AUTOMATED METHODS FOR TEXT CORRECTION

BY

ALLA ROZOVSKAYA

DISSERTATION

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in Linguistics
in the Graduate College of the
University of Illinois at Urbana-Champaign, 2013

Urbana, Illinois

Abstract

Development of automatic text correction systems has a long history in natural language processing research. This thesis considers the problem of correcting writing mistakes made by non-native English speakers. We address several types of errors commonly exhibited by non-native English writers – misuse of articles, prepositions, noun number, and verb properties – and build a robust, state-of-the-art system that combines machine learning methods and linguistic knowledge.

The proposed approach is distinguished from other related work in several respects. First, several machine learning methods are compared to determine which methods are most effective for this problem. Earlier evaluations, because they are based on incomparable data sets, have questionable conclusions. Our results reverse these conclusions and pave the way for the next contribution.

Using the important observation that mistakes made by non-native writers are systematic, we develop models that utilize knowledge about error regularities with minimal annotation costs. Our approach differs from earlier ones that either built models that had no knowledge about error regularities or required a lot of annotated data.

Next, we develop special strategies for correcting errors on open-class words. These errors, while being very prevalent among non-native English speakers, are the least studied and are not well-understood linguistically. The challenges that these mistakes present are addressed in a linguistically-informed approach.

Finally, a novel global approach to error correction is proposed that considers grammatical dependencies among error types and addresses these via joint learning and joint inference.

The systems and techniques described in this thesis are evaluated empirically and competitively in the context of several shared tasks, where they have demonstrated superior performance. In particular, our system ranked first in the most prestigious competition in the natural language processing field, the CoNLL-2013 shared task on text correction. Based on the analysis of this system, four design principles that are crucial for building a state-of-the-art error correction system are identified.

Publication Notes

Some of the work in this dissertation is based on the following publications:

Chapter 5:

- A. Rozovskaya and D. Roth. "Algorithm Selection and Model Adaptation for ESL Correction Tasks." In *Proceedings of the Annual Meeting of the Association of Computational Linguistics (ACL)*, 2011.

Chapter 6:

- A. Rozovskaya and D. Roth. "Training Paradigms for Correcting Errors in Grammar and Usage." In *Proceedings of the Annual Meeting of the North American Association of Computational Linguistics (NAACL)*, 2010.
- A. Rozovskaya and D. Roth. "Generating Confusion Sets for Context-Sensitive Error Correction." In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2010.
- A. Rozovskaya and D. Roth. "Algorithm Selection and Model Adaptation for ESL Correction Tasks." In *Proceedings of the Annual Meeting of the Association of Computational Linguistics (ACL)*, 2011.
- A. Rozovskaya, M. Sammons and D. Roth. "The UI System in the HOO 2012 Shared Task on Error Correction."[1] In *Proceedings of the NAACL-HLT Workshop on Innovative Use of NLP for Building Educational Applications*, 2012.

Chapter 9:

- A. Rozovskaya and D. Roth. "Joint Learning and Inference for Grammatical Error Correction." In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2013.

Other relevant works by the author:

- A. Rozovskaya, K.-W. Chang, M. Sammons, and D. Roth. "The University of Illinois System in the CoNLL-2013 Shared Task."[2] In *Proceedings of the Conference on Computational Natural Language Learning (CoNLL)*, 2013.

[1]Our system ranked first in some metrics and second in others out of 14 teams.
[2]Our system ranked first out of 17 participating teams.

- A. Rozovskaya, M. Sammons, J. Gioja and D. Roth. "University of Illinois System in HOO Text Correction Shared Task."[3] In *Proceedings of the European Workshop on Natural Language Generation (ENLG)*, 2011.

[3]Our system ranked first out of six participating teams.

To my family.

Acknowledgments

First and foremost, I want to thank my advisor Dan Roth. He has been an amazing mentor, always supportive, optimistic, and kind. I am very grateful to Dan for giving me the opportunity to work on the project that led to my thesis, for always providing me with guidance, good advice, and encouragement. Dan is a rare person, and I learned a lot from him, both academically and personally. This dissertation would never be possible without Dan, and I cannot thank him enough.

I would also like to express my gratitude to my doctoral committee: Graeme Hirst, Julia Hockenmaier, and Jennifer Cole. I thank Graeme Hirst for detailed feedback on my thesis, his kindness, and help. I thank Jennifer Cole for providing a linguistic point of view on the ideas in this thesis, which allowed me to enrich it. I am grateful to Julia Hockenmaier for her feedback and for being very supportive throughout the whole process. Their help is greatly appreciated.

I had the privilege to be a part of Dan's Cognitive Computation Group, which gave me an opportunity to interact with amazing people and work closely with several members of CCG on exciting research projects. This has been an invaluable experience for me. I enjoyed the warm, friendly, and supportive atmosphere that Dan created and continues to foster among the members of the group, and I met extremely intelligent and unique individuals that I will miss a lot: Mark Sammons, Nick Rizzolo, Kai-Wei Chang, Josh Gioja, Vivek Srikumar, Quang Do, Jeff Pasternack, Gourab Kundu, Vinod Vydiswaran, Yuancheng Tu, Rajhans Samdani, Wei Lu, Prateek Jindal, James Clarke, Yee Seng Chan, Ming-Wei Chang, Michael Connor, Lev Ratinov, Dan Goldwasser, and Stephen Mayhew. This work has benefited tremendously from interactions with all of these people. I am especially indebted to several colleagues on this list. I am very grateful to Mark Sammons. I was fortunate to work with Mark on several projects, and I have always admired his openness, kindness, and integrity. Several ideas described in this work resulted from our collaboration. Mark has also provided invaluable feedback on chapters in this thesis and many related papers. I am very grateful to Nick Rizzolo for helpful discussions on LBJ. I also wish to thank Josh Gioja whom I have met through the text correction project and with whom I also collaborated. I am grateful to Vivek Srikumar and Kai-Wei Chang for discussions on research, for their help and collaboration. I wish to thank my colleague and friend Dan Goldwasser. I enjoyed warm and genuine conversations with Dan that made graduate life more enjoyable.

I wish to thank Eric Horn, Keely Ashman, and Dawn Cheek – the administrative team from CCG – for very friendly and amazing support they provided numerous times throughout my studies. I am also extremely grateful to Mary Ellen Fryer for all the help with the paperwork for my graduation.

During graduate school, I was fortunate to participate in many teaching and research opportunities. I would like to express my gratitude to Richard Sproat from whom I learned a lot about NLP, and to Roxana Girju who taught me about Computational Semantics. I am very grateful to Yael Gertner; I will always have good memories of working with Yael on the language acquisition project. I would also like to thank fellow students and colleagues whom I met through classes and teaching in the Linguistics Department: Vardit Caspi, Ayelet Weiss, Cecilia Alm, Suma Bhat, and Chongwon Park. I wish to thank Peter Chew for reading and giving extremely helpful feedback on this dissertation and related papers.

Outside of Illinois, I spent a summer in the NLP group at the Educational Testing Service, where I learned a lot working with Martin Chodorow and Joel Tetreault. I thank them for that. I am also grateful to many wonderful people that I met there: Michael Flor, Jill Burstein, Yoko Futagi, Nitin Madnani, and Claudia Leacock.

Last but not least, my family has always stood by me. I cannot thank them enough: my mom, dad, and my brother. They are happier than me that this thesis came to completion.

Table of Contents

List of Tables

List of Figures

List of Abbreviations

AP	Averaged Perceptron
ILP	Integer Linear Programming
LBJ	Learning-Based Java
LM	Language Model
NB	Naïve Bayes
NLP	Natural Language Processing
NP	Noun Phrase
POS	Part-of-Speech Tagging
SVM	Support Vector Machine

Chapter 1

Introduction

The task of text correction has been of interest to researchers in natural language processing and machine learning for many years. Over the last several decades, as people rely increasingly on computers for writing and editing, the need for automatic systems that can identify and correct errors not typically targeted by standard text correction tools has grown even more.

Recently, the task of text correction has shifted focus to mistakes[1] made by English as a Second Language (ESL) writers. ESL error correction is an important problem, since today the majority of people who write in English are not native English speakers (Gamon, 2011).

The increased interest in the field of correcting non-native writing in English can be observed not only from the number of papers published on the topic but also in that three competitions devoted to grammatical error correction for non-native writers have recently taken place: HOO-2011 (Dale and Kilgarriff, 2011), HOO-2012 (Dale et al., 2012), and the CoNLL-2013 shared task (Ng et al., 2013). These shared tasks have demonstrated that we are far from solving the problem even with the availability of annotated ESL data sets, large native corpora, and various natural language processing resources.

The most recent and also most prominent among these, the CoNLL-2013 shared task, covers several common ESL errors including article and preposition usage mistakes, mistakes in noun number, and various verb errors. These are illustrated in Figure 1.1, where the words containing an error are marked with an asterisk and $\varnothing$ denotes a missing word. Seventeen teams that participated in the task built a wide array of both statistical approaches – which include discriminative classifiers, language models, and statistical machine translation systems – and "knowledge"-engineered modules. Many of the systems also made use of linguistic resources, augmented training with additional annotated learner corpora, and defined high-level features that take into account syntactic

[1]The words *mistake* and *error* are used interchangeably throughout this thesis.

Nowadays **phone/phones* **has/have* many functionalities, **included/including* *∅*/a* camera and *∅/a Wi-Fi receiver.

Figure 1.1: **Examples of representative ESL errors.**

and semantic knowledge. In spite of the fact that the submitted systems incorporated similar resources and employed sophisticated models, there was a wide variation among the scores obtained by the teams, which suggests that there is not enough understanding of what works best and what elements are essential for building a state-of-the-art error correction system.

Furthermore, in all three shared tasks on ESL error correction, the participating systems performed at a level that is considered extremely low compared to performance obtained in other areas of NLP: even the best systems attained F1 scores in the range of 20-30 points. Note that the shared tasks targeted learners of various first language backgrounds, different proficiency levels and focused on different types of writing (essay writing or scientific writing). Compare these results to those from the task of co-reference resolution, arguably a very difficult high-level NLP problem that requires deep understanding of text, where systems typically obtain F1 scores in the range of 60-70 points (Pradhan et al., 2012).

1.1 Why Text Correction Is a Difficult Problem

The key reason that text correction is a difficult task is that even for non-native English speakers, the accuracy of the writing is very high, as *errors are very sparse.* Even for some of the most common types of errors such as article and preposition usage the majority of the words in these categories (over 90%) are used correctly. Because errors are so sparse, it is more difficult for a system to identify a mistake accurately and without introducing many false alarms. More specifically, to *improve* over the learner writing, an automatic system must perform at an accuracy level higher than 90%. Today, there are very few NLP tasks where systems attain this level of performance.

To understand why it is difficult to develop a system that can improve over learner text, note that the majority of ESL errors result in valid English words. Identifying and correcting such errors thus requires considering the context around the potentially erroneous word. Characterizing the surrounding context is not trivial as it may depend on various linguistic dimensions such as the surface forms of the neighboring words, their parts of speech, syntactic functions in the sentence, and so on. These contextual cues are highly variable and thus are difficult to encode via rules. For this reason, the dominant approach to ESL error correction is to use statistical models. These models typically *learn* the correct usage of a specific language phenomenon from large corpora of correct native data.

Because automatic systems determine the presence of mistakes based on the contextual cues around the target word, the context in the text to be corrected should be as similar as possible to the training data from which the system learns correct usage. However, learner data is *noisy*. The noise is manifested both in the presence of multiple mistakes, as errors tend to co-occur (see Figure 1.1), and in the presence of unusual word combinations: even when learner writing does not contain obvious infelicities, it commonly exhibits expressions and constructions that, while acceptable to native English speakers, are not likely to be produced by native speakers. In sum, the contextual cues present in the learner data are distinct from those found in correct native data and are therefore misleading. This observation is supported by the finding that word n-grams extracted from writings by ESL learners on average occur $2-3$ times less frequently in native English corpora than word n-grams obtained from a native English corpus (Rozovskaya and Roth, 2011).

When people start thinking about the problem of error correction, language models are the first intuitive solution that comes to mind. In fact, it is commonly believed that the majority of mistakes can be easily identified with a language model. In this thesis, it is shown that language models, even when they are built using very large corpora of native English data, have very limited performance. Noisy context is one reason why language modeling is not good enough for this problem: most of the relevant contextual cues simply do not occur even in very large English corpora.

In many NLP applications, including text correction, we wish to generalize beyond surface word information, but even for the most basic pre-processing tasks, such as part-of-speech (POS) tagging and shallow parsing, performance on native data is below the level of the accuracy of non-native text. Note that on ESL data pre-processing tools tend to do even worse because of the noisy data problem alluded to above (Nagata et al., 2011). This implies that the context representation generated using even the most accurate NLP tools is going to be noisier than the text the system is trying to correct.

It is clear that automatic error correction must go beyond the standard approaches of automatically inferring rules from correct native data.

1.2 Building a State-of-the-Art Text Correction System

In this thesis, we show that the ESL error correction task can be addressed through a combination of machine learning techniques and linguistic knowledge. We identify and investigate several key issues in ESL error correction that were not previously considered by researchers in this area, with a focus on several types of prominent mistakes both on closed- and open-class words that ESL learners exhibit. The proposed approaches result in superior error correction systems; in fact, the systems that implement these approaches won several error correction competitions.

The first question addressed in this thesis is that of *the choice of a machine learning algorithm* for ESL error correction: using two types of prominent ESL errors – in article and preposition usage – we compare several state-of-the-art machine learning approaches that have been applied to ESL correction tasks and determine which methods are most effective for this problem, and under what conditions. These experiments are motivated by previous studies that compared models of different training sizes and features (e.g., Gamon, 2010) and thus had questionable conclusions.

The second question that we address is motivated by the observation that ESL mistakes are not random but systematic. For example, errors may depend on the writer's first language. Previous research either trains error correction models on native English data or on annotated ESL data. Native-trained models do not take into account the specific types of errors that non-native speakers make, while models trained on annotated learner data require expensive annotation. To address this problem, this dissertation offers the idea of model *adaptation* to learner errors: the adaptation approach allows models trained on native English data to utilize knowledge about error regularities without incurring expensive annotation costs. Adapted models combine a lot of native data and small amounts of annotated learner data *in training*. Learner data is used only to estimate parameters relating to specific error patterns – thus requiring minimal annotation of ESL data – while most of the model parameters are estimated from native data that is readily available. The adaptation approach combines the advantages of training on native and annotated data and demonstrably improves error correction.

To date, research on correcting ESL mistakes has used models that assume a closed set of triggers – e.g. the set of all prepositions, or the set of determiners – but open-class words such as nouns and verbs do not fit this paradigm. To correct open-class words, the approach must be extended to identify the relevant words, and to do so in text with additional errors of other types. Verb errors in English are also more complex linguistically than mistakes on other parts of speech due to the multiple functions verbs fulfill in a sentence. These additional complexities may explain why mistakes on open-class words have received so little attention, despite the fact that some of these errors are more frequent than the errors studied in previous work; the existing work (e.g., Lee and Seneff, 2008b; Gamon et al., 2009) focuses on special cases of open-class errors and does not address the challenges outlined above.

In this thesis, we focus on two groups of open-class words – nouns and verbs – and address several grammar and usage errors commonly associated with these parts of speech. General techniques to correct mistakes on open-class words are described and evaluated. To account for the complexity exhibited by verb errors, we propose a linguistically-motivated approach that makes use of the notion of *verb finiteness* to identify types of verb mistakes, before using a statistical machine learning approach to correct these errors.

The approaches proposed in this thesis led up to building state-of-the-art models in error correction that demonstrated superior performance in three error correction competitions. Specifically, the system developed for the HOO-2011 shared task placed first in that competition; the system that was built to participate in the HOO-2012 ranked first and second on different metrics; finally, the system developed for the CoNLL-2013 competition outperformed the other 16 teams by a large margin. Chapter 8 describes a complete error correction system that participated in the CoNLL-2013 shared task and implements many of the ideas proposed in this dissertation. We analyze the system and compare it against implementations by other teams in order to understand why this system is doing so much better. As a result, four design dimensions are identified that are crucial for building a robust error correction model and which are closely aligned with the methods proposed in this thesis.

Finally, the last question that we address is the interplay of various linguistic phenomena as they relate to error correction. Consider the example in Figure 1.1, where replacing the singular *phone* with the plural *phones* requires also changing the agreement marker on the verb. Standard models make decisions about each word independently, thereby ignoring these linguistic interactions altogether, and tend either to make inconsistent predictions or simply fail to identify such errors. We address such interactions in a model that *considers the phenomena jointly.*

1.3 Contributions

In spite of the growing interest in the field, research on automatic text correction is in the beginning stages. This dissertation identifies and addresses several fundamental questions that establish promising research directions in this area.

1.3.1 Choice of the Learning Algorithm

The first contribution of this thesis is an extensive, fair comparison of four state-of-the-art machine learning methods for error correction tasks. The results of this comparison clear up problems and contradictory conclusions from earlier evaluations that performed comparisons using incomparable data sets. Our results do not support earlier conclusions with respect to the performance of count-based models (Bergsma et al., 2009) and language models (Gamon, 2010), where no attention was paid to the size of the training set when comparing different learning algorithms. We demonstrate that the language modeling approach does not perform as well as other machine learning techniques. Furthermore, we show that discriminative classifiers are superior models for ESL correction tasks. In particular, it is shown that language models require five times more training data to achieve the performance of a discriminative classifier. The findings in this part pave the way for the next contribution and play a key role in later chapters where we build a complete error correction model.

1.3.2 Adaptation to Learner Errors

The second contribution of this thesis makes use of the well-known fact that learners make mistakes in a systematic manner and concerns the problem of *adapting* models to learner mistakes. We show that taking into account the specific types of errors that non-native speakers make can really help.

To expose models to error patterns, we introduce *adaptation* methods that allow us to build models that utilize knowledge about learner errors *minimal annotation costs*.

The adaptation idea is based on the key observation that estimating parameters relating to error regularities does not require a lot of annotation and can be done with a small annotation sample from the representative group of learners. Given these error statistics estimated from minimal amounts of annotated ESL data, the question is how to "inject" this knowledge into a learned model. We describe *two such methods* for two state-of-the-art machine learning algorithms that are identified to be superior for ESL correction tasks in the algorithm comparison experiments: one method applies to generative models, while the other approach applies to discriminative models.

1.3.3 Correcting Errors on Open-Class Words

Our contribution here is that we develop strategies for addressing the specific challenges of building models for correcting errors on open-class words. Specifically, methods for identifying nouns and verbs in noisy learner text are developed that take into account pre-processing errors. Furthermore, an end-to-end linguistically-motivated framework for the problem of correcting grammatical verb mistakes is proposed. This framework draws on the linguistic analysis of verbs: first, the type of verb error is determined using the notion of *verb finiteness*, and then the model proceeds to correct it. This is a departure from prior research that assumed that the type of verb error to be corrected is known in advance. Comparison of these techniques against other approaches, including those adopted by other systems in the CoNLL-2013 task, reveals that the proposed methods considerably outperform other techniques. This demonstrates that resolving these issues is key to building a successful error correction system for errors on open-class words.

1.3.4 Key Principles for Building a State-of-the-Art Error Correction System

This dissertation also contributes four design principles crucial for building a state-of-the-art error correction model. These principles are identified based on the analysis of the system developed for the CoNLL-2013 competition and closely correspond to the methods proposed in this work: choice of the learning algorithm; choice of training data (native or annotated learner data); model adaptation to the specific types of errors made by writers; and the use of linguistic knowledge.

1.3.5 Joint Learning and Inference

Finally, the proposed joint approaches that address errors occurring in interacting grammatical structures constitute an important contribution of this thesis. Specifically, we identify two pairs of interacting phenomena, and, crucially, show how to reliably recognize these structures in noisy ESL data, thereby facilitating the joint correction of these phenomena. Linguistic structures with interacting grammatical properties are then addressed via *joint inference* and *joint learning.* The joint inference approach is implemented on top of individually learned models using an integer linear programming formulation (Roth and Yih, 2004a). The joint learning model learns each pair of these phenomena jointly. Both of these approaches solve the challenges encountered by models that operate at the word level: the joint methods correct incoherent predictions that independently-trained classifiers tend to produce. Furthermore, because the joint learning model considers interacting phenomena during training, it is able to identify mistakes that require making multiple changes simultaneously and that standard approaches miss. Overall, the joint model significantly outperforms the system developed for the CoNLL-2013 shared task and that consists of a collection of independent classifiers. The joint models mark the first successful approach to jointly resolving grammatical errors.

1.3.6 Features and Resources

In addition to the primary contributions outlined above, as part of this work, *specific techniques and resources* for treating common ESL mistakes are developed. These constitute an important contribution of the thesis.

- We develop state-of-the-art features for article and preposition error correction (Chapter 5). The models that implement these features exhibited superior performance in several competitions (Rozovskaya et al., 2011, 2012, 2013).
- As we develop a linguistically-motivated approach to verb error correction based on verb finiteness, we generate annotation for verb finiteness for a subset of one of the learner corpora (Chapter 7).

1.4 Thesis Outline

This thesis is structured as follows. In Chapter 2, we present the machine learning background. Related work in text correction is presented in Chapter 3. Chapter 4 describes the learner corpora and the native English data sets that are used in this thesis, and the evaluation metrics. Next, in Chapter 5 we present experiments on the comparison of learning algorithms. In Chapter 6, we study closely the adaptation techniques for ESL error correction tasks. Chapter 7 describes methods for addressing errors on open-class words. Based on the work in these chapters, we present an end-to-end error correction model that focuses on five types of errors, which is described in Chapter 8. Chapter 9 focuses on developing models that address mistakes in interacting structures. Chapter 10 concludes with a discussion of directions for future work.

Chapter 2

Machine Learning Background

The approaches developed in this thesis rely on statistical and machine learning techniques. This chapter provides the necessary machine learning background and introduces the key machine learning concepts used throughout this thesis. Table 2.1 lists all the relevant machine learning notation.

2.1 Motivating Example

Machine learning studies algorithms for building systems that can learn from data. As an example, consider the task of *word sense disambiguation*: given an ambiguous word such as *bass*, predict its sense in the specific context. Examples below illustrate the two senses:

1. "The *bass* guitar is a stringed instrument played with the fingers or thumb."

2. "Fishing tips for striped *bass*."

Given the above two sentences, the goal is to build a model that learns to classify the occurrences of the word *bass* in text into two classes that correspond to the two senses of this word. Specifically,

Notation	**Description**
$\mathbf{x}$	Input example
y	Output (binary/multiclass)
$\mathbf{y} = (y^1, y^2, \ldots, y^M)$	Output (structure)
$\Phi(\mathbf{x})$	Feature generating function (binary/multiclass): takes input elements from $\mathcal{X}$ and maps them to d-dimensional feature vector. Often $\{0,1\}^d$
$\Phi(\mathbf{x},\mathbf{y})$	Feature generating function (structure): defined jointly over input and output
$\mathbf{w} \in \mathbb{R}^d$	Weight vector
$\mathcal{Y}(\mathbf{x}_i)$	The set of possible structures for input $\mathbf{x}_i$

Table 2.1: **Machine Learning Notation.**

the learned model should assign the “music” sense to the first occurrence of the word and the “fish” sense to the second one.

To develop such a system, one can make use of a large collection of texts (corpus) and use a machine learning algorithm to *learn* the typical contexts that correspond to each of the senses. To this end, each occurrence of the word *bass* in the corpus is represented as a vector of *features* that characterize the context of the target word. Based on these features, a machine learning algorithm learns a *function* h that distinguishes between the two senses. Then the learned model can be used to classify new unseen instances of the word *bass* by assigning them to one of the two classes –“fish” or “music”.

2.2 Learning Protocols

To build a machine learning system, such as the one for the word sense disambiguation example above, we assume an existence of a training corpus from which the system will learn function h. Learning protocol specifies the format of the training data. In this thesis, we use *supervised learning protocols*. In supervised learning, the training data is *labeled*, i.e. all occurrences of the word *bass* are specified for the correct sense. In contrast, in semi-supervised or unsupervised learning protocols, the training data may have no labels or may be only partially annotated.

Formally, given an *input space* $\mathcal{X}$ (sentences containing the word *bass*) and *output space* $\mathcal{Y}$ that consists of the label set (e.g., {fish, music}), the task is to find a function from $\mathcal{X}$ to $\mathcal{Y}$:

$$h : \mathcal{X} \rightarrow \mathcal{Y}.$$

In supervised learning, the input to the learner is the *training set* S_{train} $\{(x_1, y_1), \ldots, (x_n, y_n)\}$. We also assume a *feature generating function* Φ that maps each training example to a feature vector. An example of a feature generating function for the word sense disambiguation problem is a function that represents the context around the target word by defining a Boolean feature for each word x_w whose value is 1 if and only if x_w occurs in the sentence, and 0 otherwise. This maps a sentence containing the word *bass* to a boolean feature vector.

In the *training* stage, the learner produces function h that maps input $\mathbf{x} \in \mathcal{X}$ to output $y \in \mathcal{Y}$

based on the training samples.

In the *prediction* stage, given a test set S_{test}, $\{(x_{n+1}, y_{n+1}), \ldots, (x_{n+m}, y_{n+m})\}$ of unseen examples, the learned function is used to classify them, i.e. to assign an unobserved output value $y \in \mathcal{Y}$ based on an observed input vector $\mathbf{x}$.

2.3 Linear Models

Given the input and output space and a feature generating function, a machine learning algorithm is applied to learn function h from a set of all possible functions that can be used to represent the input. The set of functions from which h is selected is called a hypothesis space and denoted H. The choice of the function is important, as more expressive functions are able to separate the data into classes better but require more training data. Typically, the hypothesis space is restricted in some way. The most common approach in many NLP tasks is to restrict the hypothesis space to linear functions. Machine learning algorithms that assume that the input can be expressed by a linear function are called linear models and can be defined by a weight vector $\mathbf{w} \in \mathbb{R}^d$ that corresponds to the feature vector, and the *score* of instance $\mathbf{x}$ can be expressed as a weighted sum of the elements in the feature vector:

$$g(\mathbf{w}, \mathbf{x}) = \sum w_i \Phi_i(x) = \mathbf{w}^T \Phi(\mathbf{x}).$$

In the training stage, the goal is to learn a weight vector $\mathbf{w}$ using the training data. In the prediction stage, the learned weight vector is used to score all possible outputs and to select the highest-scoring output y. The actual prediction functions are presented in the following sections that define specific learning scenarios.

Linear models encompass a large family of machine learning algorithms, both discriminative and generative (see Section 2.5).

2.4 Binary and Multiclass Classification

Binary prediction is a classification task where the output space $\mathcal{Y}$ contains two classes. Binary classification can be formalized as a linear threshold function $\mathbf{w}^T \cdot \Phi(\mathbf{x}) > 0$. The prediction function of a binary classifier is defined as follows:

$$y^* = \mathrm{sgn}(\mathbf{w}^T \Phi(\mathbf{x})).$$

That is, the model will predict the example as positive if and only if $\mathbf{w}^T \Phi(\mathbf{x}) \geq 0$.

Most NLP problems, including the task of text correction, are cases that require classification into more than just two classes. Several strategies exist for extending the binary classification approach to multiclass cases. In this thesis, we make use of multiclass classification using the *one-versus-all* (OvA) decomposition, which makes the assumption that each class can be separated from the rest using a binary classifier. In this method, given m labels, m independent binary classifiers $h_y(\mathbf{x})$ are trained, corresponding to each of the m classes, where example $(\mathbf{x}, y)$ is considered positive for classifier y and negative for all other classifiers. Prediction is then performed by applying each binary classifier and choosing the label $\mathbf{y}$, for which the corresponding classifier produces the highest confidence score:

$$y^* = \arg\max_y h_y(\mathbf{x}).$$

2.5 Discriminative and Generative Classifiers

The choice of a machine learning model is important and is one of the central questions addressed in this thesis. Depending on how a decision on unseen examples is made, a distinction is drawn between discriminative and generative models. Discriminative algorithms model $P(\mathcal{Y}|\mathcal{X}) = P(\mathcal{Y}|\mathcal{X}, \mathbf{w})$. Generative or probabilistic algorithms, on the other hand, learn a model of the joint probability $P(\mathcal{X}, \mathcal{Y}) = P(\mathcal{Y}, \mathcal{X}|\mathbf{w})$, i.e. finding $\mathbf{w}$ that best matches $P(\mathcal{X}, \mathcal{Y})$ on the training data. Two of the most widely used discriminative algorithms are the Perceptron algorithm (Rosenblatt, 1958) and Support Vector Machine (SVM) (Boser et al., 1992). In the generative family, one of the most well-known algorithms is the Naïve Bayes learning algorithm. We study representatives of each family of these algorithms in more detail in Chapter 5.

2.6 Structure Prediction

In the previous section, we considered a classification problem where prediction for each example is made independently of other examples. In many NLP problems, however, predicting a label for a given instance depends on decisions for other instances. Consider, for example, the Part-of-Speech (POS) tagging task that is concerned with assigning POS tags to a sequence of words. Even though tags need to be assigned to individual words, it is clear that the tags assigned to words in the same sentence are not independent of one another. In structure prediction problems, predictions are made for structures, where the decisions made on the elements of the structure are interdependent. In the POS tagging example, a structure corresponds to a sequence of POS tags for a sentence.

Classifiers that we considered in the previous sections make predictions independently and are called *local* models, while models that learn the best sequence of labels for a structure are referred to as *global* predictors. We will use $\mathbf{y}$ to refer to the sequence of labels assigned to a structure; $\mathbf{y}$ consists of multiple decisions assigned to the elements of the structure and can be expressed as follows:

$$\mathbf{y} = (y^1, y^2, \dots y^M),$$

where y^i denotes the decision for the i^{th} element of the structure and M is the total number of elements in the structure.

2.6.1 Learning and Inference

In structure prediction, there are generally constraints imposed on the sequence of labels that comprise the structure, which restrict the set of all possible sequences. For instance, in the POS tagging example, the tag sequences should satisfy linguistic constraints imposed by the language. In English, constraints could be enforced that exclude sequences containing multiple consecutive determiners or clauses that do not have a predicate.

Finding the best assignment $\mathbf{y}^*$ to a sequence of output variables $\mathbf{y}$ is called *inference*. Several learning algorithms for modeling inference exist; these can be categorized into two frameworks. One approach models the dependencies among the output variables in the learning process. Examples include Hidden Markov Models (HMMs), Markov Random Fields (MRFs), and Conditional

Random Fields (CRFs, Lafferty et al., 2001).

The other solution is to decompose the model into two stages: *learning* and *inference*. In this framework, learning is performed by local models, and in the learning stage, output structure is completely ignored. The coherent assignment that satisfies the constraints on the label sequence is imposed at decision time based on the scoring functions produced by individual classifiers (Roth and Yih, 2004b, 2007). The main question in this framework is how to optimize the global solution given the output produced by individual classifiers. Formally, the inference objective can be stated as follows:

$$\mathbf{y}^* = \operatorname*{arg\,max}_{\mathbf{y} \in \mathcal{Y}(\mathbf{x}_i)} \mathbf{w}^T \Phi(\mathbf{x}_i, \mathbf{y}).$$

The set $\mathcal{Y}(\mathbf{x}_i)$ represents all possible structures that can be generated for example $\mathbf{x}_i$. Note that, unlike the feature generating function for local models $\Phi(\mathbf{x})$, in structure prediction problems, the feature function is defined jointly over the input and output space $\Phi(\mathbf{x}, \mathbf{y})$ and expresses dependencies between decisions for instances that belong to the same structures.

Since not all possible assignments to the structure output $\mathbf{y}$ are allowed, we need a method that allows us to solve the inference problem in the presence of constraints enforced on the output sequence. Roth and Yih (2004b) formalize the inference problem as a constrained optimization problem solved via the Integer Linear Programming (ILP) framework. This approach allows for combining the output produced by individually-trained classifiers at inference stage, thereby generating consistent predictions that satisfy structural and linguistic constraints. This approach provides a framework for an easy incorporation of linguistic or other structural knowledge into the model in an elegant way.

ILP has been widely used in many NLP tasks, such as semantic role labeling (Punyakanok and Roth, 2005); entity and relation recognition (Roth and Yih, 2004a); dependency parsing (Riedel and Clarke, 2006); and event extraction (Riedel and McCallum, 2011).

Note that while no polynomial-time algorithms exist for ILP and many approaches for inference have been used in the NLP literature, e.g. the cutting plane method (Sontag and Jaakkola, 2007), in this thesis the ILP instances are relatively simple and do not require complex solutions. The reason is that we decompose the problem and define ILP instances over linguistic structures (Chapter 9).

Chapter 3

Background on Text Correction

This chapter gives an overview of prior research on text correction and ESL error correction. The related work is clustered based on the problems that are identified and addressed in this dissertation. Section 3.1 reviews research research on text correction with a special focus on approaches to context-sensitive spelling correction, a problem that is very related to ESL correction and that has had a significant impact on the ESL error correction field. Section 3.2 presents related work in the area of correcting errors made by non-native writers of English, most of which was carried out on two error types – articles and prepositions. We justify the choice of the machine learning approach to ESL error correction adopted in this thesis, and, based on the prior work, identify two key issues in the standard application of this approach to ESL error correction tasks – the choice of the learning algorithm and the development of an appropriate training paradigm for error correction tasks. Section 3.3 reviews studies on other, less-researched errors involving open-class words, and uncovers the special challenges of addressing these errors. Section 3.4 studies the problem of correcting errors in the presence of interacting linguistic phenomena.

3.1 Text Correction

Both native and non-native speakers make a variety of errors that are not always easy to detect. Consider, for example, the problem of context-sensitive spelling correction (Golding and Roth, 1996; Carlson et al., 2001; Golding and Roth, 1999; Carlson and Fette, 2007; Banko and Brill, 2001). The *context-sensitive spelling correction task* involves correcting spelling mistakes that result in legitimate words, such as confusing *peace* and *piece.* Identifying *context-sensitive* mistakes is more difficult than identifying errors that result in non-words, which can be identified by simply checking whether the target word is a valid word in the dictionary. Detecting context-sensitive mistakes, on the other hand, requires taking into account the surrounding context. In this respect, this task is similar to the word sense disambiguation problem alluded to in Chapter 2.

To identify and correct context-sensitive errors, a system must be able to characterize the linguistic context in which the different words, such as *peace* and *piece*, tend to occur. These linguistic characteristics are highly variable, and the choice of a specific word in a given context may depend on the parts of speech of the neighboring words, the presence of lexical items in the vicinity of the target, and so on. It is thus difficult to develop knowledge-engineering methods to correct such errors.

Approaches to context-sensitive spelling correction can be classified into two main categories: semantic methods that primarily rely on human-made linguistic resources and methods that address the spelling problem via statistical and machine learning techniques.

An example of a semantic method is Hirst and Budanitsky (2005) that makes use of semantic distance measures in WordNet (Fellbaum, 1998) to detect words that are anomalous (or semantically distinct from nearby words) in context. This approach is based on the observation that the correct word is generally semantically related to the words occurring in its immediate context, but a context-sensitive mistake is not. Note that because the semantics-based techniques rely on characterizing words via semantic relatedness, these approaches cannot handle well mistakes that involve confusing words that do not carry a lot of semantic content or that are semantically similar (Hirst and Budanitsky, 2005).

The machine learning approaches to the context-sensitive spelling problem are considered in more detail, as they also represent the dominant paradigm in ESL correction tasks.

3.1.1 Machine Learning Approaches to Context-Sensitive Spelling Correction

The machine learning line of research investigating context-sensitive spelling correction views the problem as the task of word disambiguation or *word selection* (Roth, 1998). Existing works typically follow the supervised learning protocol and rely on pre-defined *confusion sets* for specifying the label space.

A *confusion set* or a *candidate set* (Golding and Roth, 1996) specifies the list of words that are frequently confused or used instead of one another in writing. Examples of confusion sets for context-sensitive spelling correction are {*sight, site, cite*} and {*piece, peace*}.

Given a *confusion set*, the context-sensitive spelling task can be treated as a word disambiguation problem over members of the confusion set. The problem is cast as a multiclass classification task, with a classifier trained on native English data that is assumed to be error-free. Each occurrence of a confusable word in text is represented as a *vector of features* derived from a context window around the target.

The machine learning paradigm offers two important advantages. First, unlike the semantics-based methods, it is not restricted to a specific type of confusions and can handle mistakes on both semantically-related and function words. Most of the ESL errors belong in one of these categories. Second, this supervised training setting allows for training models on a lot of data without the annotation effort that is usually needed to generate labels for training examples. The reason is that given a corpus of native English data from a reliable source that is assumed to have no spelling mistakes, each target word occurrence in text (e.g., *peace*) is treated as a positive training example for the corresponding word class, while negative examples can be inferred by substituting words in the confusion set for the correct word. Given text to correct, for each word in text that belongs to the confusion set, the classifier predicts the most likely candidate in the confusion set.

Different machine learning methods within this general paradigm have been applied to the context-sensitive spelling problem (e.g., Mays et al., 1991; Gale et al., 1993; Yarowsky, 1994; Golding, 1995; Golding and Schabes, 1996; Golding and Roth, 1999; Banko and Brill, 2001; Carlson et al., 2001; Liu and Curran, 2006; Carlson and Fette, 2007).

3.2 Approaches to Correcting ESL Errors

ESL error correction has recently become the focus of considerable interest in the NLP field, generating many publications and resulting in three recent shared tasks (Dale and Kilgarriff, 2011; Dale et al., 2012; Ng et al., 2013). A large proportion of this research has focused on correcting mistakes in article and preposition usage (Izumi et al., 2003; Han et al., 2006; Felice and Pulman, 2008; Gamon et al., 2008; Tetreault and Chodorow, 2008b; Gamon, 2010; Tetreault et al., 2010; Dahlmeier and Ng, 2011), although several selected studies also consider verb-related and noun-related errors (Lee and Seneff, 2008a; Gamon et al., 2008; Dahlmeier and Ng, 2012).

With the exception that learners and native writers exhibit errors of different types, most of the grammar and usage mistakes made by non-native speakers of English also fall into the category of context-sensitive errors, since they result in valid English words being confused. For instance, preposition and article confusions are some of the most common grammar and usage errors for ESL writers (Dalgish, 1985; Bitchener et al., 2005; Leacock et al., 2010). An additional challenge of ESL error correction is the noisy context that is manifest both in the presence of multiple mistakes, as errors tend to co-occur, and in the tendency of non-native writers to use constructions that may sound unnatural to native speakers, even when a native speaker finds the sentence to be acceptable.

Several ideas in this thesis are developed using article and preposition usage errors. We illustrate these below and, in parallel, introduce concepts and terminology related to error correction that are used throughout the thesis. The error correction terminology is summarized in Table 3.1. Chapter 4 provides more detail on all error types considered in this work.

1. "They listen to *$\varnothing$/*the* lecture carefully."
2. "Laziness is the engine of **the*/$\varnothing$ progress."
3. "He is an engineer with a passion **to*/*for* what he does."

In the above sentences, the *source word* (the word chosen by the ESL writer) and the *label* (the correct choice, as specified by a native English speaker) are emphasized. Incorrectly used words are marked with an asterisk, and $\varnothing$ denotes an error that involves an incorrectly omitted word (1) or a superfluous word (2). In (1), the definite article is incorrectly omitted. (2) is an example of an unnecessary article. In (3), the writer uses an incorrect preposition.

It is clear that the errors exemplified above do not lend themselves well to "knowledge"-engineering approaches. Indeed, prior research showed that rule-based systems cannot generalize well and are thus extremely limited in the types of mistakes that they can handle. Gamon et al. (2009) developed rules that address misuse of English irregular verbs. In the CoNLL-2013 shared task, selected teams enhanced their systems with rule-based modules but these dealt only with a small subsets of errors in the data, such as choosing an appropriate verb form in a restricted set of contexts or correcting number errors on nouns that belong to special categories (Xing et al., 2013).

Because it was obvious that rule-based approaches are not appropriate for ESL error correction, machine learning methods have become the dominant framework in the field.

Concept name	**Description**
Target word	A word in text that is selected as input to a classifier
Candidates	The words that are selected as input to a classifier
Confusion set (*candidate set*)	The set of words that are likely to be misused in place of one another, e.g., the set of prepositions
Source	The original word appearing in text, as chosen by the writer
Label	The correct word, as judged by the annotator
Context features	Features that characterize the context of the target word

Table 3.1: **Error Correction Terminology.**

3.2.1 Standard Machine Learning Approach to ESL Error Correction

The standard machine learning approach to correcting ESL mistakes follows the methodology of the *context-sensitive spelling correction task.* In the application of this training approach to ESL correction tasks, a model is tailored toward one specific mistake type, e.g. errors involving preposition usage, and is trained on well-formed native English text with features defined based on the surrounding context. *Confusion sets* are formed for each task. For instance, in preposition error correction, it is common to include the top n most frequent English prepositions. All occurrences of the top n prepositions in text are selected as input to the classifier. These input words are called *candidates.* In preposition systems, candidates can be identified in text based on a closed set of prepositions pre-defined in the confusion set. The *features* are generally based on the surface form, part-of-speech information and syntactic function of words in the immediate context around the potentially erroneous word. The classifier is then applied to non-native text to *select* the most appropriate candidate from the confusion set in context.

Based on this standard approach to ESL error correction that trains a machine learning model on native English data, we identify two key questions that are addressed in subsequent chapters:

- What is the right way to train a model for correction tasks (Chapter 6)?
- How does the choice of a machine-learning algorithm affect the performance of an error correction system (Chapter 5)?

We expand on these questions below.

3.2.2 Selection Training Paradigm for ESL Correction Tasks

In the approach described above, a machine learning algorithm is used to train a model on native English data. The model is then applied to non-native text to *select* the most appropriate candidate from the confusion set (e.g., article or preposition). We call this approach to ESL error correction a *selection* paradigm, because the candidate is *selected* based solely on the context around the target word, while the author's word choice is not used.

A number of works in ESL error correction follow the *selection* training paradigm. Izumi et al. (2003) used a maximum entropy model to detect 13 types of errors, including preposition and article usage errors, in essays written by Japanese learners of English. Han et al. (2006) trained a maximum entropy model to correct article mistakes. Chodorow et al. (2007), Tetreault and Chodorow (2008b), and Felice and Pulman (2008), Tetreault et al. (2010) trained maximum entropy models, and Felice and Pulman (2007) trained a voted perceptron algorithm using various word-n-gram, part-of-speech and parse features to correct preposition errors. Gamon et al. (2008) trained a decision tree model and a language model to correct errors in article and preposition usage. Yi et al. (2008) proposed a web count-based system to correct determiner errors. Bergsma et al. (2009) applied frequency-based algorithms with web-scale n-grams as features to preposition selection and context-sensitive spelling correction.

While the above systems followed different machine learning frameworks and had slight differences in the exact choice of features used, all of them adopted the selection paradigm outlined above.

3.2.3 Selection and Correction Training Paradigms

The key reason that the selection approach has been popular in error correction is that it does not require annotated data for training. Native data (from a "trustworthy" source, and therefore presumed to be correct) is used for training the system; it is cheap and is available in large quantities.

It is clear, though, that there is a problem with this standard approach of training on native data, as it ignores a key feature of the error correction problem that distinguishes it from the typical word disambiguation problem: in error correction tasks, we also have the word used by the author.

Depending on whether the author's word choice is used as a feature in training, we distinguish

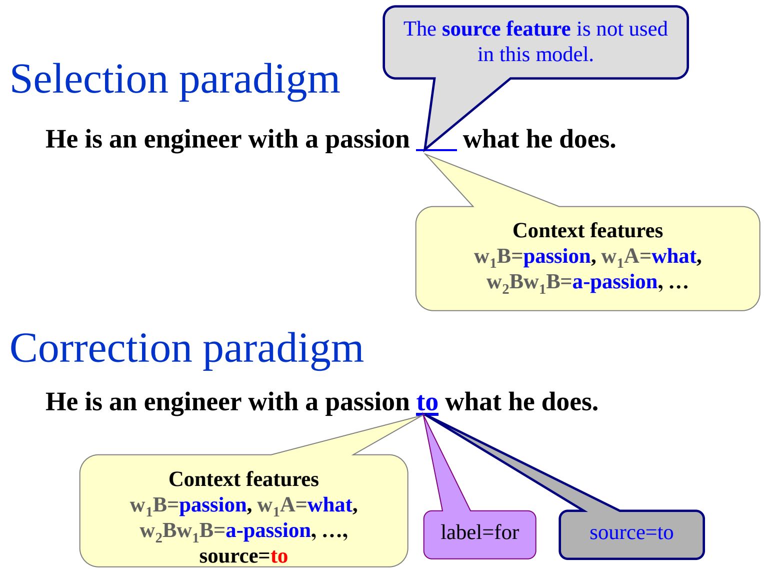


Figure 3.1: **Two training paradigms for error correction tasks.** In the *selection* training paradigm, models use only contextual information around the target word both in training and when making a prediction. In the *correction* paradigm, the author's word (the *source* word) is also used as a feature.

between two training paradigms for ESL correction tasks. In the standard *selection* paradigm, the decision of the classifier depends only on the context around the author's word, and the author's word itself is not taken into consideration in training. In the *correction* training paradigm, the source word is also used as a feature. The two training paradigms and the information available to the models in training and at prediction time in each case are illustrated in Figure 3.1.

The source word is an important piece of information. To begin with, even for ESL learners over 90% of their article and preposition usage is correct (Chapter 4); this is higher than the performance of the state-of-the-art classifiers for article and preposition word selection, (e.g., Han et al., 2006). Consequently, not using the writer's word, when making a prediction, may result in a system that adds more mistakes than there are in the data.

Furthermore, non-native English speakers make mistakes in a consistent manner, which is reflected in specific error patterns (see Chapters 4 and 6). These error patterns may be individual to a specific learner, first-language dependent, or prominent across multiple first languages.

One way in which error regularities are manifested is the *error rates* – i.e. the percentage of incorrectly used words – characteristic for ESL tasks. While generally error rates are low (see Table 5.7 in Section 6.4.2), more proficient learners will make fewer mistakes than beginners.

First-Language Influence

It is well known that the native language of the learner makes a significant impact on the use of English. To illustrate how error regularities may result from first-language influence, consider, for instance, a Chinese learner of English who might say "congratulations *to* this achievement" instead of "congratulations *on* this achievement", while a Russian speaker might say "congratulations *with* this achievement".

The effect of "language transfer" – that is applying knowledge from the native language, when learning a foreign language – has been studied for a long time in the second language acquisition literature. Language transfer can occur in grammar, pronunciation, vocabulary, discourse, and reading (Rod, 2008). Studies on cross-linguistic influence have clearly shown that the use of a foreign language is affected by the grammar of the first language, including phonology, the use of syntactic constructions and word order, morphosyntactic properties, lexical word choice, and so on (Odlin, 1989; Gass and Selinker, 1992; Ionin et al., 2008; Montrul and Slabakova, 2002; Montrul, 2000; Oh and Zubizaretta, 2003). The study of Ringbom (1978) provides evidence that the vocabulary of the native language affects the word choice in the second language. A number of studies consider the acquisition of collocations and how the collocations in the first language affect the acquisition and production of collocations in the second language (Biskup, 1992; Nesselhauf, 2003).

Article usage is another classic example of first-language influence. Article usage in English is known to be notoriously difficult even for very advanced ESL speakers (Ionin and Montrul, 2009), and the misuse of English articles by ESL learners is a topic that attracted a considerable amount of interest in second language acquisition research. Languages vary in whether they have articles (English, German, French do, but Chinese, Russian, and Korean do not). Moreover, languages that have articles differ also with respect to how articles are used. For instance, French and Spanish mark plural nouns that have generic references with the definite article, while English does not. Thus, the misuse of English articles by language learners reflects the use of articles in their native language and is manifested in how speakers of another language interpret and produce articles in

English. These findings as well as first-language-dependent regularities related to other mistake types are clearly supported by error statistics from learner corpora (Dalgish, 1985; Lee and Seneff, 2008a). For example, speakers of article-less languages such as Russian tend to make considerably more article mistakes in English than speakers whose first language has articles (see Chapter 4).

Regularities Across First-Language Backgrounds

In addition to first-language influence, some types of confusions are much more likely to occur than others across multiple first languages. For example, regardless of the first language, ESL writers are 70 times more likely to incorrectly use *in* in place of *on* than to use *from* (see Chapter 6).

Using the Source Word In Error Correction

Because ESL mistakes are regular, knowing the original word of the ESL writer provides information about which word was intended. Therefore, in addition to the surrounding context used in *selection* tasks, in error *correction* tasks, the choice of the right candidate should also depend on the original word chosen by the author.

The system can consider the word used by the writer at evaluation time, by proposing a correction only when the confidence of the classifier is high enough, but the source word cannot be used in training if the classifier is trained on error-free native English data. Ideally, we would like to train using annotated learner text. In that case, the original word of the writer can be used as a feature for the classifier, and the correct word, as judged by the annotator, will be viewed as the label.

Indeed, models trained on error-annotated data, even when trained on smaller amounts of data, perform significantly better than models trained on native English data, as they learn about the error patterns of the ESL writers (Gamon, 2010; Han et al., 2010; Dahlmeier and Ng, 2011). Han et al. (2010) trained a model on partially annotated Korean learner data. Gamon (2010) proposed a hybrid system for preposition and article correction, where two classifiers, trained on native and ESL data, are combined into a meta-classifier. Dahlmeier and Ng (2011) proposed an approach to correcting article and preposition mistakes using the Alternating Structure Optimization algorithm, which also combined learner and non-learner text for training. But note that while training on annotated ESL data offers the possibility of improved models, this approach requires large amounts of annotated ESL data (Gamon, 2010).

3.2.4 What is the Right Way to Train for Correction Tasks?

Training on native versus annotated learner data raises the question of the trade-off between the availability of expensive supervision in the form of annotated ESL data and the quality of the learned models. In Chapter 6, we address the question of what is the most appropriate way to train for correction tasks, given the limitations and advantages of each data source. In particular, Chapter 6 explores the advantages of training on native versus annotated learner data and proposes an approach to building models that *incorporate* the best of both modes: it will train on native texts to facilitate the possibility of training from large amounts of data without the need for annotation, but use the correction mode with a modest amount of annotated ESL data so that it can adapt to writers' errors.

However, before addressing the problem of the appropriate training paradigm for correction tasks, in Chapter 5 we look into algorithm comparison experiments based on the standard *selection* paradigm of training on native data: these experiments also provide insight into the error correction problem because we can learn which algorithms work better for the task.

3.3 Correcting Errors on Open-Class Words

Most of the effort in ESL error correction so far has been on article and preposition usage errors, as these are some of the most common mistakes among non-native English speakers of various first-language backgrounds. In addition, these errors lend themselves especially well to the machine learning approach outlined above: the confusion sets can be formed in a relatively simple way, and each of these phenomena can be addressed by building one classifier.

Mistakes on open-class words have attracted significantly less attention in the NLP literature, even though some of these errors are just as prevalent among non-native speakers of English. Notably, as shown in previous studies (Izumi et al., 2003; Lee and Seneff, 2008b) and Section 7.2, grammatical misuse of verbs is more common than determiner and preposition usage mistakes in several learner corpora. The acquisition of noun number in English is notoriously difficult for learners whose first language does not mark singular/plural on nouns, such as Chinese or Japanese (see Chapter 4). As an illustration of grammar errors on nouns and verbs, consider the sentences below:

1. "Some countries are having difficulties in managing a place to live for their **citizen/citizens* as they tend to get overpopulated."
2. "We **discusses/discuss* this every time."
3. "I will be lucky if I *{*will find*}/*find* something that fits."
4. "They wanted to visit many places without **spend/spending* a lot of money."
5. "They arrived early to **organized/organize* everything".

The first sentence shows an example of a mistake in noun number. The remaining examples illustrate various mistakes on verbs. Grammar errors on nouns and verbs are less understood linguistically and more difficult to treat. The reason is that correcting noun and verb errors requires a new paradigm that incorporates the specific components for correcting errors on open-class words. In particular, identifying the relevant words in text cannot be done using a closed list of words. Furthermore, because verbs fulfill multiple grammatical functions in a sentence, verb errors are sub-categorized into several groups, depending on the role of the verb in the sentence and the grammatical properties that correspond to its function.

For the above reasons, verb errors, while being more prevalent than article and preposition errors, are some of the least studied. With a few exceptions, there has been little work on verb errors. Moreover, because verb mistakes are so complex linguistically, the little earlier work done on verb errors only considered subsets of these errors and assumed the error sub-type is known in advance. Gamon et al. (2008) developed rules for detecting gerund/infinitive confusions. Lee and Seneff (2008b) proposed an approach based on pattern matching on parse trees combined with word n-gram counts for correcting agreement misuse and some types of verb form errors. However, tense mistakes are excluded in this study, even though tense errors constitute the most common category of verb errors for ESL learners and account for 40% of all grammatical verb errors (see Section 7.2.1). Tajiri et al. (2012) considered only tense mistakes. In all these studies, it was assumed that the type of mistake that needs to be corrected is known, and irrelevant verb errors were excluded (e.g., Tajiri et al. (2012) addressed only tense mistakes and excluded from the evaluation other kinds of verb errors). In other words, it was assumed that part of the task is solved. But note that, unlike in article and preposition error correction where the type of mistake is known based on the surface form of the word, in verb error correction, it is not obvious.

Work on noun errors is even more scarce; several published systems implement noun number correction modules either without describing it (Dahlmeier and Ng, 2012) or without carefully studying its components (Kao et al., 2013; Xiang et al., 2013). As a result, many challenges that are specific to correcting mistakes on open-class words have not been previously addressed.

In Chapter 7, we discuss the specific challenges of addressing mistakes on open-class words: noun number errors and most prominent mistakes on verbs. It is shown that addressing these challenges via a linguistically-informed machine learning approach is crucial for developing a state-of-the-art system that targets these errors.

3.4 Global Approaches to Error Correction

Most of the work in the area of ESL error correction has addressed the task by building statistical models that specialize in correcting a specific type of mistake. The predictions made by individual models are then applied independently (Rozovskaya et al., 2011) or pipelined (Dahlmeier and Ng, 2012).

The standard approach of training individual classifiers considers each word independently and thus assumes that there are no interactions between errors and between grammatical phenomena. But an ESL writer may make multiple mistakes in a single sentence, and these result in misleading local cues given to individual classifiers. Consider the example below:

- "In supermarket ***monitor/monitors *is/are** needed because we have to track thieves."

The agreement error on the verb *is* interacts with the noun number error: a correction system that takes into account the context may infer, because of the word *monitor*, that the verb number is correct. For this reason, a system that considers noun and agreement errors separately will fail to identify and correct the interacting errors. Furthermore, it may also produce inconsistent predictions.

Clearly, a successful error correction system needs to go beyond the approaches that train independent classifiers: it must consider the task at the sentence level. Approaches to error correction that go beyond individual linguistic phenomena can be broken down into those systems that are agnostic to the type of error they are trying to correct and those that attempt to combine the output of individual classifiers to find a global solution.

Park and Levy (2011) apply a noisy-channel model (Shannon, 1948) that is not confined to a specific error type but they find that the model has serious limitations, since in many instances the decisions are based on a bigram language model, even though language models are not as powerful as other learning methods for this task (Chapter 5). Gamon (2011) is another work that considers error correction beyond word-level phenomena by proposing a higher-order sequence model that is not restricted in the set of corrections. However, it is found that the method is not competitive with the traditional methods that target errors independently and restrict the set of potential corrections.

At the level of combining error-specific classifiers, Brockett et al. (2006) applied machine-translation techniques to correct noun number errors on mass nouns and article usage but their application was restricted to a small set of constructions. Dahlmeier and Ng (2012) proposed a decoder model that focuses on combining the output produced by individual classifiers for several types of mistakes by finding an optimal sequence in which the corrections are applied. The key idea is that by applying the models in order, one can solve the problem of noisy data and the presence of multiple corrections in context, since the subsequent classifiers will receive cleaner context. The problem, however, is that each of the models introduces more errors than it corrects. Incidentally, this problem was encountered by other systems that attempted to pipeline the output in the CoNLL-2013 shared task (Ng et al., 2013). Furthermore, because the decoder still corrected mistakes in a pipeline fashion, one at a time, it is unlikely that it could deal with cases that require simultaneous changes. Finally, in parallel to the work presented here, Wu and Ng (2013) attempted the ILP-based approach of Roth and Yih (2004b) in this domain but were not able to show any improvement because of the nature of the corpus they used and, most importantly, because the joint inference applied constraints in an indiscriminate manner.

In Chapter 9, we describe the first successful approach that addresses the problem of interacting linguistic phenomena by modeling these using joint inference and joint learning.

3.5 Summary

This chapter reviews prior work on text correction in general and related work on ESL error correction in particular. The standard approach to ESL error correction that adopts the machine learning paradigm developed for addressing context-sensitive spelling mistakes is presented. The key concepts and ideas that are important to have in mind reading the rest of the thesis are discussed: the distinction between the selection and correction paradigms based on whether the author's word is used in training and the motivation for this distinction; the lack of treatment in the literature of open-class words; the fact that errors interact, intuitively requiring a joint model as solution, but which raises problems of accurately modeling the relevant linguistic structure.

Chapter 4

Data Sets and Evaluation Metrics

This chapter introduces the data sets used to perform experiments in this dissertation and explains evaluation metrics commonly employed in error correction tasks. Section 4.1 introduces the types of learner errors addressed in this work. In Section 4.2, we describe the learner data sets. Next, we provide details on the annotation of one these data sets, the *Illinois* corpus. Section 4.4 presents the native corpora, and Section 4.5 describes the evaluation metrics. Section 4.6 summarizes the chapter.

4.1 Learner Errors

In this dissertation, we focus on errors made by learners of English as a Second Language and consider several grammar and usage mistakes that are most prominent among non-native English speakers: *article, preposition, noun number, and several types of verb errors.* These are illustrated in Table 4.1.[1] These errors reflect misuse of linguistic properties that are associated with the most frequent parts of speech, and represent some of the most common mistakes among non-native speakers.

There are some interesting properties that characterize some of these errors but not others. For instance, preposition mistakes are common across many first-language backgrounds, article mistakes are most often exhibited by learners whose first language does not have articles, while noun number errors are more strongly correlated with speakers of select first languages but less widespread among learners in general. Verb errors can be viewed as a special class of mistakes, since even though they are as common as article or preposition mistakes, they are less homogeneous

[1]We refer to these errors as *grammar and usage* mistakes, as not all of these mistakes are strictly grammar, even though some researchers denote all of these as grammatical mistakes. Lexical choice errors are not the focus of this work.

Error type	**Examples**
Article	“Laziness is the engine of **the*/∅ progress.”
Preposition	“He is an engineer with a passion **to*/*for* what he does.”
Noun number	“Some countries are having difficulties in managing a place to live for their **citizen*/*citizens* as they tend to get overpopulated.” “Science is surviving by overcoming the mistakes not by uttering the **truths*/*truth*.”
Verb (Agreement)	“We **discusses/discuss* this every time.”
Verb (Tense)	“Please check if the kids still keep throwing toys into the toy box in the backyard. I **remove*/*removed* some things but it **is*/{*will be*} full again very soon.”
Verb (Form)	“They wanted to visit many places without **spend*/*spending* a lot of money.” “It was not **surprised*/*surprising* to observe an increasing need for a convenient and cost effective platform.”

Table 4.1: **Errors addressed in this thesis.** Note that only the errors exemplifying the relevant phenomena are marked in the table; the sentences may contain other mistakes.

and include misuses that exemplify multiple grammatical properties, e.g. agreement, tense, voice, non-finite form. .

Because of these distinctions, the choice of the learner data set in the experiments is motivated in part by the type of error being addressed. For instance, to evaluate methods for noun number errors, it is essential to obtain data from a specific group of learners. Since verb errors are not homogeneous, one needs a larger learner corpus for developing a robust system.

4.2 Learner Data Sets

We use three corpora of annotated learner texts. All of them contain essays written by ESL learners, but they vary with respect to the annotation schema, proficiency level, and the first language of the learners. The methods proposed in this thesis are applied to multiple data sets and error types to demonstrate their robustness but experiments in the following chapters make use of different corpora, depending on the goal of the experiments.

4.2.1 The Illinois Data Set

The *Illinois* data set is a collection of essays written by ESL learners and annotated[2] by native English speakers (Rozovskaya and Roth, 2010a). The data set contains 63,000 words and includes data from speakers of nine first languages. The Illinois corpus is unique in containing data from speakers of multiple first-language backgrounds and, importantly, includes alternative corrections. Article and preposition data from the Illinois corpus is used in the algorithm comparison experiments in Chapter 5; the experiments use data from writers belonging to five language groups, as their writing contains a significant number of the relevant errors: Chinese, Czech, Italian, Russian, and Spanish. Section 4.3 provides details on the annotation of this corpus.

4.2.2 The FCE Data Set

The *FCE* data set (Yannakoudakis et al., 2011) contains 1244 student essays produced by learners of upper-intermediate proficiency level (1244 distinct learners) of 16 first-language backgrounds. The data set contains about 500,000 words of learner text and is corrected and error tagged using a detailed error classification schema of 88 error tags. Similar to the Illinois collection, FCE contains data by several first-language backgrounds and is annotated by native speakers but it is significantly larger. The data set, however, does not specify alternative corrections and is not evaluated with respect to inter-annotator agreement. We use the corpus in article and preposition experiments (Chapters 5 and 6) and in the verb experiments (Chapter 7).

4.2.3 The CoNLL Data Set

The techniques developed in this thesis resulted in superior error correction models developed for three text correction competitions, including the recent CoNLL-2013 shared task, where our system was ranked first among 17 participating teams. In Chapter 8, we describe and analyze this system along several dimensions that are aligned with the key contributions of this thesis. In addition, this corpus contains a good number of noun errors, and it is used in the experiments that address this error type (Chapter 7). Finally, in Chapter 9 we use the CoNLL data set to develop a model that applies joint learning and inference to account for linguistic interactions among errors.

[2]The annotation of the Illinois corpus is available at `http://cogcomp.cs.illinois.edu`.

Data set	**Experiments**	**Errors**	**Size (words)**
Illinois	Algorithms (Chp. 5)	Articles, preps.	63,000
FCE	Algorithms (Chp. 5); Adaptation (Chp. 6); Open-class errors (Chp. 7)	Articles, preps., verbs	500,000
CoNLL (Train) CoNLL(Test)	Open-class errors (Chp. 7); Illinois system at CoNLL (Chp. 8); Joint approaches (Chp. 9)	Articles, preps., nouns, verbs	1,200,000 29,000

Table 4.2: **Sizes of the ESL data sets.**

The CoNLL-2013 shared task released training and test data, and we denote these as *CoNLL train* and *CoNLL test*, respectively. The training data of the shared task is the NUCLE learner corpus (Dahlmeier et al., 2013). The corpus contains essays written by learners of English as a foreign language at the National University of Singapore and is corrected by English instructors. The test data for the task consists of an additional set of 50 student essays from the same first-language background. The training and the test data contain 1.2 million and 29,000 words, respectively.

4.2.4 Error Statistics and Error Rates

Table 4.2 lists the size of each data set (total number of words) and the experiments in which each of the data sets participates. Chapters 5 and 6 that address learning algorithm comparison and adaptation to learner errors, respectively, focus on article and preposition mistakes. The Illinois data set is used in Chapter 5, while the FCE corpus is used in Chapters 5 and 6. The FCE corpus is also used in verb error correction experiments (Chapter 7). The size of the Illinois data set is not large enough for the adaptation experiments and does not have a sufficient number of verb mistakes. In Chapter 8 we build a state-of-the-art error correction system by applying the methods proposed in the earlier chapters to the CoNLL corpus. The CoNLL data set is also an appropriate corpus for evaluating methods that address noun number errors (Chapter 7) and interacting mistakes (Chapter 9).

Table 4.3 shows the number of relevant errors in each corpus. Note that while the corpora contain other mistakes, we only show statistics for those errors that are used in the experiments. For the Illinois data, only statistics on preposition and article errors for speakers of five first-language groups are shown. The verb experiments on the FCE corpus cover three types of verb errors: agreement, non-finite verb form, and tense. The CoNLL data set excludes tense errors from evaluation but it includes other verb errors in the form category: verb form errors in the

Data set	Number of errors and error rates					
	Article	Prep.	Noun	Verb		
				Agr.	Tense	Form
Illinois	352 (8.4%)	418 (8.8%)	-	-	-	-
FCE	3,679 (4.8%)	2,609 (6.7%)	-	3,995 (4.8%)		
CoNLL (Train)	6,658 (2.4%)	2,404 (2.0%)	3,779 (1.6%)	1,527 (2.0%)	-	1,453 (0.8%)
CoNLL (Test)	690 (10.0%)	311 (10.7%)	396 (6.0%)	124 (5.2%)	-	122 (2.5%)

Table 4.3: **Statistics on errors in the ESL data sets.** Percentage denotes the error rates, i.e. the number of erroneous instances with respect to the total number of relevant instances in the data. For example, 10.7% of prepositions in the test data are used incorrectly and 10% of noun phrases in the CoNLL test data have determiner errors.

CoNLL corpus are defined differently and are not restricted to mistakes on non-finite verb forms (Chapter 8).

Table 4.3 also lists the *error rates*, i.e. the number of erroneous instances with respect to the total number of relevant instances in the data. For example, 10.7% of prepositions in the CoNLL test data are used incorrectly. Error rates differ by the corpus, e.g. article errors occur twice as often in the Illinois corpus than in the FCE corpus. Error rates also differ by error type: in the CoNLL test set, 10% of noun phrases (NPs) have article errors but only 6% of nouns have mistakes in noun number.

The CoNLL task made available two sets of annotations for the test partition: *original* and *revised*. Revised annotations include additional corrections missed in the initial stage. Since the revised version is based on the input from the participating teams and their systems, the revised corrections may be biased (Ng et al., 2013). In this thesis, we thus use the original version of the test data. Table 4.3 lists the number of errors in the *original* test set. The numbers in the revised version are slightly higher.

We observe that in the CoNLL train partition, the error rates for all of the phenomena are on average four to five times lower than the error rates in the CoNLL test, even though the linguistic profiles of the writers are similar. We noticed that quite a few errors in the training data set are missed in the annotation, and we believe that this is the key reason for the discrepancies in the error rates.

Finally, Table 4.4 shows the distributions of these errors across the three corpora and the coverage attained by considering these errors. Mistakes that involve misuse of determiners,[3] prepositions,

[3]Article errors constitute the majority of determiner errors. We refer the reader to Section 5.2.2 for more detail.

Data set	**Rel. freq. of error type (%)**				**Cumul.**
	Art.	**Prep.**	**Noun number**	**Verb**	**(%)**
Illinois	12.5	17.1	3.0	5.2	37.8
FCE	8.9	9.4	2.4	11.7	32.4
CoNLL (train)	14.8	5.3	8.4	6.6	35.1
CoNLL (test)	19.9	9.0	11.4	7.1	47.4

Table 4.4: **Relative frequencies of the errors that involve misuse of determiners, prepositions, noun number, and grammatical properties on verbs.** Percentage denotes the relative frequency of the error with respect to all errors in the data set. The last column shows the cumulative distribution.

and grammatical properties of verbs represent some of the most common types of errors in all of the corpora. Noun number errors are very common for the writers of the CoNLL corpus, but are considerably less prominent for speakers of many other languages as the statistics on the Illinois and the FCE corpora indicate. Overall, the errors that involve the linguistic phenomena that are considered in this dissertation account for 32.4% to 47.4% of all the errors depending on the learner corpus.

Two prominent error categories that are out of the scope of this thesis are word choice errors and spelling mistakes. While it is difficult to accurately estimate the frequency of word choice errors across different corpora due to the differences in the annotation schemas, it is possible to obtain an approximate estimate for the Illinois and FCE corpora. It is clear that word choice mistakes are very prominent for ESL writers and form the largest error category in both of these data sets: in the Illinois data set, word choice errors constitute 28% of all mistakes (Table 4.6), while in the FCE corpus, these account for about 20% of all errors. Spelling mistakes make up about 6.5% and 5.0% in the Illinois and FCE corpora, respectively. Estimating error frequencies for other error types is more difficult, as they depend on the annotation schema. Section 4.3 provides more detail on the error distributions in the Illinois corpus.

It is also interesting to compare the relative frequencies of error types across the corpora. