Grammatical Error Correction Using Integer Linear Programming

Yuanbin Wu
Department of Computer Science
National University of Singapore
13 Computing Drive
Singapore 117417
wuyb@comp.nus.edu.sg

Hwee Tou Ng
Department of Computer Science
National University of Singapore
13 Computing Drive
Singapore 117417
nght@comp.nus.edu.sg

Abstract

We propose a joint inference algorithm for grammatical error correction. Different from most previous work where different error types are corrected independently, our proposed inference process considers all possible errors in a unified framework. We use integer linear programming (ILP) to model the inference process, which can easily incorporate both the power of existing error classifiers and prior knowledge on grammatical error correction. Experimental results on the Helping Our Own shared task show that our method is competitive with state-of-the-art systems.

1 Introduction

Grammatical error correction is an important task of natural language processing (NLP). It has many potential applications and may help millions of people who learn English as a second language (ESL). As a research field, it faces the challenge of processing ungrammatical language, which is different from other NLP tasks. The task has received much attention in recent years, and was the focus of two shared tasks on grammatical error correction in 2011 and 2012 (Dale and Kilgarriff, 2011; Dale et al., 2012).

To detect and correct grammatical errors, two different approaches are typically used — knowledge engineering or machine learning. The first relies on handcrafting a set of rules. For example, the superlative adjective *best* is preceded by the article *the*. In contrast, the machine learning approach formulates the task as a classification problem based on learning from training data. For example, an article classifier takes a noun phrase (NP) as input and predicts its article using class labels *a/an*, *the*, or ϵ (no article).

Both approaches have their advantages and disadvantages. One can readily handcraft a set of rules to incorporate various prior knowledge from grammar books and dictionaries, but rules often have exceptions and it is difficult to build rules for all grammatical errors. On the other hand, the machine learning approach can learn from texts written by ESL learners where grammatical errors have been annotated. However, training data may be noisy and classifiers may need prior knowledge to guide their predictions.

Another consideration in grammatical error correction is how to deal with multiple errors in an input sentence. Most previous work deals with errors individually: different classifiers (or rules) are developed for different types of errors (article classifier, preposition classifier, etc). Classifiers are then deployed independently. An example is a pipeline system, where each classifier takes the output of the previous classifier as its input and proposes corrections of one error type.

One problem of this pipeline approach is that the relations between errors are ignored. For example, assume that an input sentence contains *a cats*. An article classifier may propose to delete *a*, while a noun number classifier may propose to change *cats* to *cat*. A pipeline approach will choose one of the two corrections based purely on which error classifier is applied first. Another problem is that when applying a classifier, the surrounding words in the context are assumed to be correct, which is not true if grammatical errors appear close to each other in a sentence.

In this paper, we formulate grammatical error correction as a task suited for joint inference. Given an input sentence, different types of errors are jointly corrected as follows. For every possible error correction, we assign a score which measures how grammatical the resulting sentence is if the correction is accepted. We then choose a set of corrections which will result in a corrected sentence that is judged to be the most grammatical.

The inference problem is solved by integer lin-

Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics, pages 1456–1465,
Sofia, Bulgaria, August 4-9 2013.

ear programming (ILP). Variables of ILP are indicators of possible grammatical error corrections, the objective function aims to select the best set of corrections, and the constraints help to enforce a valid and grammatical output. Furthermore, ILP not only provides a method to solve the inference problem, but also allows for a natural integration of grammatical constraints into a machine learning approach. We will show that ILP fully utilizes individual error classifiers, while prior knowledge on grammatical error correction can be easily expressed using linear constraints. We evaluate our proposed ILP approach on the test data from the Helping Our Own (HOO) 2011 shared task (Dale and Kilgarriff, 2011). Experimental results show that the ILP formulation is competitive with state-of-the-art grammatical error correction systems.

The remainder of this paper is organized as follows. Section 2 gives the related work. Section 3 introduces a basic ILP formulation. Sections 4 and 5 improve the basic ILP formulation with more constraints and second order variables, respectively. Section 6 presents the experimental results. Section 7 concludes the paper.

2 Related Work

The knowledge engineering approach has been used in early grammatical error correction systems (Murata and Nagao, 1993; Bond et al., 1995; Bond and Ikehara, 1996; Heine, 1998). However, as noted by (Han et al., 2006), rules usually have exceptions, and it is hard to utilize corpus statistics in handcrafted rules. As such, the machine learning approach has become the dominant approach in grammatical error correction.

Previous work in the machine learning approach typically formulates the task as a classification problem. Article and preposition errors are the two main research topics (Knight and Chander, 1994; Han et al., 2006; Tetreault and Chodorow, 2008; Dahlmeier and Ng, 2011). Features used in classification include surrounding words, part-of-speech tags, language model scores (Gamon, 2010), and parse tree structures (Tetreault et al., 2010). Learning algorithms used include maximum entropy (Han et al., 2006; Tetreault and Chodorow, 2008), averaged perceptron, naïve Bayes (Rozovskaya and Roth, 2011), etc. Besides article and preposition errors, verb form errors also attract some attention recently (Liu et al., 2010; Tajiri et al., 2012).

