Syntax-Driven Machine Translation as a Model of ESL Revision

Huichao Xue and **Rebecca Hwa**
Department of Computer Science
University of Pittsburgh
{hux10,hwa}@cs.pitt.edu

Abstract

In this work, we model the writing revision process of English as a Second Language (ESL) students with syntax-driven machine translation methods. We compare two approaches: tree-to-string transformations (Yamada and Knight, 2001) and tree-to-tree transformations (Smith and Eisner, 2006). Results suggest that while the tree-to-tree model provides a greater coverage, the tree-to-string approach offers a more plausible model of ESL learners' revision writing process.

1 Introduction

When learning a second language, students make mistakes along the way. While some mistakes are idiosyncratic and individual, many are systematic and common to people who share the same primary language. There has been extensive research on grammar error detection. Most previous efforts focus on identifying specific types of problems commonly encountered by English as a Second Language (ESL) learners. Some examples include the proper usage of determiners (Yi et al., 2008; Gamon et al., 2008), prepositions (Chodorow et al., 2007; Gamon et al., 2008; Hermet et al., 2008), and mass versus count nouns (Nagata et al., 2006). However, previous work suggests that *grammar error correction* is considerably more challenging than detection (Han et al., 2010). Furthermore, an ESL learner's writing may contain multiple interacting errors that are difficult to detect and correct in isolation.

A promising research direction is to tackle automatic grammar error correction as a machine translation (MT) problem. The disfluent sentences produced by an ESL learner can be seen as the input source language, and the corrected revision is the result of the translation. Brockett et al. (2006) showed that phrase-based statistical MT can help to correct mistakes made on mass nouns. To our knowledge, phrase-based MT techniques have not been applied for rewriting entire sentences. One major challenge is the lack of appropriate training data such as a sizable parallel corpus. Another concern is that phrase-based MT may not be similar enough to the problem of correcting ESL learner mistakes. While MT rewrites an entire source sentence into the target language, not every word written by an ESL learner needs to be modified.

Another alternative that may afford a more general model of ESL error corrections is to consider syntax-driven MT approaches. We argue that syntax-based approaches can overcome the expected challenges in applying MT to this domain. First, it can be less data-intensive because the mapping is formed at a structural level rather than the surface word level. While it does require a robust parser, a syntax-driven MT model may not need to train on a very large parallel corpus. Second, syntactic transformations provide an intuitive description of how second language learners revise their writings: they are transforming structures in their primary language to those in the new language.

In this paper, we conduct a first inquiry into the applicability of syntax-driven MT methods to automatic grammar error correc-

tion. In particular, we investigate whether a syntax-driven model can capture ESL students' process of writing revisions. We compare two approaches: a tree-to-string mapping proposed by Yamada & Knight (2001) and a tree-to-tree mapping using the Quasi-Synchronous Grammar (QG) formalism (Smith and Eisner, 2006). We train both models on a parallel corpus consisting of multiple drafts of essays by ESL students. The approaches are evaluated on how well they model the revision pairs in an unseen test corpus. Experimental results suggest that 1) the QG model has more flexibility and is able to describe more types of transformations; but 2) the YK model is better at capturing the incremental improvements in the ESL learners' revision writing process.

2 Problem Description

This paper explores the research question: can ESL learners' process of revising their writings be described by a computational model? A successful model of the revision process has several potential applications. In addition to automatic grammar error detection and correction, it may also be useful as an automatic metric in an intelligent tutoring system to evaluate how well the students are learning to make their own revisions.

Revising an ESL student's writing bears some resemblance to translating. The student's first draft is likely to contain disfluent expressions that arose from translation divergences between English and the student's primary language. In the revised draft, the divergences should be resolved so that the text becomes fluent English. We investigate to what extent are formalisms used for machine translation applicable to model writing revision. We hypothesize that ESL students typically modify sentences to make them sound more fluent rather than to drastically change the meanings of what they are trying to convey. Thus, our work focuses on syntax-driven MT models.

