Automated Grammar Correction Using Hierarchical Phrase-Based Statistical Machine Translation

Bibek Behera, Pushpak Bhattacharyya
Dept. of Computer Science and Engineering
IIT Bombay, Mumbai, India
{bibek,pb}@cse.iitb.ac.in

Abstract

We introduce a novel technique that uses hierarchical phrase-based statistical machine translation (SMT) for grammar correction. SMT systems provide a uniform platform for any sequence transformation task. Thus grammar correction can be considered a translation problem from incorrect text to correct text. Over the years, grammar correction data in the electronic form (i.e., parallel corpora of incorrect and correct sentences) has increased manifolds in quality and quantity, making SMT systems feasible for grammar correction. Firstly, sophisticated translation models like hierarchical phrase-based SMT can handle errors as complicated as reordering or insertion, which were difficult to deal with previously throuh the mediation of rule based systems. Secondly, this SMT based correction technique is similar in spirit to human correction, because the system extracts grammar rules from the corpus and later uses these rules to translate incorrect sentences to correct sentences. We describe how to use Joshua, a hierarchical phrase-based SMT system for grammar correction. An accuracy of 0.77 (BLEU score) establishes the efficacy of our approach.

1 Introduction

We consider grammar correction as a translation problem - translation from an incorrect sentence to a correct sentence. The correcting system is trained using a parallel corpus of incorrect and their corresponding correct sentences. The system learns SCFG (synchronous context free grammar) rules (Chiang, 2005) during translation. SCFG rules look like this:-

- $X \rightarrow X_1\, of\, X_2, X_1\, for\, X_2$

The above rule implies that phrases X_1 and X_2 in source language are translated to phrases in target language, while *of* is replaced with *for*. The position of both phrases w.r.t. *of* remains same in the target language, which means there is no reordering of phrases.

After generating such grammar rules, it converts the erroneous sentence to a tree using the rules of grammar, *i.e.*, the left hand side of the SCFG rules. It then applies correction rules, *i.e.*, the right hand side of the SCFG rules, to convert the tree as explained later in section 3. The yield of the tree generates the corrected sentence.

Here are various types of errors that one encounters in grammar correction:

Article choice errors:- *a Himalayas is the longest mountain range in the world.* The correct translation is '*the Himalayas is the longest mountain range in the world*'.

Preposition errors:- *Helicopter crashed at central London.* The correct translation is '*Helicopter crashed in central London*'.

Word form errors:- *The rain mays fall in July* should be changed to '*The rain may fall in July*'.

Word insertion errors:- *The court deemed necessary that she respond to the summons* should be changed to '*The court deemed it necessary that she respond to the summons*'.

Reordering errors:- *never we miss deadlines* should be corrected to '*we never miss deadlines*'.

Article choice errors and preposition errors have been tackled by rule based techniques. But rules are customized, so to say, for each error, which is a time consuming and fragile process. SMT, on the other hand, treats all errors uniformly, considering error correction as a translation problem. Secondly, problems such as reordering or word insertion are well known in machine translation.

The roadmap of the paper is as follows. In section 2, we discuss previous work. In section 3, we elaborate on how hierarchical machine translation system can do automatic grammar correction. Section 4 states the grammar rules that are extracted by the system automatically. In section 5, we present our experiments followed by the results in section 6. We conclude in section 7 with pointers to future work.

2 Background

Initially the work that has been done in grammar correction is based on identifying grammar errors. Chodorow and Leacock (2000) used an ngram model for error detection by comparing correct ngrams with ngrams to be tested. Later, classification techniques like Maximum entropy models have been proposed (Izumi et al., 2003; Tetreault and Chodorow, 2008; Tetreault and Chodorow, 2008). These classifiers not only identify errors, but also correct them. These methods do not make use of erroneous words thus making error correction similar to the task of filling empty blanks. While in editing sentences, humans often require the information in the erroneous words for grammar correction.

In other works, machine translation has been previously used for grammar correction. Brockett (2006) used phrasal based MT for noun correction of ESL students. Désilets and Hermet (2009) translate from native language L1 to L2 and back to L1 to correct grammar in their native languages. Mizumoto (2012) also used phrase-based SMT for error correction. He used large-scale learner corpus to train his system. These translation techniques suffered from lack of good quality parallel corpora and also good translation systems.

