for POS tagging and spelling correction. Branched nodes represent the output of POS-BASELINE, POS-PIPELINE and POS-JOINT models respectively. Paths and nodes are dotted when they are incorrect. (1) is an example of *typographical error*, (2) is *homophone error*, (3) is *confusion error*, (4) is *split error*, (5) is *merge error*, (6) is *inflection error*, and (7) is *derivation error*.

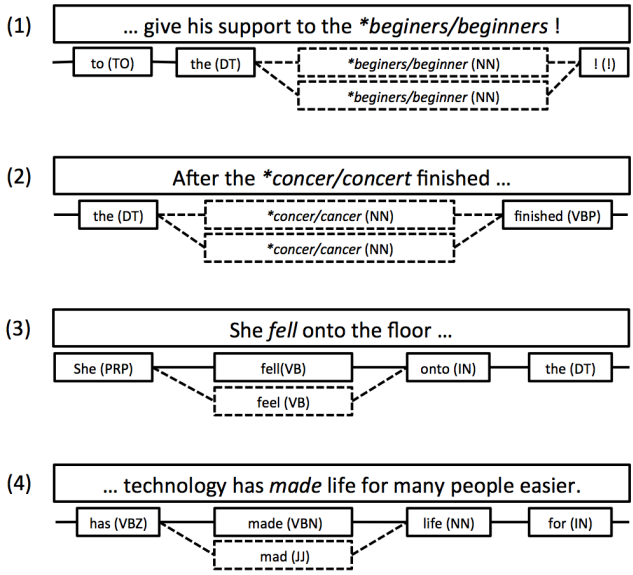

Figure 4: Examples of false positives for spelling correction. Branched nodes represent the output of SP-BASELINE and SP-JOINT models respectively. Edges and nodes are dotted when they are incorrect.

as *feel* and *made* is changed to *mad*. These false positives may be caused by insufficient feature templates and/or data sparseness (overfitting), and we need to deal with them in further research. Of course, both in (3) and (4), this type of wrong corrections does not occur in the baseline, because baseline concerns only typographical errors and does nothing for other types of spelling errors. Since the joint model can detect and correct a wider range of spelling errors, as shown in Figure 3, it causes more false positives, resulting in a lower precision than the baseline. We also find some false positives where the corrected words are also acceptable but regarded as false positives due to the gold standard. Examples of these are British spellings such as *color/colour*, and some adverbs (e.g. *first/firstly*). If we can deal with these cases, the precision will increase.

As shown in Figure 5, we find several examples of false negatives where the system cannot detect spelling errors. In the false negatives, most errors belong to confusion or derivation types, whereas some errors are also found in split and inflection types, indicating that when words before correction are existing words they are hard to detect. For example, Figure 5(1) shows that a misspelled *main* is not detected as an error by the joint model. The error in Figure 5(2) "**After words/Afterwards*" is not corrected, since this error contains a combination of split and typographical errors. With regard to inflection and derivation errors, as Figures 5(3) and 5(4) show, some errors are hard to detect, because the POS sequence before correction is also acceptable to some extent. In order to reduce false negatives, and also false positives, more contextual information will be needed.

Finally, we find that there are some annotation errors in the CLC FCE dataset. For instance, **ab-*

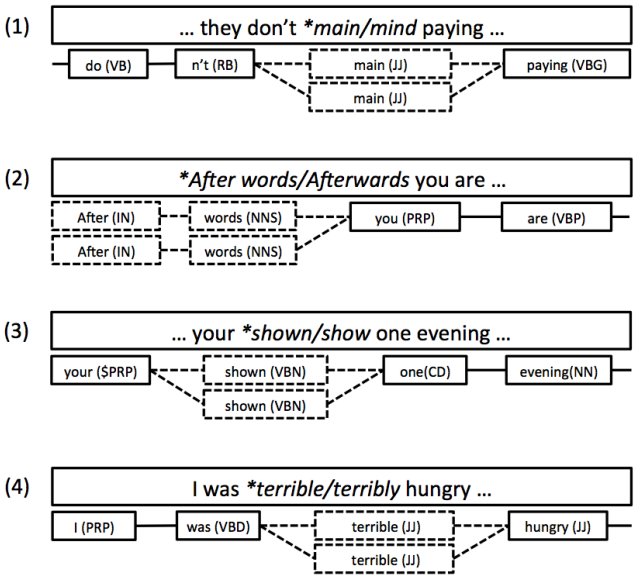

Figure 5: Examples of false negatives for spelling correction. Branched nodes represent the output of SP-BASELINE and SP-JOINT models respectively. Paths and nodes are dotted when they are incorrect.

solutly is corrected to **abolutely* instead of *absolutely*, and **dissapointing* is corrected to **diappointing* instead of *disappointing*. This may force precision downward, though perhaps not to such a great extent.

6 Conclusion

We have presented a joint model of POS tagging and spelling error correction for ESL writing. The model is a CRF-based morphological analysis with word boundary disambiguation. Because the model deals with word boundary ambiguities, it can detect and correctly split and merge errors. In addition, we add misspelled words and their correct/candidate forms into the lexicon, so that the model can deal with not only typographical errors but also a wider range of spelling errors such as homophone, confusion, split, merge, inflection, and derivation errors that often appear in ESL learners' corpora. Our model shows statistically significant improvements in POS tagging and spelling correction, achieving 2.1% and 3.8% of F-value improvements for POS tagging and 5.0% of F-value improvement for spelling error correction compared to the baseline. We have also showed that the joint model improves F-values more than the pipeline model, which is statistically significant.