Several research efforts have started to deal with correcting different errors in an integrated manner (Gamon, 2011; Park and Levy, 2011; Dahlmeier and Ng, 2012a). Gamon (2011) uses a high-order sequential labeling model to detect various errors. Park and Levy (2011) models grammatical error correction using a noisy channel model, where a predefined generative model produces correct sentences and errors are added through a noise model. The work of (Dahlmeier and Ng, 2012a) is probably the closest to our current work. It uses a beam-search decoder, which iteratively corrects an input sentence to arrive at the best corrected output. The difference between their work and our ILP approach is that the beam-search decoder returns an approximate solution to the original inference problem, while ILP returns an exact solution to an approximate inference problem.

Integer linear programming has been successfully applied to many NLP tasks, such as dependency parsing (Riedel and Clarke, 2006; Martins et al., 2009), semantic role labeling (Punyakanok et al., 2005), and event extraction (Riedel and McCallum, 2011).

3 Inference with First Order Variables

The inference problem for grammatical error correction can be stated as follows: *"Given an input sentence, choose a set of corrections which results in the best output sentence."* In this paper, this problem will be expressed and solved by integer linear programming (ILP).

To express an NLP task in the framework of ILP requires the following steps:

1. Encode the output space of the NLP task using integer variables;
2. Express the inference objective as a linear objective function; and
3. Introduce problem-specific constraints to refine the feasible output space.

In the following sections, we follow the above formulation. For the grammatical error correction task, the variables in ILP are indicators of the corrections that a word needs, the objective function measures how grammatical the whole sentence is if some corrections are accepted, and the constraints guarantee that the corrections do not conflict with each other.

3.1 First Order Variables

Given an input sentence, the main question that a grammatical error correction system needs to answer is: *What corrections at which positions?* For example, is it reasonable to change the word *cats* to *cat* in the sentence *A cats sat on the mat*? Given the corrections at various positions in a sentence, the system can readily come up with the corrected sentence. Thus, a natural way to encode the output space of grammatical error correction requires information about sentence position, error type (e.g., noun number error), and correction (e.g., *cat*).

Suppose s is an input sentence, and $|s|$ is its length (i.e., the number of words in s). Define *first order variables*:

$$Z_{l,p}^{k} \in \{0, 1\}, \tag{1}$$

where

$p \in \{1, 2, \ldots, |s|\}$ is a position in a sentence,
$l \in L$ is an error type,
$k \in \{1, 2, \ldots, C(l)\}$ is a correction of type l.
L: the set of error types,
$C(l)$: the number of corrections for error type l.

If $Z_{l,p}^{k} = 1$, the word at position p should be corrected to k that is of error type l. Otherwise, the word at position p is not applicable for this correction. Deletion of a word is represented as $k = \epsilon$. For example, $Z_{\text{Art},1}^{\text{a}} = 1$ means that the article (Art) at position 1 of the sentence should be *a*. If $Z_{\text{Art},1}^{\text{a}} = 0$, then the article should not be *a*. Table 1 contains the error types handled in this work, their possible corrections and applicable positions in a sentence.

3.2 The Objective Function

The objective of the inference problem is to find the best output sentence. However, there are exponentially many different combinations of corrections, and it is not possible to consider all combinations. Therefore, instead of solving the original inference problem, we will solve an approximate inference problem by introducing the following decomposable assumption: *Measuring the output quality of multiple corrections can be decomposed into measuring the quality of the individual corrections.*

Let s' be the resulting sentence if the correction $Z_{l,p}^{k}$ is accepted for s, or for simplicity denoting it as $s \xrightarrow{Z_{l,p}^{k}} s'$. Let $w_{l,p,k} \in \mathbb{R}$, measure how grammatical s' is. Define the objective function as

$$\max \quad \sum_{l,p,k} w_{l,p,k} Z_{l,p}^{k}.$$

This linear objective function aims to select a set of $Z_{l,p}^{k}$, such that the sum of their weights is the largest among all possible candidate corrections, which in turn gives the most grammatical sentence under the decomposable assumption.

Although the decomposable assumption is a strong assumption, it performs well in practice, and one can relax the assumption by using higher order variables (see Section 5).

For an individual correction $Z_{l,p}^{k}$, we measure the quality of s' based on three factors:

1. The language model score $h(s', \texttt{LM})$ of s' based on a large web corpus;

2. The confidence scores $f(s', t)$ of classifiers, where $t \in E$ and E is the set of classifiers. For example, an article classifier trained on well-written documents will score every article in s', and measure the quality of s' from the perspective of an article "expert".

3. The disagreement scores $g(s', t)$ of classifiers, where $t \in E$. A disagreement score measures how ungrammatical s' is from the perspective of a classifier. Take the article classifier as an example. For each article instance in s', the classifier computes the difference between the maximum confidence score among all possible choices of articles, and the confidence score of the observed article. This difference represents the disagreement on the observed article by the article classifier or "expert". Define the maximum difference over all article instances in s' to be the article classifier disagreement score of s'. In general, this score is large if the sentence s' is more ungrammatical.

The weight $w_{l,p,k}$ is a combination of these scores:

$$\begin{aligned} w_{l,p,k} \;=\; & \nu_{\texttt{LM}} h(s', \texttt{LM}) + \sum_{t \in E} \lambda_t f(s', t) \\ & + \sum_{t \in E} \mu_t g(s', t), \end{aligned} \tag{2}$$

where $\nu_{\texttt{LM}}$, λ_t, and μ_t are the coefficients.

3.3 Constraints

An observation on the objective function is that it is possible, for example, to set $Z_{\text{Art},p}^{\text{a}} = 1$ and

Type l	**Correction** k	$C(l)$	**Applicable**	**Variables**
article	a, the, ϵ	3	article or NP	$Z^{\text{a}}_{\text{Art},p}$, $Z^{\text{the}}_{\text{Art},p}$,$Z^{\epsilon}_{\text{Art},p}$
preposition	on, at, in, ...	|confusion set|	preposition	$Z^{\text{on}}_{\text{Prep},p}$, $Z^{\text{at}}_{\text{Prep},p}$, $Z^{\text{in}}_{\text{Prep},p}$, ...
noun number	singular, plural	2	noun	$Z^{\text{singular}}_{\text{Noun},p}$, $Z^{\text{plural}}_{\text{Noun},p}$
punctuation	punctuation symbols	|candidates|	determined by rules	$Z^{\text{original}}_{\text{Punct},p}$, $Z^{\text{cand1}}_{\text{Punct},p}$, $Z^{\text{cand2}}_{\text{Punct},p}$,...
spelling	correctly spelled words	|candidates|	determined by a spell checker	$Z^{\text{original}}_{\text{Spell},p}$, $Z^{\text{cand1}}_{\text{Spell},p}$, $Z^{\text{cand2}}_{\text{Spell},p}$,...

Table 1: Error types and corrections. The *Applicable* column indicates which parts of a sentence are applicable to an error type. In the first row, ϵ means deleting an article.

$Z^{\text{the}}_{\text{Art},p} = 1$, which means there are two corrections *a* and *the* for the same sentence position p, but obviously only one article is allowed.

A simple constraint to avoid these conflicts is

$$\sum_k Z^k_{l,p} = 1, \quad \forall \text{ applicable } l, p$$

It reads as follows: for each error type l, only one output k is allowed at any applicable position p (note that $Z^k_{l,p}$ is a Boolean variable).

Putting the variables, objective function, and constraints together, the ILP problem with respect to first order variables is as follows:

$$\max \quad \sum_{l,p,k} w_{l,p,k} Z^k_{l,p} \tag{3}$$

$$\text{s.t.} \quad \sum_k Z^k_{l,p} = 1, \quad \forall \text{ applicable } l, p \tag{4}$$

$$Z^k_{l,p} \in \{0, 1\} \tag{5}$$

The ILP problem is solved using lp_solve[1], an integer linear programming solver based on the revised simplex method and the branch-and-bound method for integers.

3.4 An Illustrating Example

To illustrate the ILP formulation, consider an example input sentence s:

$$\text{A cats sat on the mat .} \tag{6}$$

First, the constraint (4) at position 1 is:

$$Z^{\text{a}}_{\text{Art},1} + Z^{\text{the}}_{\text{Art},1} + Z^{\epsilon}_{\text{Art},1} = 1,$$

which means only one article in {a, the, ϵ} is selected.

[1] http://lpsolve.sourceforge.net/

Next, to compute $w_{l,p,k}$, we collect language model score and confidence scores from the article (ART), preposition (PREP), and noun number (NOUN) classifier, i.e., $E = \{\text{ART}, \text{PREP}, \text{NOUN}\}$.

The weight for $Z^{\text{singular}}_{\text{Noun},2}$ is:

$$\begin{aligned} w_{\text{Noun},2,\text{singular}} &= \nu_{\text{LM}} h(s', \text{LM}) + \\ &\lambda_{\text{ART}} f(s', \text{ART}) + \lambda_{\text{PREP}} f(s', \text{PREP}) + \lambda_{\text{NOUN}} f(s', \text{NOUN}) + \\ &\mu_{\text{ART}} g(s', \text{ART}) + \mu_{\text{PREP}} g(s', \text{PREP}) + \mu_{\text{NOUN}} g(s', \text{NOUN}). \end{aligned}$$

where $s \xrightarrow{Z^{\text{singular}}_{\text{Noun},2}} s' = \text{A } \textit{cat} \text{ sat on the mat .}$

The confidence score $f(s', t)$ of classifier t is the average of the confidence scores of t on the applicable instances in s'.