One challenge of applying MT methods to model grammar error correction is the lack of appropriate training data. The equivalence to the bilingual parallel corpus used for developing MT systems would be a corpus in which each student sentence is paired with a fluent version re-written by an instructor. Unlike bilingual text, however, there is not much data of this type in practice because there are typically too many students for the teachers to provide detailed manual inspection and correction at a large scale. More commonly, students are asked to revise their previously written essays as they learn more about the English language. Here is an example of a student sentence from a first-draft essay:

> The problem here is that they come to the US like illegal.

In a later draft, it has been revised into:

> The problem here is that they come to the US illegally.

Although the students are not able to create "gold standard revisions" due to their still imperfect understanding of English, a corpus that pairs the students' earlier and later drafts still offers us an opportunity to model how ESL speakers make mistakes.

More formally, the corpus $\mathcal{C}$ consists of a set of sentence pairs (O, R), where O represents the student's original draft and R represents the revised draft. Note that while R is assumed to be an improvement upon O, its quality may fall short of the gold standard revision, G. To train the syntax-driven MT models, we optimize the joint probability of observing the sentence pair, $\Pr(O, R)$, through some form of mapping between their parse trees, τ_O and τ_R.

An added wrinkle to our problem is that it might not always be possible to assign a sensible syntactic structure to an ungrammatical sentence. It is well-known that an English parser trained on the Penn Treebank is bad at handling disfluent sentences (Charniak et al., 2003; Foster et al., 2008). In our domain,

since O (and perhaps also R) might be disfluent, an important question that a translation model must address is: how should the mapping between the trees τ_O and τ_R be handled?

3 Syntax-Driven Models for Essay Revisions

There is extensive literature on syntax-driven approaches to MT (cf. a recent survey by Lopez (2008)); we focus on two particular formalisms that reflects different perspectives on the role of syntax. Our goal is to assess which formalism is a better fit with the domain of essay revision modeling, in which the data largely consist of imperfect sentences that may not support a plausible syntactic interpretation.

3.1 Tree-to-String Model

The Yamada & Knight (henceforth, YK) tree-to-string model is an instance of noisy channel translation systems, which assumes that the observed source sentence is the result of transformation performed on the parse tree of the intended target sentence due to a noisy communication channel. Given a parallel corpus, and a parser for the the target side, the parameters of this model can be estimated using EM(Expectation Maximization). The trained model's job is to recover the target sentence (and tree) through decoding.

While the noisy channel generation story may sound somewhat counter-intuitive for translation, it gives a plausible account of ESL learner's writing process. The student really wants to convey a fluent English sentence with a well-formed structure, but due to an imperfect understanding of the language, writes down an ungrammatical sentence, O, as a first draft. The student serves as the noisy channel. The YK model describes this as a stochastic process that performs three operations on τ_G, the parse of the intended sentence, G:

1. Each node in τ_G may have its children **reordered** with some probability.
2. Each node in τ_G may have a child node **inserted** to its left or right with some probability.
3. Each leaf node (i.e., surface word) in τ_G is **replaced** by some (possibly empty) string according to its lexical translation distribution.

The resulting sentence, O, is the concatenation of the leaf nodes of the transformed τ_G.

Common mistakes made by ESL learners, such as misuses of determiners and prepositions, word choice errors, and incorrect constituency orderings, can be modeled by a combination of the **insert**, **replace**, and **reorder** operators. The YK model allows us to perform transformations on a higher syntactic level. Another potential benefit is that the model does not attempt to assign syntactic interpretations over the source sentences (i.e., the less fluent original draft).

3.2 Tree-to-Tree Model

The Quasi-Synchronous Grammar formalism (Smith and Eisner, 2006) is a generative model that aims to produce the most likely target tree for a given source tree. It differs from the more strict synchronous grammar formalisms (Wu, 1995; Melamed et al., 2004) because it does not try to perform simultaneous parsing on parallel grammars; instead, the model learns an augmented target-language grammar whose rules make "soft alignments" with a given source tree.

