Grammar Error Correction Using Pseudo-Error Sentences and Domain Adaptation

Kenji Imamura, Kuniko Saito, Kugatsu Sadamitsu, and **Hitoshi Nishikawa**
NTT Cyber Space Laboratories, NTT Corporation
1-1 Hikari-no-oka, Yokosuka, 239-0847, Japan
{imamura.kenji, saito.kuniko, sadamitsu.kugatsu, nishikawa.hitoshi}@lab.ntt.co.jp

Abstract

This paper presents grammar error correction for Japanese particles that uses discriminative sequence conversion, which corrects erroneous particles by substitution, insertion, and deletion. The error correction task is hindered by the difficulty of collecting large error corpora. We tackle this problem by using pseudo-error sentences generated automatically. Furthermore, we apply domain adaptation, the pseudo-error sentences are from the source domain, and the real-error sentences are from the target domain. Experiments show that stable improvement is achieved by using domain adaptation.

1 Introduction

Case marks of a sentence are represented by postpositional particles in Japanese. Incorrect usage of the particles causes serious communication errors because the cases become unclear. For example, in the following sentence, it is unclear what must be deleted.

mail **o** *todoi tara sakujo onegai-shi-masu*
mail **ACC.** arrive when delete please
"When ϕ has arrived an e-mail, please delete it."

If the accusative particle *o* is replaced by a nominative one *ga*, it becomes clear that the writer wants to delete the e-mail ("When the e-mail has arrived, please delete it."). Such particle errors frequently occur in sentences written by non-native Japanese speakers.

This paper presents a method that can automatically correct Japanese particle errors. This task corresponds to preposition/article error correction in English. For English error correction, many studies employ classifiers, which select the appropriate prepositions/articles, by restricting the error types to articles and frequent prepositions (Gamon, 2010; Han et al., 2010; Rozovskaya and Roth, 2011).

On the contrary, Mizumoto et al. (2011) proposed translator-based error correction. This approach can handle all error types by converting the learner's sentences into the correct ones. Although the target of this paper is particle error, we employ a similar approach based on sequence conversion (Imamura et al., 2011) since this offers excellent scalability.

The conversion approach requires pairs of the learner's and the correct sentences. However, collecting a sufficient number of pairs is expensive. To avoid this problem, we use additional corpus consisting of pseudo-error sentences automatically generated from correct sentences that mimic the real-errors (Rozovskaya and Roth, 2010b). Furthermore, we apply a domain adaptation technique that regards the pseudo-errors and the real-errors as the source and the target domain, respectively, so that the pseudo-errors better match the real-errors.

2 Error Correction by Discriminative Sequence Conversion

We start by describing discriminative sequence conversion. Our error correction method converts the learner's word sequences into the correct sequences. Our method is similar to phrase-based statistical machine translation (PBSMT), but there are three differences; 1) it adopts the conditional random fields, 2) it allows insertion and deletion, and 3) binary and real features are combined. Unlike the classification

Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics, pages 388–392,
Jeju, Republic of Korea, 8-14 July 2012.

Incorrect Particle	Correct Particle	Note
ϕ	no/POSS.	INS
ϕ	o/ACC.	INS
ga/NOM.	o/ACC.	SUB
o/ACC.	ni/DAT.	SUB
o/ACC.	ga/NOM.	SUB
wa/TOP.	o/ACC.	SUB
no/POSS.	ϕ	DEL
:	:	

Table 1: Example of Phrase Table (partial)

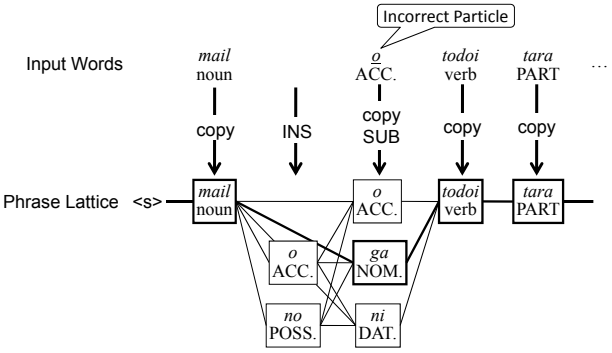

Figure 1: Example of Phrase Lattice

approach, the conversion approach can correct multiple errors of all types in a sentence.

2.1 Basic Procedure

We apply the morpheme conversion approach that converts the results of a speech recognizer into word sequences for language analyzer processing (Imamura et al., 2011). It corrects particle errors in the input sentences as follows.

- First, all modification candidates are obtained by referring to a phrase table. This table, called the confusion set (Rozovskaya and Roth, 2010a) in the error correction task, stores pairs of incorrect and correct particles (Table 1). The candidates are packed into a lattice structure, called the phrase lattice (Figure 1). To deal with unchanged words, it also copies the input words and inserts them into the phrase lattice.
- Next, the best phrase sequence in the phrase lattice is identified based on the conditional random fields (CRFs (Lafferty et al., 2001)). The Viterbi algorithm is applied to the decoding because error correction does not change the word order.
- While training, word alignment is carried out by dynamic programming matching. From the alignment results, the phrase table is constructed by acquiring particle errors, and the CRF models are trained using the alignment results as supervised data.

2.2 Insertion / Deletion

Since an insertion can be regarded as replacing an empty word with an actual word, and deletion is the replacement of an actual word with an empty one, we treat these operations as substitution without distinction while learning/applying the CRF models.

However, insertion is a high cost operation because it may occur at any location and can cause lattice size to explode. To avoid this problem, we permit insertion only immediately after nouns.

2.3 Features

In this paper, we use mapping features and link features. The former measure the correspondence between input and output words (similar to the translation models of PBSMT). The latter measure the fluency of the output word sequence (similar to language models).

The mapping features are all binary. The focusing phrase and its two surrounding words of the input are regarded as the window. The mapping features are defined as the pairs of the output phrase and 1-, 2-, and 3-grams in the window.

The link features are important for the error correction task because the system has to judge output correctness. Fortunately, CRF, which is a kind of discriminative model, can handle features that depend on each other; we mix two types of features as follows and optimize their weights in the CRF framework.

- **N-gram features:** N-grams of the output words, from 1 to 3, are used as binary features. These are obtained from a training corpus (paired sentences). Since the feature weights are optimized considering the entire feature space, fine-tuning can be achieved. The accuracy becomes almost perfect on the training corpus.
- **Language model probability:** This is a logarithmic value (real value) of the n-gram probability of the output word sequence. One feature weight is assigned. The n-gram language model can be

constructed from a large sentence set because it does not need the learner's sentences.

Incorporating binary and real features yields a rough approximation of generative models in semi-supervised CRFs (Suzuki and Isozaki, 2008). It can appropriately correct new sentences while maintaining high accuracy on the training corpus.

3 Pseudo-error Sentences and Domain Adaptation

The error corrector described in Section 2 requires paired sentences. However, it is expensive to collect them. We resolve this problem by using pseudo-error sentences and domain adaptation.

3.1 Pseudo-Error Generation

Correct sentences, which are halves of the paired sentences, can be easily acquired from corpora such as newspaper articles. Pseudo-errors are generated from them by the substitution, insertion, and deletion functions according to the desired error patterns.

We utilize the method of Rozovskaya and Roth (2010b). Namely, when particles appear in the correct sentence, they are replaced by incorrect ones in a probabilistic manner by applying the phrase table (which stores the error patterns) in the opposite direction. The error generation probabilities are relative frequencies on the training corpus. The models are learnt using both the training corpus and the pseudo-error sentences.

3.2 Adaptation by Feature Augmentation

Although the error generation probabilities are computed from the real-error corpus, the error distribution that results may be inappropriate. To better fit the pseudo-errors to the real-errors, we apply a domain adaptation technique. Namely, we regard the pseudo-error corpus as the source domain and the real-error corpus as the target domain, and models are learnt that fit the target domain.

In this paper, we use Daume (2007)'s feature augmentation method for the domain adaptation, which eliminates the need to change the learning algorithm. This method regards the models for the source domain as the prior distribution and learns the models for the target domain.

	Feature Space: Common	Source	Target
Source Data	$\boldsymbol{D_s}$	$\boldsymbol{D_s}$	$\mathbf{0}$
Target Data	$\boldsymbol{D_t}$	$\mathbf{0}$	$\boldsymbol{D_t}$

Figure 2: Feature Augmentation

We briefly review feature augmentation. The feature space is segmented into three parts: common, source, and target. The features extracted from the source domain data are deployed to the common and the source spaces, and those from the target domain data are deployed to the common and the target spaces. Namely, the feature space is tripled (Figure 2).

The parameter estimation is carried out in the usual way on the above feature space. Consequently, the weights of the common features are emphasized if the features are consistent between the source and the target. With regard to domain dependent features, the weights in the source or the target space are emphasized.

Error correction uses only the features in the common and target spaces. The error distribution approaches that of the real-errors because the weights of features are optimized to the target domain. In addition, it becomes robust against new sentences because the common features acquired from the source domain can be used even when they do not appear in the target domain.

4 Experiments

4.1 Experimental Settings

Real-error Corpus: We collected learner's sentences written by Chinese native speakers. The sentences were created from English Linux manuals and figures, and Japanese native speakers revised them. From these sentences, only particle errors were retained; the other errors were corrected. As a result, we obtained 2,770 paired sentences. The number of incorrect particles was 1,087 (8.0%) of 13,534. Note that most particles did not need to be revised. The number of pair types of incorrect particles and their correct ones was 132.

Language Model: It was constructed from Japanese Wikipedia articles about computers and

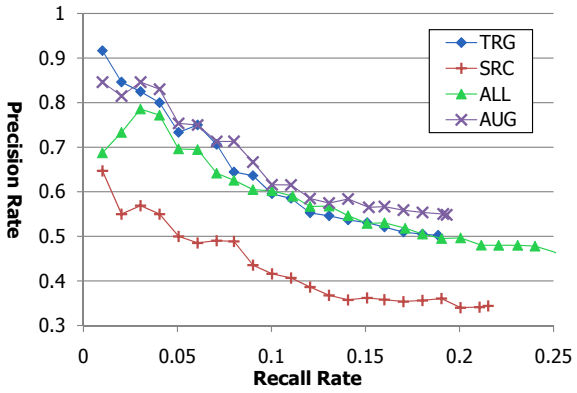

Figure 3: Recall/Precision Curve (Error Generation Magnification is 1.0)

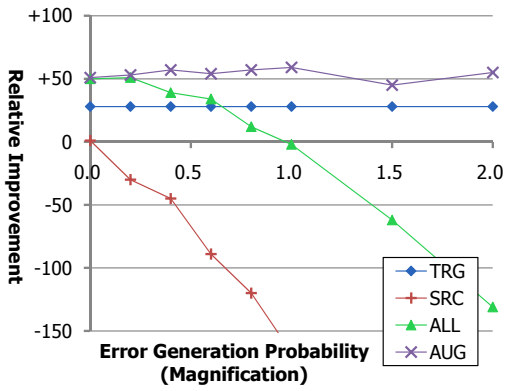

Figure 4: Relative Improvement among Error Generation Probabilities

Japanese Linux manuals, 527,151 sentences in total. SRILM (Stolcke et al., 2011) was used to train a trigram model.

Pseudo-error Corpus: The pseudo-errors were generated using 10,000 sentences randomly selected from the corpus for the language model. The magnification of the error generation probabilities was changed from 0.0 (i.e., no errors) to 2.0 (the relative frequency in the real-error corpus was taken as 1.0).