If high quality parallel corpus can be obtained, the task of grammar correction becomes easy using a powerful translation model like hierarchical phrase based machine translation.

3 Automatic grammar correction using hierarchical phrase-based SMT

In this section we discuss the working and the implementation of the grammar correction system.

3.1 Working

Grammar correction can be seen as a process of translation of incorrect sentences to correct ones. Basically the translation system needs a parallel corpus of incorrect and correct sentences. The system starts with an alignment to obtain word to word translation probabilities. The second stage is grammar extraction using the hiero style of grammar (Chiang, 2005). Non-terminals are generalized form of phrases, *i.e.*, all possible phrases allowed in the framework of Chiang (2005) are represented by the symbol *X*. There is another symbol *S* to start the parse tree. These rules are in the form of SCFG rules. If the incorrect sentence is, '*few has arrived*' and the correct sentence is, '*few have arrived*', the grammar rules extracted are :-

- $X \rightarrow$ few has X_1, few have X_1
- $X \rightarrow$ arrived, arrived

The first rule means that *few has* followed by a phrase may be translated to *few have* followed by translation of that phrase. Second rule suggests that any phrase that yields *arrived* can be translated to *arrived*.

After the grammar extraction is done, the left sides of the grammar rules are stripped and used to generate the parse tree of the sentence *few has arrived*.
Here are the left side rules:-

- X $\rightarrow$ few has X_1
- X $\rightarrow$ arrived

Also, there is a "glue rule" to combine two trees or just derive a non terminal from the start symbol *S*.

- S $\rightarrow$ S[1] X | X

The glue rule is used to start the parsing process. It generates a sub-tree for the string *few has* and a non-terminal for *arrived*. Then the right side rules are used to convert *few has* to *few have* as shown in Figure 1, while *arrived* is translated as *arrived*.

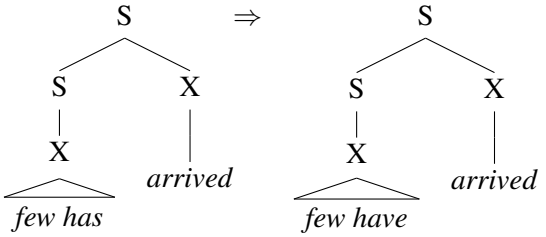

Figure 1: Parse tree for transformation from incorrect to correct sentences.

[1]Here S means start of the tree

The yield of the tree generates *few have arrived*, which is the required correction. This is the essence of decoding in hierarchical machine translation.

3.2 Implementation

The translation system being used is the Joshua Machine translation system (Li et al., 2010). The SMT based correction pipeline is a six step process in conformity with the Joshua decoder (Ganitkevitch et al., 2012). First we create the dataset in a input folder with six files such as:-

1. train.incorrect- Incorrect sentences in our training corpus
2. train.correct- Correct sentences in our training corpus
3. tune.correct- Incorrect sentences in our development set
4. tune.incorrect- Correct sentences in our development set
5. test.correct- Incorrect sentences in our testing set
6. test.incorrect- Correct sentences in our testing set

The pipeline starts with preprocessing the corpus, *i.e.*, tokenisation and lowercasing followed by word alignment. The result of word alignment is stored in training.align file. Then a file, "grammar.gz" is created by joshua that stores SCFG rules using information from the training.align file and the training corpus. This process is called grammar generation and is followed by the building of the language model.

For developing the language model, the Joshua MT system uses KenLM (Heafield, 2011) toolkit or BerkeleyLM. This is the end of the training process. The steps that follow in the pipeline are tuning and testing. Tuning iterates over the development set to obtain the best parameters for the translation model. At the end of tuning, the system obtains the optimized parameters that can be deployed into the translation model for testing. The testing phase translates sentences from test set to evaluate the overall BLEU score (Papineni et al., 2002).

4 Analysis of grammar rules extracted

In this section we look at how various grammar corrections have been handled. The various types of errors handled are article choice errors, preposition errors, word-form choice errors and word insertion errors as mentioned in Park and Levy (2011). Apart from these errors, we also discuss errors due to reordering and errors due to unseen verbs which have not been implemented in previous models.