QG has been applied to some NLP tasks other than MT, including answer selection for question-answering (Wang et al., 2007), paraphrase identification (Das and Smith, 2009), and parser adaptation and projection (Smith and Eisner, 2009). In this work we use an instantiation of QG that largely follows the model described by Smith and Eisner (2006). The model is trained on a parallel corpus in which both the first-draft and revised sentences have been parsed. Using

EM to estimate its parameters, it learns an augmented target PCFG grammar[1] whose production rules form associations with the given source trees.

Consider the scenario in Figure 1. Given a source tree τ_O, the trained model generates a target tree by expanding the production rules in the augmented target PCFG. To apply a target-side production rule such as

$$A \rightarrow BC,$$

the model considers which source tree nodes might be associated with each target-side non-terminals:

$$(\alpha, A) \rightarrow (\beta, B)(\gamma, C)$$

where α, β, γ are nodes in τ_O. Thus, assuming that the target symbol A has already been aligned to source node α from an earlier derivation step, the likelihood of expanding (α, A) with the above production rule depends on three factors:

1. the likelihood of the **monolingual target rule**, $\Pr(A \rightarrow BC)$
2. the likelihood of **alignments** between B and β as well as C and γ.
3. the likelihood that the source nodes form some expected **configuration** (i.e., between α and β as well as between α and γ). In this work, we distinguish between two configuration types: *parent-child* and *other*. This restriction doesn't reduce the explanatory power of the resulting QG model, though it may not be as fine-tuned as some models in (Smith and Eisner, 2006).

Under QG, the ESL students' first drafts are seen as text in a different language that has its own syntactic constructions. QG explains the grammar rules that govern the revised text in terms of how different components map to structures in the original draft.

[1]For expository purposes, we illustrate the model using a PCFG production rule. In the experiment, a statistical English *dependency* parser (Klein and Manning, 2004) was used.

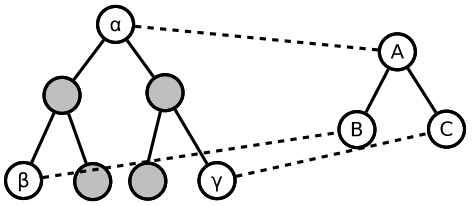

Figure 1: An example of QG's soft alignments between a given source tree and a possible target rule expansion.

It makes explicit the representation of divergences between the students' original mental model and the expected structure.

3.3 Method of Model Comparison

Cross entropy can be used as a metric that measures the distance between the learned probabilistic model and the real data. It can be interpreted as measuring the amount of information that is needed in addition to the model to accurately recover the observed data. In language modeling, cross entropy is widely used in showing a given model's prediction power.

To determine how well the two syntax-driven MT models capture the ESL student revision generation process, we measure the cross entropy of each trained model on an unseen test corpus. This quantity measures how surprised a model is about relating an initial sentence, O, to its corresponding revision, R. Specifically, the cross entropy for some model M on a test corpus $\mathcal{C}$ of original and revised sentence pairs (O, R) is:

$$-\frac{1}{|\mathcal{C}|} \sum_{(O,R)\in\mathcal{C}} \log \Pr_M(O, R)$$

Because neither model computes the joint probability of the sentence pair, we need to make additional computations so that the models can be compared directly.

The YK model computes the likelihood of the first-draft sentence O given an assumed gold parse τ_R of the revised sentence: $\Pr_{YK}(O \mid \tau_R)$. To determine the joint prob-

ability, we would need to compute:

$$\begin{aligned}\Pr_{YK}(O,R) &= \sum_{\tau_R\in\Lambda_R}\Pr_{YK}(O,\tau_R)\\ &= \sum_{\tau_R\in\Lambda_R}\Pr_{YK}(O\mid\tau_R)\Pr(\tau_R)\end{aligned}$$

where Λ_R represents the set of possible parse trees for sentence R. Practically, performing tree-to-string mapping over the entire set of trees in Λ_R is computationally intractable. Moreover, the motivation behind the YK model is to trust the given τ_R. Thus, we made a Viterbi approximation:

$$\begin{aligned}\Pr_{YK}(O,R) &= \sum_{\tau_R\in\Lambda_R}\Pr_{YK}(O\mid\tau_R)\Pr(\tau_R)\\ &\approx \Pr_{YK}(O\mid\hat{\tau_R})\Pr(\hat{\tau_R})\end{aligned}$$

where $\Pr(\hat{\tau_R})$ is the probability of the single best parse tree according to a standard English parser.