Evaluation Metrics: Five-fold cross-validation on the real-error corpus was used. We used two metrics: 1) Precision and recall rates of the error correction by the systems, and 2) Relative improvement, the number of differences between improved and degraded particles in the output sentences (no changes were ignored). This is a practical metric because it denotes the number of particles that human rewriters do not need to revise after the system correction.

4.2 Results

Figure 3 plots the precision/recall curves for the following four combinations of training corpora and method.

- **TRG:** The models were trained using only the real-error corpus (baseline).
- **SRC:** Trained using only the pseudo-error corpus.
- **ALL:** Trained using the real-error and pseudo-error corpora by simply adding them.
- **AUG:**
  The proposed method. The feature augmentation was realized by regarding the pseudo-errors as the source domain and the real-errors as the target domain.

The SRC case, which uses only the pseudo-error sentences, did not match the precision of TRG. The ALL case matched the precision of TRG at high recall rates. AUG, the proposed method, achieved higher precision than TRG at high recall rates. At the recall rate of 18%, the precision rate of AUG was 55.4%; in contrast, that of TRG was 50.5%. Feature augmentation effectively leverages the pseudo-errors for error correction.

Figure 4 shows the relative improvement of each method according to the error generation probabilities. In this experiment, ALL achieved higher improvement than TRG at error generation probabilities ranging from 0.0 to 0.6. Although the improvements were high, we have to control the error generation probability because the improvements in the SRC case fell as the magnification was raised. On the other hand, AUG achieved stable improvement regardless of the error generation probability. We can conclude that domain adaptation to the pseudo-error sentences is the preferred approach.

5 Conclusions

This paper presented an error correction method of Japanese particles that uses pseudo-error generation. We applied domain adaptation in which the pseudo-errors are regarded as the source domain and the real-errors as the target domain. In our experiments, domain adaptation achieved stable improvement in system performance regardless of the error generation probability.

References

Hal Daume, III. 2007. Frustratingly easy domain adaptation. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics (ACL 2007)*, pages 256–263, Prague, Czech Republic.

Michael Gamon. 2010. Using mostly native data to correct errors in learners' writing. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL-HLT 2010)*, pages 163–171, Los Angeles, California.

Na-Rae Han, Joel Tetreault, Soo-Hwa Lee, and Jin-Young Ha. 2010. Using an error-annotated learner corpus to develop an ESL/EFL error correction system. In *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*, Valletta, Malta.

Kenji Imamura, Tomoko Izumi, Kugatsu Sadamitsu, Kuniko Saito, Satoshi Kobashikawa, and Hirokazu Masataki. 2011. Morpheme conversion for connecting speech recognizer and language analyzers in unsegmented languages. In *Proceedings of Interspeech 2011*, pages 1405–1408, Florence, Italy.

John Lafferty, Andrew McCallum, and Fernando Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the 18th International Conference on Machine Learning (ICML-2001)*, pages 282–289, Williamstown, Massachusetts.

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 5th International Joint Conference on Natural Language Processing (IJCNLP 2011)*, pages 147–155, Chiang Mai, Thailand.

Alla Rozovskaya and Dan Roth. 2010a. Generating confusion sets for context-sensitive error correction. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing (EMNLP 2010)*, pages 961–970, Cambridge, Massachusetts.

Alla Rozovskaya and Dan Roth. 2010b. Training paradigms for correcting errors in grammar and usage. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL-HLT 2010)*, pages 154–162, Los Angeles, California.

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: Human Language Techologies (ACL-HLT 2011)*, pages 924–933, Portland, Oregon.

Andreas Stolcke, Jing Zheng, Wen Wang, and Victor Abrash. 2011. SRILM at sixteen: Update and outlook. In *Proceedings of IEEE Automatic Speech Recognition and Understanding Workshop (ASRU 2011)*, Waikoloa, Hawaii.

Jun Suzuki and Hideki Isozaki. 2008. Semi-supervised sequential labeling and segmentation using giga-word scale unlabeled data. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (ACL-08: HLT)*, pages 665–673, Columbus, Ohio.