For example, there are two article instances in s', located at position 1 and 5 respectively, hence,

$$\begin{aligned} f(s', \text{ART}) &= \frac{1}{2}\big(f(s'[1], 1, \text{ART}) + f(s'[5], 5, \text{ART})\big) \\ &= \frac{1}{2}\big(f(\text{a}, 1, \text{ART}) + f(\text{the}, 5, \text{ART})\big). \end{aligned}$$

Here, the symbol $f_t(s'[p], p, \text{ART})$ refers to the confidence score of the article classifier at position p, and $s'[p]$ is the word at position p of s'.

Similarly, the disagreement score $g(s', \text{ART})$ of the article classifier is

$$\begin{aligned} g(s', \text{ART}) &= \max(g_1, g_2) \\ g_1 &= \arg\max_k f(k, 1, \text{ART}) - f(\text{a}, 1, \text{ART}) \\ g_2 &= \arg\max_k f(k, 5, \text{ART}) - f(\text{the}, 5, \text{ART}) \end{aligned}$$

Putting them together, the weight for $Z^{\text{singular}}_{\text{Noun},2}$ is:

$$\begin{aligned} w_{\text{Noun},2,\text{singular}} &= \nu_{\text{LM}} h(s', \text{LM}) \\ &+ \frac{\lambda_{\text{ART}}}{2}\big(f(\text{a}, 1, \text{ART}) + f(\text{the}, 5, \text{ART})\big) \\ &+ \lambda_{\text{PREP}} f(\text{on}, 4, \text{PREP}) \\ &+ \frac{\lambda_{\text{NOUN}}}{2}\big(f(\text{cat}, 2, \text{NOUN}) + f(\text{mat}, 6, \text{NOUN})\big) \\ &+ \mu_{\text{ART}} g(s', \text{ART}) \\ &+ \mu_{\text{PREP}} g(s', \text{PREP}) \\ &+ \mu_{\text{NOUN}} g(s', \text{NOUN}) \end{aligned}$$

Input	**A**	**cats**	**sat**	**on**	**the**	**mat**
Corrections	The, ϵ	cat		at, in	a, ϵ	mats
Variables	$Z^{\text{a}}_{\text{Art},1}$	$Z^{\text{singular}}_{\text{Noun},2}$		$Z^{\text{on}}_{\text{Prep},4}$	$Z^{\text{a}}_{\text{Art},5}$	$Z^{\text{singular}}_{\text{Noun},6}$
	$Z^{\text{the}}_{\text{Art},1}$	$Z^{\text{plural}}_{\text{Noun},2}$		$Z^{\text{at}}_{\text{Prep},4}$	$Z^{\text{the}}_{\text{Art},5}$	$Z^{\text{plural}}_{\text{Noun},6}$
	$Z^{\epsilon}_{\text{Art},1}$			$Z^{\text{in}}_{\text{Prep},4}$	$Z^{\epsilon}_{\text{Art},5}$	

Table 2: The possible corrections on example (6).

3.5 Complexity

The time complexity of ILP is determined by the number of variables and constraints. Assume that for each sentence position, at most K classifiers are applicable[2]. The number of variables is $O(K|s|C(l^*))$, where $l^* = \arg\max_{l \in L} C(l)$. The number of constraints is $O(K|s|)$.

4 Constraints for Prior Knowledge

4.1 Modification Count Constraints

In practice, we usually have some rough gauge of the quality of an input sentence. If an input sentence is mostly grammatical, the system is expected to make few corrections. This requirement can be easily satisfied by adding modification count constraints.

In this work, we constrain the number of modifications according to error types. For the error type l, a parameter N_l controls the number of modifications allowed for type l. For example, the modification count constraint for article corrections is

$$\sum_{p,k} Z^k_{\text{Art},p} \leq N_{\text{Art}}, \quad \text{where } k \neq s[p]. \tag{7}$$

The condition ensures that the correction k is different from the original word in the input sentence. Hence, the summation only counts real modifications. There are similar constraints for preposition, noun number, and spelling corrections:

$$\sum_{p,k} Z^k_{\text{Prep},p} \leq N_{\text{Prep}}, \quad \text{where } k \neq s[p], \tag{8}$$

$$\sum_{p,k} Z^k_{\text{Noun},p} \leq N_{\text{Noun}}, \quad \text{where } k \neq s[p], \tag{9}$$

$$\sum_{p,k} Z^k_{\text{Spell},p} \leq N_{\text{Spell}}, \quad \text{where } k \neq s[p]. \tag{10}$$

[2]In most cases, $K = 1$. An example of $K > 1$ is a noun that requires changing the word form (between singular and plural) and inserting an article, for which $K = 2$.

4.2 Article-Noun Agreement Constraints

An advantage of the ILP formulation is that it is relatively easy to incorporate prior linguistic knowledge. We now take article-noun agreement as an example to illustrate how to encode such prior knowledge using linear constraints.

A noun in plural form cannot have *a* (or *an*) as its article. That two Boolean variables Z_1 and Z_2 are mutually exclusive can be handled using a simple inequality $Z_1 + Z_2 \leq 1$. Thus, the following inequality correctly enforces article-noun agreement:

$$Z^{\text{a}}_{\text{Art},p_1} + Z^{\text{plural}}_{\text{Noun},p_2} \leq 1, \tag{11}$$

where the article at p_1 modifies the noun at p_2.