Similarly, to compute the joint sentence pair probability under the QG model would require summing over both sets of trees because the model computes $\Pr_{QG}(\tau_R \mid \tau_O)$. Here, we make the Viterbi approximation on both trees.

$$\begin{aligned}\Pr_{QG}(O,R) &= \sum_{\tau_R\in\Lambda_R}\sum_{\tau_O\in\Lambda_O}\Pr_{QG}(\tau_O,\tau_R)\\ &= \sum_{\tau_R\in\Lambda_R}\sum_{\tau_O\in\Lambda_O}\Pr_{QG}(\tau_R\mid\tau_O)\Pr(\tau_O)\\ &\approx \Pr_{QG}(\hat{\tau_R}\mid\hat{\tau_O})\Pr(\hat{\tau_O})\end{aligned}$$

where $\hat{\tau_O}$ and $\hat{\tau_R}$ are the best parses for sentences O and R according to the underlying English dependency parser, respectively.

4 Experiments

4.1 Data

Our experiments are conducted using a collection of ESL students' writing samples[2].

[2]The dataset is made available by the Pittsburgh Science of Learning Center English as a Second Language Course Committee, supported by NSF Award SBE-0354420.

	mean	stdev
percentage of $O = R$	54.11%	N/A
O's length	12.95	4.87
R's length	12.74	4.20
edit distance	1.88	3.58

Table 1: This table summarizes some statistics of the dataset.

These are short essays of approximately 30 sentences on topics such as "a letter to your parents." The students are asked to revise their essays at least once. From the dataset, we extracted 358 article pairs.

Typically, the changes between the drafts are incremental. Approximately half of the sentences are not changed at all. These sentences are considered useful because this phenomenon strongly implies that the original version is good enough to the best of the author's knowledge. In a few rare cases, students may write an entirely different essay. We applied TF-IDF to automatically align the sentences between essay drafts. Any sentence pair with a cosine similarity score of less than 0.3 is filtered. This resulted in a parallel corpus of 7580 sentence pairs.

Because both models are computational intensive, we further restricted our experiments to sentence pairs for which the revised sentence has no more than 20 words. This reduces our corpus to 4666 sentence pairs. Some statistics of the sentence pairs are shown in Table 1.

4.2 Experimental Setup

We randomly split the resulting dataset into a training corpus of 4566 sentence pairs and a test corpus of 100 pairs.

The training of both models involve an EM algorithm. We initialize the model parameters with some reasonable values. Then, in each iteration of training, the model parameters are re-estimated by collecting the expected counts across possible alignments between each sentence pair in the training corpus. In out experiments, both models had two iterations of training. Below, we

highlight our initialization procedure for each model.

In the YK model, the initial **reordering** probability distribution is set to prefer no change 50% of the time. The remaining probability mass is distributed evenly over all of the other permutations. For the **insertion** operation, for each node, the YK model first chooses whether to insert a new string to its left, to its right, or not at all, conditioned on the node's label and its parent's label. These distributions are initialized uniformly ($\frac{1}{3}$). If a new string should be inserted, the model then makes that choice with some probability. The insertion probability of each string in the dictionary is assigned evenly with $\frac{1}{N}$, where N is the number of words in the dictionary. Finally, the **replace** probability distribution is initialized uniformly with the same value ($\frac{1}{N+1}$) across all words in the dictionary, including the empty string.

For the QG model, the initial parameters are determined as follows: For the **monolingual target parsing model parameters**, we first parse the target side of the corpus (i.e., the revised sentences) with the Stanford parser; we then use the maximum likelihood estimates based on these parse trees to initialize the parameters of the target parser, Dependency Model with Valence (DMV). We uniformly initialized the **configuration parameters**; the *parent-child* configuration and *other* configuration each has 0.5 probability. For the **alignment parameters**, we ran the GIZA++ implementation of the IBM word alignment model (Och and Ney, 2003) on the sentence pairs, and used the resulting translation table as our initial estimation. There may be better initialization setups, but the difference between those setups will become small after a few rounds of EM.