Acknowledgments

We would like to thank anonymous reviewers for their valuable and very helpful comments.

References

Baba, Y. and Suzuki, H. (2012). How Are Spelling Errors Generated and Corrected? A Study of Corrected and Uncorrected Spelling Errors Using Keystroke Logs. *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics*, pages 373–377.

Bao, Z., Kimelfeld, B., and Li, Y. (2011). A Graph Approach to Spelling Correction in Domain-Centric Search. *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics*, pages 905–914.

Brants, T. and Franz, A. (2006). Web 1T 5-gram Corpus version 1.1. *Technical report, Google Research*.

Brill, E. and Moore, R. C. (2000). An Improved Error Model for Noisy Channel Spelling Correction. *Proceedings of the 38th Annual Meeting on Association for Computational Linguistics*, pages 286–293.

Brockett, C., Dolan, W. B., and Gamon, M. (2006). Correcting ESL Errors Using Phrasal SMT Techniques. *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 249–256.

Chen, Q., Li, M., and Zhou, M. (2007). Improving Query Spelling Correction Using Web Search Results. *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 181–189.

Dahlmeier, D. and Ng, H. T. (2011). Grammatical Error Correction with Alternating Structure Optimization. *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics*, pages 915–923.

Dahlmeier, D. and Ng, H. T. (2012). A Beam-Search Decoder for Grammatical Error Correction. *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 568–578.

Dale, R., Anisimoff, I., and Narroway, G. (2012). HOO 2012 : A Report on the Preposition and Determiner Error Correction Shared Task. *The 7th Workshop on the Innovative Use of NLP for Building Educational Applications*, pages 54–62.

Felice, R. D. and Pulman, S. G. (2008). A Classifier-based Approach to Preposition and Determiner Error Correction in L2 English. *Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008)*, pages 169–176.

Gamon, M. (2010). Using Mostly Native Data to Correct Errors in Learners' Writing: a Meta-Classifier Approach. *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 163–171.

Gao, J., Li, X., Micol, D., Quirk, C., and Sun, X. (2010). A Large Scale Ranker-based System for Search Query Spelling Correction. *Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010)*, pages 358–366.

Goldberg, Y. and Tsarfaty, R. (2008). A Single Generative Model for Joint Morphological Segmentation and Syntactic Parsing. *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics*, pages 371–379.

Golding, A. R. and Roth, D. (1999). A Winnow-Based Approach to Context-Sensitive Spelling Correction. *Machine Learning*, 34:107–130.

Han, N.-R., Chodorow, M., and Leacock, C. (2006). Detecting Errors in English Article Usage by Non-Native Speakers. *Natural Language Engineering*, 12(02):115–129.

Hatori, J., Matsuzaki, T., Miyao, Y., and Tsujii, J. (2012). Incremental Joint Approach to Word Segmentation, POS Tagging, and Dependency Parsing in Chinese. *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics*, pages 1045–1053.

Islam, A. and Inkpen, D. (2009). Real-Word Spelling Correction using Google Web 1T 3-grams. *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 1241–1249.

Izumi, E., Uchimoto, K., Saiga, T., Supnithi, T., and Isahara, H. (2003). Automatic Error Detection in the Japanese Learners' English Spoken Data. *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics*, pages 145–148.

Kudo, T., Yamamoto, K., and Matsumoto, Y. (2004). Applying Conditional Random Fields to Japanese Morphological Analysis. *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pages 230–237.

Kukich, K. (1992). Techniques for Automatically Correcting Words in Text. *ACM Computing Surveys*, 24(4):377–439.

Lafferty, J., McCallum, A., and Pereira, F. (1999). Conditional Random Fields : Probabilistic Models for Segmenting and Labeling Sequence Data. *Proceedings of the 18th International Conference on Machine Learning 2001 (ICML 2001)*, pages 282–289.

Lee, J. and Seneff, S. (2008). Correcting Misuse of Verb Forms. *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics*, pages 174–182.

Nagata, R., Whittaker, E., and Sheinman, V. (2011). Creating a Manually Error-tagged and Shallow-parsed Learner Corpus. *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics*, pages 1210–1219.

Park, Y. A. and Levy, R. (2011). Automated Whole Sentence Grammar Correction Using a Noisy Channel Model. *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics*, pages 934–944.

Rozovskaya, A. and Roth, D. (2011). Algorithm Selection and Model Adaptation for ESL Correction Tasks. *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics*, pages 924–933.

Sun, X., Gao, J., Micol, D., and Quirk, C. (2010). Learning Phrase-Based Spelling Error Models from Clickthrough Data. *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 266–274.

Surdeanu, M., Johansson, R., Meyers, A., Màrquez, L., and Nivre, J. (2008). The CoNLL-2008 Shared Task on Joint Parsing of Syntactic and Semantic Dependencies. *Proceedings of the 12th Conference on Computational Natural Language Learning*, pages 159–177.

Tajiri, T., Komachi, M., and Matsumoto, Y. (2012). Tense and Aspect Error Correction for ESL Learners Using Global Context. *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 198–202.

Toutanova, K. and Moore, R. C. (2002). Pronunciation Modeling for Improved Spelling Correction. *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 144–151.

Yannakoudakis, H., Briscoe, T., and Medlock, B. (2011). A New Dataset and Method for Automatically Grading ESOL Texts. *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics*, pages 180–189.

Zhang, Y. and Clark, S. (2010). A Fast Decoder for Joint Word Segmentation and POS -Tagging Using a Single Discriminative Model. *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 843–852.