4.3 Dependency Relation Constraints

Another set of constraints involves dependency relations, including subject-verb relation and determiner-noun relation. Specifically, for a noun n at position p, we check the word w related to n via a child-parent or parent-child relation. If w belongs to a set of verbs or determiners (*are, were, these, all*) that takes a plural noun, then the noun n is required to be in plural form by adding the following constraint:

$$Z^{\text{plural}}_{\text{Noun},p} = 1. \tag{12}$$

Similarly, if a noun n at position p is required to be in singular form due to subject-verb relation or determiner-noun relation, we add the following constraint:

$$Z^{\text{singular}}_{\text{Noun},p} = 1. \tag{13}$$

5 Inference with Second Order Variables

5.1 Motivation and Definition

To relax the decomposable assumption in Section 3.2, instead of treating each correction separately, one can combine multiple corrections into a single correction by introducing higher order variables.

Consider the sentence *A cat sat on the mat.* When measuring the gain due to $Z^{\text{plural}}_{\text{Noun},2} = 1$ (change *cat* to *cats*), the weight $w_{\text{Noun},2,\text{plural}}$ is likely to be small since *A cats* will get a low language model score, a low article classifier confidence score, and a low noun number classifier confidence score. Similarly, the weight $w_{\text{Art},1,\epsilon}$ of $Z^{\epsilon}_{\text{Art},1}$ (delete article *A*) is also likely to be small because of the missing article. Thus, if one considers the two corrections separately, they are both unlikely to appear in the final corrected output.

However, the correction from *A cat sat on the mat.* to *Cats sat on the mat.* should be a reasonable candidate, especially if the context indicates that there are many cats (more than one) on the mat. Due to treating corrections separately, it is difficult to deal with multiple interacting corrections with only first order variables.

In order to include the correction ϵ *Cats*, one can use a new set of variables, *second order variables*. To keep symbols clear, let $Z = \{Z_u | Z_u = Z^k_{l,p}, \forall l, p, k\}$ be the set of first order variables, and $w_u = w_{l,p,k}$ be the weight of $Z_u = Z^k_{l,p}$. Define a second order variable $X_{u,v}$:

$$X_{u,v} = Z_u \wedge Z_v, \tag{14}$$

where Z_u and Z_v are first order variables:

$$Z_u \triangleq Z^{k_1}_{l_1,p_1}, \quad Z_v \triangleq Z^{k_2}_{l_2,p_2}. \tag{15}$$

The definition of $X_{u,v}$ states that a second order variable is set to 1 if and only if its two component first order variables are both set to 1. Thus, it combines two corrections into a single correction. In the above example, a second order variable is introduced:

$$X_{u,v} = Z^{\epsilon}_{\text{Art},1} \wedge Z^{\text{plural}}_{\text{Noun},2},$$
$$s \xrightarrow{X_{u,v}} s' = \textit{Cats} \text{ sat on the mat .}$$

Similar to first order variables, let $w_{u,v}$ be the weight of $X_{u,v}$. Note that definition (2) only depends on the output sentence s', and the weight of the second order variable $w_{u,v}$ can be defined in the same way:

$$\begin{aligned} w_{u,v} &= \nu_{\text{LM}} h(s', \text{LM}) + \sum_{t \in E} \lambda_t f(s', t) \\ &\quad + \sum_{t \in E} \mu_t g(s', t). \end{aligned} \tag{16}$$

5.2 ILP with Second Order Variables

A set of new constraints is needed to enforce consistency between the first and second order variables. These constraints are the linearization of definition (14) of $X_{u,v}$:

$$X_{u,v} = Z_u \wedge Z_v \Leftrightarrow \begin{array}{ll} X_{u,v} \leq & Z_u \\ X_{u,v} \leq & Z_v \\ X_{u,v} \geq & Z_u + Z_v - 1 \end{array} \tag{17}$$

A new objective function combines the weights from both first and second order variables:

$$\max \quad \sum_{l,p,k} w_{l,p,k} Z^k_{l,p} + \sum_{u,v} w_{u,v} X_{u,v}. \tag{18}$$

In our experiments, due to noisy data, some weights of second order variables are small, even if both of its first order variables have large weights and satisfy all prior knowledge constraints. They will affect ILP proposing good corrections. We find that the performance will be better if we change the weights of second order variables to $w'_{u,v}$, where

$$w'_{u,v} \triangleq \max\{w_{u,v}, w_u, w_v\}. \tag{19}$$

Putting them together, (20)-(25) is an ILP formulation using second order variables, where X is the set of all second order variables which will be explained in the next subsection.