Once trained, the two models compute the joint probability of every sentence pair in the test corpus as described in Section 3.3.

4.3 Experiment I

To evaluate how well the models describe the ESL revision domain, we want to see which model is less "surprised" by the test data. We expected that the better model should be able to transform more sentence pair in the test corpus; we also expect that the better model should have a lower cross entropy with respect to the test corpus.

Applying both YK and QG to the test corpus, we find that neither model is able to transform all the test sentence pairs. Of the two, QG had the better coverage; it successfully modeled 59 pairs out of 100 (we denote this subset as D_{QG}). In contrast, YK modeled 36 pairs (this subset is denoted as D_{YK}).

To determine whether there were some characteristics of the data that made one model better at performing transformations for certain sentence pairs, we compare corpus statistics for different test subsets. Based on the results summarized in Table 2, we make a few observations.

First, the sentence pairs that neither model could transform seem, as a whole, more difficult. Their average lengths are longer, and the average per word Levenshtein edit distance is bigger. The differences between *Neither* and the other subsets are statistically significant with 90% confidence. For the length difference, we applied standard two-sample t-test. For the edit distance difference, we applied hypothesis testing with the null-hypothesis that "longer sentence pairs are as likely to be covered by our model as shorter ones."

Second, both models sometimes have trouble with sentence pairs that require no change. This may be due to out-of-vocabulary words in the test corpus. A more aggressive smoothing strategy could improve the coverage for both models.

Third, comparing the subset of sentence pairs that only QG could transform ($D_{QG} - D_{YK}$) against the subset of sentences that both models could transform ($D_{QG} \cap D_{YK}$), the former has slightly higher average edit

	Neither	$D_{QG} \cap D_{YK}$	$D_{QG} - D_{YK}$	$D_{YK} - D_{QG}$
number of instances	38	33	26	3
average edit distance	2.42	1.88	2.08	1
% of identical pairs	53%	48%	58%	67%
average O length	14.63	12.36	12.58	6.67
average R length	13.87	12.06	12.62	6.67
QG cross entropy	N/A	127.95	138.9	N/A
YK cross entropy	N/A	78.76	N/A	43.84

Table 2: A comparison of the two models based on their coverage of the test corpus. Some relevant statistics on the sentence subsets are also summarized in the table.

	YK	QG
overall entropy	78.76	127.95
on identical pairs	52.59	85.40
on non-identical pairs	103.99	168.00

Table 3: A further comparison of the two models on $D_{QG} \cap D_{YK}$, the sentence pairs in the test corpus that both could transform.

distance and length, but the difference is not statistically significant. Although QG could transform more sentence pairs, the cross entropy of $D_{QG} - D_{YK}$ is higher than QG's estimate for the $D_{QG} \cap D_{YK}$ subset. QG's soft alignment property allows it to model more complex transformations with greater flexibility.

Finally, while the YK model has a more limited coverage, it models those transformations with a greater certainty. For the common subset of sentence pairs that both models could transform, YK has a much lower cross entropy than QG. Table 3 further breaks down the common subset. It is not surprising that both models have low entropy for identical sentence pairs. For modeling sentence pairs that contain revisions, YK is more efficient than QG.

4.4 Experiment II

The results of the previous experiment raises the possibility that QG might have a greater coverage because it is too flexible. However, an appropriate model should not only assign large probability mass to positive examples, but it should also have a low chance of choosing negative examples. In this next experiment, we construct a "negative" test corpus to see how it affects the models.

To construct a negative scenario, we still use the same test corpus as before, but we *reverse* the sentence pairs. That is, we use the revised sentences as "originals" and the original sentences as "revisions." We would expect a good model to have a raised cross entropy values along with a drop in coverage on the new dataset because the "revisions" should be more disfluent than the "original" sentences.

Table 4 summarizes the results. We observe that the number of instances that can be transformed has dropped for both models: from 59 to 49 pairs for QG, and from 36 to 20 pairs for YK; also, the proportion of identical instances in each set has raised. This means that both models are more surprised by the reverse test corpus, suggesting that both models have, to some extent, succeeded in modeling the ESL revision domain. However, QG still allows for many more transformations. Moreover, 16 out of the 49 instances are non-identical pairs. In contrast, YK modeled only 1 non-identical sentence pair. The results from these two experiments suggest that YK is more suited for modeling the ESL revision domain than QG. One possible explanation is that QG allows more flexibility and would require more training. Another possible explanation is that because YK assumes well-formed syntax structure for only the target side, the philosophy behind

	Neither	$D_{QG} \cap D_{YK}$	$D_{QG} - D_{YK}$	$D_{YK} - D_{QG}$
number of instances	50	19	30	1
average edit distance	2.88	0.05	2.17	1
percentage of identical pairs	0.40	0.95	0.5	0
average O length	14.18	9.00	12.53	17
average R length	14.98	9.05	12.47	16
QG cross entropy	N/A	81.85	139.36	N/A
YK cross entropy	N/A	51.2	N/A	103.75

Table 4: This table compares the two models on a "trick" test corpus in which the earlier and later drafts are reversed. If a model is trained to prefer more fluent English sentences are the revision, it should be perplexed on this corpus.

its design is a better fit with the ESL revision problem.

5 Related Work

There are many research directions in the field of ESL error correction. A great deal of the work focuses on the lexical or shallow syntactic level. Typically, local features such as word identity and POS tagging information are combined to deal with some specific kind of error. Among them, (Burstein et al., 2004) developed a tool called Critique that detects collocation errors and word choice errors. Nagata et al. (2006) uses a rule-based approach in distinguishing mass and count nouns. Knight and Chander (1994) and Han et al. (2006) both addressed the misuse of articles. Chodorow et al. (2007), Gamon et al. (2008), Hermet et al. (2008) proposed several techniques in detecting and correcting proposition errors. In detecting errors and giving suggestions, Liu et al. (2000), Gamon et al. (2008) and Hermet et al. (2008) make use of information retrieval techniques. Chodorow et al. (2007) instead treat it as a classification problem and employed a maximum entropy classifier. Similar to our approach, Brockett et al. (2006) view error correction as a Machine Translation problem. But their translation system is built on phrase level, with the purpose of correcting local errors such as mass noun errors.

The problem of error correction at a syntactic level is less explored. Lee and Seneff (2008) examined the task of correcting verb form misuse by applying tree template matching rules. The parse tree transformation rules are learned from synthesized training data.

6 Conclusion

This paper investigates the suitability of syntax-driven MT approaches for modeling the revision writing process of ESL learners. We have considered both the Yamada & Knight tree-to-string model, which only considers syntactic information from the typically more fluent revised text, as well as Quasi-Synchronous Grammar, a tree-to-tree model that attempts to learn syntactic transformation patterns between the students' original and revised texts. Our results suggests that while QG offers a greater degree of freedom, thus allowing for a better coverage of the transformations, YK has a lower entropy on the test corpus. Moreover, when presented with an alternative "trick" corpus in which the "revision" is in fact the earlier draft, YK was more perplexed than QG. These results suggest that the YK model may be a promising approach for automatic grammar error correction.

Acknowledgments

This work has been supported by NSF Grant IIS-0745914. We thank Joel Tetreault and the anonymous reviewers for their helpful comments and suggestions.

References

Brockett, Chris, William B. Dolan, and Michael Gamon. 2006. Correcting esl errors using phrasal smt techniques. In *Proceedings of COLING-ACL 2006*, Sydney, Australia, July.

Burstein, Jill, Martin Chodorow, and Claudia Leacock. 2004. Automated essay evaluation: The criterion online writing service. *AI Magazine*, 25(3).

Charniak, Eugene, Kevin Knight, and Kenji Yamada. 2003. Syntax-based language models for machine translation. In *Proc. MT Summit IX*, New Orleans, Louisiana, USA.

Chodorow, Martin, Joel Tetreault, and Na-Rae Han. 2007. Detection of grammatical errors involving prepositions. In *Proceedings of the 4th ACL-SIGSEM Workshop on Prepositions*, Prague, Czech Republic.

Das, Dipanjan and Noah A. Smith. 2009. Paraphrase identification as probabilistic quasi-synchronous recognition. In *Proceedings of ACL-IJCNLP 2009*, Suntec, Singapore, August.

Foster, Jennifer, Joachim Wagner, and Josef van Genabith. 2008. Adapting a WSJ-trained parser to grammatically noisy text. In *Proceedings of the 46th ACL on Human Language Technologies: Short Papers*, Columbus, Ohio.

Gamon, Michael, Jianfeng Gao, Chris Brockett, Alexandre Klementiev, William B. Dolan, Dmitriy Belenko, and Lucy Vanderwende. 2008. Using contextual speller techniques and language modeling for ESL error correction. In *Proceedings of IJCNLP*, Hyderabad, India.

Han, Na-Rae, Martin Chodorow, and Claudia Leacock. 2006. Detecting errors in English article usage by non-native speakers. *Natural Language Engineering*, 12(02).

Han, Na-Rae, Joel Tetreault, Soo-Hwa Lee, and Jin-Young Han. 2010. Using an error-annotated learner corpus to develop and ESL/EFL error correction system. In *Proceedings of LREC 2010*, Valletta, Malta.

Hermet, Matthieu, Alain Désilets, and Stan Szpakowicz. 2008. Using the web as a linguistic resource to automatically correct Lexico-Syntactic errors. In *Proceedings of the LREC*, volume 8.

Klein, Dan and Christopher Manning. 2004. Corpus-based induction of syntactic structure: Models of dependency and constituency. In *Proceedings of ACL 2004*, Barcelona, Spain.

Knight, Kevin and Ishwar Chander. 1994. Automated postediting of documents. In *Proceedings of AAAI-94*, Seattle, Washington.

Lee, John and Stephanie Seneff. 2008. Correcting misuse of verb forms. *Proceedings of the 46th ACL, Columbus*.

Liu, Ting, Ming Zhou, Jianfeng Gao, Endong Xun, and Changning Huang. 2000. PENS: a machine-aided english writing system for chinese users. In *Proceedings of the 38th ACL*, Hong Kong, China.

Lopez, Adam. 2008. Statistical machine translation. *ACM Computing Surveys*, 40(3), September.

Melamed, I. Dan, Giorgio Satta, and Ben Wellington. 2004. Generalized multitext grammars. In *Proceedings of the 42nd ACL*, Barcelona, Spain.

Nagata, Ryo, Atsuo Kawai, Koichiro Morihiro, and Naoki Isu. 2006. A feedback-augmented method for detecting errors in the writing of learners of english. In *Proceedings of COLING-ACL 2006*, Sydney, Australia, July.

Och, Franz Josef and Hermann Ney. 2003. A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1).

Smith, David A. and Jason Eisner. 2006. Quasi-synchronous grammars: Alignment by soft projection of syntactic dependencies. In *Proceedings on the Workshop on Statistical Machine Translation*, New York City, June.

Smith, David A. and Jason Eisner. 2009. Parser adaptation and projection with quasi-synchronous grammar features. In *Proceedings of EMNLP 2009*, Singapore, August.

Wang, Mengqiu, Noah A. Smith, and Teruko Mitamura. 2007. What is the Jeopardy model? a quasi-synchronous grammar for QA. In *Proceedings of EMNLP-CoNLL 2007*, Prague, Czech Republic, June.

Wu, Dekai. 1995. Stochastic inversion transduction grammars, with application to segmentation, bracketing, and alignment of parallel corpora. In *Proc. of the 14th Intl. Joint Conf. on Artificial Intelligence*, Montreal, Aug.

Yamada, Kenji and Kevin Knight. 2001. A syntax-based statistical translation model. In *Proceedings of the 39th ACL*, Toulouse, France.

Yi, Xing, Jianfeng Gao, and William B Dolan. 2008. A web-based english proofing system for english as a second language users. In *Proceedings of IJCNLP*, Hyderabad, India.