4.1 Article choice errors

The article *a* has been replaced by *the* before proper nouns like *a amazon* and *a himalayas*. The grammar rules are:-

- $X \rightarrow$ a himalayas X_1, the himalayas X_1
- $X \rightarrow$ a amazon X_1, the amazon X_1

The rules suggest that *a himalayas* succeeded by a phrase X_1 can be replaced by *the himalayas* followed by the same phrase.

4.2 Preposition errors

Preposition *at* has been replaced by *in* before a place like *at central London*. The grammar rule is:-

- $X \rightarrow X_1$ at central London, X_1 in central London

4.3 Unknown Verb correction

Lets say the training data has these sentences

- *He like milk* $\rightarrow$ *He likes milk*
- *They hate the pollution* $\rightarrow$ *They hate pollution*

This system will not be able to correct *He hate milk*, because hate needs to be corrected to hates and its grammar has no rule for *hate* $\rightarrow$ *hates*. But it has a rule for *like* $\rightarrow$ *likes*. From these two rules, the grammar extractor wont be able to derive *hate* $\rightarrow$ *hates*. This can be solved by splitting *likes* to *like s*

- *He like milk* $\rightarrow$ *He like s milk*

Now extractor will have a rule for this training sentence.

- $X \rightarrow$ He X_1 milk, He X_1 s milk

- $X \rightarrow$ hate, hate

Using these two rules it generates *He hate s milk* from *He hate milk*. Later we combine all the split verbs to get *He hate s milk*.

4.4 Word insertion errors

As the name suggests these errors are due to missing words, *e.g.*,

- *The court deemed necessary that she respond to the summons.* $\rightarrow$ *The court deemed it necessary that she respond to the summons.*

For this example the grammar rule extracted is :-

- $X \rightarrow X_1$ deemed X_2, X_1 deemed it X_2

4.5 Reordering errors

Reordering errors arise due to incorrect ordering of the subject object verb, *e.g.*,

- Given Hindi sentence:- *सेन्ट्रल लन्डन मे गिरा हेलिकोप्टर*
- Transliteration of Hindi sentence is:- *sentrala landana me giraa helicoptera*
- The correct translation of this sentence is:- *helicopter crash in central London*
- Output translation from Hindi-English translation system of this sentence:- *central down in London helicopter.*

If the output translation and correct translation is added to the training corpus of grammar correction system such as,

- *central down in London helicopter* $\rightarrow$ *helicopter down in central London*.

we can obtain the correct translation.

5 Experiments

Now we present the data set and evaluation techniques for our experiment.

5.1 Data set

We ran the grammar correction system on the NUCLE (NUS Corpus of Learner English) corpus (Dahlmeier et al., 2013). The dataset has been developed at NUS in collaboration with the Centre for English Language Communication (CELC). This is a parallel corpus of 50000 incorrect and correct sentences, all aligned. We took a subset of 4000 line training corpus, 3000 for training and 1000 for testing.

5.2 Cleaning training corpus

This is a preprocessing step before training the grammar correction system. This was primarily due to the presence of noisy data like:-

1. HYPERLINK- *http://en.wikipedia.org/wiki/*
2. Bracketed information:- (DoD) {Common Access Card}
3. Citations:- (Ben, 2008)
4. Presence of sentence pairs without any changes.

6 Results

We present the results of SMT based grammar correction in table 1. The results show improvement in BLEU score with increase in the size of training corpus. The baseline is the system which passes incorrect sentences as such *i.e.*, performs *no correction*. We wanted to check what the bleu score would be when no correction is incorporated.

Size of training corpus (sentences)	**Size of tuning corpus (sentences)**	**Size of testing corpus (sentences)**	**BLEU score**
Baseline			0.7551
1000	1000	1000	0.7668
2000	1000	1000	0.7679
3000	1000	1000	0.7744

Table 1: Variation of accuracy with variation of training size

7 Conclusion

We have shown how a hierarchical phrase-based MT system like Joshua could be used as a grammar correction system. We observed that increasing training data definitely increases accuracy because patterns in grammar correction keep repeating even if test data is completely different from training set. In future work, we would like to concentrate on "unknown word handling".

References

Chris Brockett, William B. Dolan and Michael Gamon. 2006. *Correcting ESL errors using phrasal SMT techniques*. ACL '06, Sydney, Australia.

Daniel Dahlmeier, Hwee Tou Ng and Siew Mei Wu. 2013. *Building a Large Annotated Corpus of Learner English: The NUS Corpus of Learner English.*. Proceedings of the 8th Workshop on Innovative Use of NLP for Building Educational Applications. BEA 2013, Atlanta, Georgia, USA.

David Chiang. 2005. *A Hierarchical Phrase-Based Model for Statistical Machine Translation*. Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics. ACL '05, Ann Arbor, Michigan.

Emi Izumi, Kiyotaka Uchimoto, Toyomi Saiga, Thepchai Supnithi and Hitoshi Isahara. 2003. *Automatic error detection in the Japanese learners' English spoken data*. Proceedings of the 41st Annual Meeting on Association for Computational Linguistics-Volume 2:145–148. ACL '03, Sapporo, Japan.

Joel R. Tetreault and Martin Chodorow . 2008. *The ups and downs of preposition error detection in ESL writing*. COLING '08, Manchester, United Kingdom.

Juri Ganitkevitch, Yuan Cao, Jonathan Weese, Matt Post and Chris Callison-Burch. 2012. *Joshua 4.0: packing, PRO, and paraphrases*. Proceedings of the Seventh Workshop on Statistical Machine Translation. WMT '12, Montreal, Canada.

Kenneth Heafield. 2011. *KenLM: faster and smaller language model queries*. Proceedings of the Sixth Workshop on Statistical Machine Translation. WMT '11, Edinburgh, Scotland.

Kishore Papineni, Salim Roukos and Todd Ward and Wei-Jing Zhu. 2002. *BLEU: a method for automatic evaluation of machine translation*. Proceedings of the 40th Annual Meeting on Association for Computational Linguistics. ACL '02, Philadelphia, Pennsylvania.

Martin Chodorow and Claudia Leacock. 2000. *An Unsupervised Method for Detecting Grammatical Errors*. NAACL 2000, Seattle, Washington.

Matthieu Hermet and Alain Désilets. 2009 *Using first and second language models to correct preposition errors in second language authoring*. EdAppsNLP '09, Boulder, Colorado.

Omar F. Zaidan. 2009. *Z-MERT: A Fully Configurable Open Source Tool for Minimum Error Rate Training of Machine Translation Systems*. The Prague Bulletin of Mathematical Linguistics, 91:79–88.

Percy Liang, Ben Taskar and Dan Klein. . 2006. *Alignment by agreement*. Proceedings of the main conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics. HLT-NAACL '06, New York.

Peter F. Brown, Vincent J. Della Pietra, Stephen A. Della Pietra and Robert L. Mercer. . 1993. *The mathematics of statistical machine translation: parameter estimation*. Computational Linguistics, 1993.

Rachele De Felice and Stephen G Pulman. 2008. *A classifier-based approach to preposition and determiner error correction in L2 English*. COLING '08, Manchester, United Kingdom.

Tomoya Mizumoto, Yuta Hayashibe, Mamoru Komachi, Masaaki Nagata and Yuji Matsumoto. 2012. *The Effect of Learner Corpus Size in Grammatical Error Correction of ESL Writings*. COLING '12, Mumbai, India.

Y. Albert Park and Roger Levy. 2011. *Automated whole sentence grammar correction using a noisy channel model*. HLT '11, Portland, Oregon.

Zhifei Li, Chris Callison-Burch, Chris Dyer, Juri Ganitkevitch, Ann Irvine, Sanjeev Khudanpur, Lane Schwartz, Wren N. G. Thornton, Ziyuan Wang and Jonathan Weese, and Omar F. Zaidan. 2010. *Joshua 2.0: a toolkit for parsing-based machine translation with syntax, semirings, discriminative training and other goodies*. Proceedings of the Joint Fifth Workshop on Statistical Machine Translation and Metrics MATR. WMT '10, Uppsala, Sweden.