$$\max \quad \sum_{l,p,k} w_{l,p,k} Z^k_{l,p} + \sum_{u,v} w'_{u,v} X_{u,v} \tag{20}$$

$$\text{s.t.} \quad \sum_{k} Z^k_{l,p} = 1, \quad \forall \text{ applicable } l, p \tag{21}$$

$$X_{u,v} \leq Z_u, \tag{22}$$

$$X_{u,v} \leq Z_v, \tag{23}$$

$$X_{u,v} \geq Z_u + Z_v - 1, \forall X_{u,v} \in X \tag{24}$$

$$X_{u,v}, Z^k_{l,p} \in \{0, 1\} \tag{25}$$

5.3 Complexity and Variable Selection

Using the notation in section 3.5, the number of second order variables is $O(|Z|^2) = O(K^2|s|^2C(l^*)^2)$ and the number of constraints is $O(K^2|s|^2C(l^*)^2)$. More generally, for variables with higher order $h \geq 2$, the number of variables (and constraints) is $O(K^h|s|^hC(l^*)^h)$.

Note that both the number of variables and the number of constraints increase exponentially with increasing variable order. In practice, a small subset of second order variables is sufficient to

Data set	Sentences	Words	Edits
Dev set	939	22,808	1,264
Test set	722	18,790	1,057

Table 3: Overview of the HOO 2011 data sets. Corrections are called *edits* in the HOO 2011 shared task.

achieve good performance. For example, noun number corrections are only coupled with nearby article corrections, and have no connection with distant or other types of corrections.

In this work, we only introduce second order variables that combine article corrections and noun number corrections. Furthermore, we require that the article and the noun be in the same noun phrase. The set X of second order variables in Equation (24) is defined as follows:

$$X = \{X_{u,v} = Z_u \wedge Z_v | l_1 = \text{Art}, l_2 = \text{Noun}, \\ s[p_1], s[p_2] \text{ are in the same noun phrase}\},$$

where l_1, l_2, p_1, p_2 are taken from Equation (15).

6 Experiments

Our experiments mainly focus on two aspects: how our ILP approach performs compared to other grammatical error correction systems; and how the different constraints and the second order variables affect the ILP performance.

6.1 Evaluation Corpus and Metric

We follow the evaluation setup in the HOO 2011 shared task on grammatical error correction (Dale and Kilgarriff, 2011). The development set and test set in the shared task consist of conference and workshop papers taken from the Association for Computational Linguistics (ACL). Table 3 gives an overview of the data sets.

System performance is measured by precision, recall, and F measure:

$$P = \frac{\text{\# true edits}}{\text{\# system edits}}, \; R = \frac{\text{\# true edits}}{\text{\# gold edits}}, \; F = \frac{2PR}{P+R}. \tag{26}$$

The difficulty lies in how to generate the system edits from the system output. In the HOO 2011 shared task, participants can submit system edits directly or the corrected plain-text system output. In the latter case, the official HOO scorer will extract system edits based on the original (ungrammatical) input text and the corrected system output text, using GNU Wdiff[3].

Consider an input sentence *The data is similar with test set.* taken from (Dahlmeier and Ng, 2012a). The gold-standard edits are *with* $\rightarrow$ *to* and $\epsilon \rightarrow$ *the*. That is, the grammatically correct sentence should be *The data is similar to the test set.* Suppose the corrected output of a system to be evaluated is exactly this perfectly corrected sentence *The data is similar to the test set.* However, the official HOO scorer using GNU Wdiff will automatically extract only one system edit *with* $\rightarrow$ *to the* for this system output. Since this single system edit does not match any of the two gold-standard edits, the HOO scorer returns an F measure of 0, even though the system output is perfectly correct.

In order to overcome this problem, the *Max-Match* (M^2) scorer was proposed in (Dahlmeier and Ng, 2012b). Given a set of gold-standard edits, the original (ungrammatical) input text, and the corrected system output text, the M^2 scorer searches for the system edits that have the largest overlap with the gold-standard edits. For the above example, the system edits automatically determined by the M^2 scorer are identical to the gold-standard edits, resulting in an F measure of 1 as we would expect. We will use the M^2 scorer in this paper to determine the best system edits. Once the system edits are found, P, R, and F are computed using the standard definition (26).

6.2 ILP Configuration

6.2.1 Variables

The first order variables are given in Table 1. If the indefinite article correction *a* is chosen, then the final choice between *a* and *an* is decided by a rule-based post-processing step. For each preposition error variable $Z^k_{\text{Prep},p}$, the correction k is restricted to a pre-defined confusion set of prepositions which depends on the observed preposition at position p. For example, the confusion set of *on* is { *at, for, in, of* }. The list of prepositions corrected by our system is *about, among, at, by, for, in, into, of, on, over, to, under, with,* and *within*. Only selected positions in a sentence (determined by rules) undergo punctuation correction. The spelling correction candidates are given by a spell checker. We used GNU Aspell[4] in our work.

[3] http://www.gnu.org/software/wdiff/

[4] http://aspell.net

6.2.2 Weights

As described in Section 3.2, the weight of each variable is a linear combination of the language model score, three classifier confidence scores, and three classifier disagreement scores. We use the Web 1T 5-gram corpus (Brants and Franz, 2006) to compute the language model score for a sentence. Each of the three classifiers (article, preposition, and noun number) is trained with the multi-class confidence weighted algorithm (Crammer et al., 2009). The training data consists of all non-OCR papers in the ACL Anthology[5], minus the documents that overlap with the HOO 2011 data set. The features used for the classifiers follow those in (Dahlmeier and Ng, 2012a), which include lexical and part-of-speech n-grams, lexical head words, web-scale n-gram counts, dependency heads and children, etc. Over 5 million training examples are extracted from the ACL Anthology for use as training data for the article and noun number classifiers, and over 1 million training examples for the preposition classifier.

Finally, the language model score, classifier confidence scores, and classifier disagreement scores are normalized to take values in $[0, 1]$, based on the HOO 2011 development data. We use the following values for the coefficients: $\nu_{\text{LM}} = 1$ (language model); $\lambda_t = 1$ (classifier confidence); and $\mu_t = -1$ (classifier disagreement).

6.2.3 Constraints

In Section 4, three sets of constraints are introduced: modification count (MC), article-noun agreement (ANA), and dependency relation (DR) constraints. The values for the modification count parameters are set as follows: $N_{\text{Art}} = 3, N_{\text{Prep}} = 2, N_{\text{Noun}} = 2$, and $N_{\text{Spell}} = 1$.

6.3 Experimental Results

We compare our ILP approach with two other systems: the beam search decoder of (Dahlmeier and Ng, 2012a) which achieves the best published performance to date on the HOO 2011 data set, and UI Run1 (Rozovskaya et al., 2011) which achieves the best performance among all participating systems at the HOO 2011 shared task. The results are given in Table 4.

The HOO 2011 shared task provides two sets of gold-standard edits: the original gold-standard edits produced by the annotator, and the official gold-standard edits which incorporated corrections proposed by the HOO 2011 shared task participants. All three systems listed in Table 4 use the M^2 scorer to extract system edits. The results of the beam search decoder and UI Run1 are taken from Table 2 of (Dahlmeier and Ng, 2012a).

[5]http://aclweb.org/anthology-new/

System	Original			Official		
	P	R	F	P	R	F
UI Run1	40.86	11.21	17.59	54.61	14.57	23.00
Beam search	30.28	19.17	23.48	33.59	20.53	25.48
ILP	20.54	27.93	23.67	21.99	29.04	25.03

Table 4: Comparison of three grammatical error correction systems.

Overall, ILP inference outperforms UI Run1 on both the original and official gold-standard edits, and the improvements are statistically significant at the level of significance 0.01. The performance of ILP inference is also competitive with the beam search decoder. The results indicate that a grammatical error correction system benefits from corrections made at a whole sentence level, and that joint correction of multiple error types achieves state-of-the-art performance.

Table 5 provides the comparison of the beam search decoder and ILP inference in detail. The main difference between the two is that, except for spelling errors, ILP inference gives higher recall than the beam search decoder, while its precision is lower. This indicates that ILP inference is more aggressive in proposing corrections.

Next, we evaluate ILP inference in different configurations. We only focus on article and noun number error types. Table 6 shows the performance of ILP in different configurations. From the results, MC and DR constraints improve precision, indicating that the two constraints can help to restrict the number of erroneous corrections. Including second order variables gives the best F measure, which supports our motivation for introducing higher order variables.

Adding article-noun agreement constraints (ANA) slightly decreases performance. By examining the output, we find that although the overall performance worsens slightly, the agreement requirement is satisfied. For example, for the input *We utilize search engine to ...*, the output without ANA is *We utilize* a *search* engines *to ...* but with ANA is *We utilize* the *search* engines *to ...*, while the only gold edit inserts *a*.

Error type	Original						Official					
	Beam search			ILP			Beam search			ILP		
	P	R	F	P	R	F	P	R	F	P	R	F
Spelling	36.84	0.69	1.35	60.00	0.59	1.17	36.84	0.66	1.30	60.00	0.57	1.12
+ Article	19.84	12.59	15.40	18.54	14.75	16.43	22.45	13.72	17.03	20.37	15.61	17.68
+ Preposition	22.62	14.26	17.49	17.61	18.58	18.09	24.84	15.14	18.81	19.24	19.68	19.46
+ Punctuation	24.27	18.09	20.73	20.52	23.50	21.91	27.13	19.58	22.75	22.49	24.98	23.67
+ Noun number	30.28	19.17	23.48	20.54	27.93	23.67	33.59	20.53	25.48	21.99	29.04	25.03

Table 5: Comparison of the beam search decoder and ILP inference. ILP is equipped with all constraints (MC, ANA, DR) and default parameters. Second order variables related to article and noun number error types are also used in the last row.

Setting	Original			Official		
	P	R	F	P	R	F
Art+Nn, 1^{st} ord.	17.19	19.37	18.22	18.59	20.44	19.47
+ MC	17.87	18.49	18.17	19.23	19.39	19.31
+ ANA	17.78	18.39	18.08	19.04	19.11	19.07
+ DR	17.95	18.58	18.26	19.23	19.30	19.26
+ 2^{nd} ord.	18.75	18.88	18.81	20.04	19.58	19.81

Table 6: The effects of different constraints and second order variables.

7 Conclusion

In this paper, we model grammatical error correction as a joint inference problem. The inference problem is solved using integer linear programming. We provide three sets of constraints to incorporate additional linguistic knowledge, and introduce a further extension with second order variables. Experiments on the HOO 2011 shared task show that ILP inference achieves state-of-the-art performance on grammatical error correction.

Acknowledgments

This research is supported by the Singapore National Research Foundation under its International Research Centre @ Singapore Funding Initiative and administered by the IDM Programme Office.

References

Francis Bond and Satoru Ikehara. 1996. When and how to disambiguate? countability in machine translation. In *Proceedings of the International Seminar on Multimodal Interactive Disambiguation*.

Francis Bond, Kentaro Ogura, and Tsukasa Kawaoka. 1995. Noun phrase reference in Japanese-to-English machine translation. In *Proceedings of the 6th International Conference on Theoretical and Methodological Issues in Machine Translation*.

Thorsten Brants and Alex Franz. 2006. Web 1T 5-gram corpus version 1.1. Technical report, Google Research.

Koby Crammer, Mark Dredze, and Alex Kulesza. 2009. Multi-class confidence weighted algorithms. In *Proceedings of EMNLP*.

Daniel Dahlmeier and Hwee Tou Ng. 2011. Grammatical error correction with alternating structure optimization. In *Proceedings of ACL*.

Daniel Dahlmeier and Hwee Tou Ng. 2012a. A beam-search decoder for grammatical error correction. In *Proceedings of EMNLP*.

Daniel Dahlmeier and Hwee Tou Ng. 2012b. Better evaluation for grammatical error correction. In *Proceedings of NAACL*.

Robert Dale and Adam Kilgarriff. 2011. Helping Our Own: The HOO 2011 pilot shared task. In *Proceedings of the 13th European Workshop on Natural Language Generation*.

Robert Dale, Ilya Anisimoff, and George Narroway. 2012. HOO 2012: A report on the preposition and determiner error correction shared task. In *Proceedings of the Seventh Workshop on Innovative Use of NLP for Building Educational Applications*, pages 54–62.

Michael Gamon. 2010. Using mostly native data to correct errors in learners' writing. In *Proceedings of NAACL*.

Michael Gamon. 2011. High-order sequence modeling for language learner error detection. In *Proceedings of the Sixth Workshop on Innovative Use of NLP for Building Educational Applications*.

Na-Rae Han, Martin Chodorow, and Claudia Leacock. 2006. Detecting errors in English article usage by non-native speakers. *Natural Language Engineering*, 12(2).

Julia Heine. 1998. Definiteness predictions for Japanese noun phrases. In *Proceedings of ACL-COLING*.

Kevin Knight and Ishwar Chander. 1994. Automated postediting of documents. In *Proceedings of AAAI*.

Xiaohua Liu, Bo Han, Kuan Li, Stephan Hyeonjun Stiller, and Ming Zhou. 2010. SRL-based verb selection for ESL. In *Proceedings of EMNLP*.

Andre Martins, Noah Smith, and Eric Xing. 2009. Concise integer linear programming formulations for dependency parsing. In *Proceedings of ACL-IJCNLP*.

Masaki Murata and Makoto Nagao. 1993. Determination of referential property and number of nouns in Japanese sentences for machine translation into English. In *Proceedings of the 5th International Conference on Theoretical and Methodological Issues in Machine Translation*.

Y. Albert Park and Roger Levy. 2011. Automated whole sentence grammar correction using a noisy channel model. In *Proceedings of ACL*.

Vasin Punyakanok, Dan Roth, Wen tau Yih, and Dav Zimak. 2005. Learning and inference over constrained output. In *Proceedings of IJCAI*.

Sebastian Riedel and James Clarke. 2006. Incremental integer linear programming for non-projective dependency parsing. In *Proceedings of EMNLP*.

Sebastian Riedel and Andrew McCallum. 2011. Fast and robust joint models for biomedical event extraction. In *Proceedings of EMNLP*.

Alla Rozovskaya and Dan Roth. 2011. Algorithm selection and model adaptation for ESL correction tasks. In *Proceedings of ACL*.

Alla Rozovskaya, Mark Sammons, Joshua Gioja, and Dan Roth. 2011. University of Illinois system in HOO text correction shared task. In *Proceedings of the 13th European Workshop on Natural Language Generation*.

Toshikazu Tajiri, Mamoru Komachi, and Yuji Matsumoto. 2012. Tense and aspect error correction for ESL learners using global context. In *Proceedings of ACL*.

Joel R. Tetreault and Martin Chodorow. 2008. The ups and downs of preposition error detection in ESL writing. In *Proceedings of COLING*.

Joel Tetreault, Jennifer Foster, and Martin Chodorow. 2010. Using parse features for preposition selection and error detection. In *Proceedings of ACL*.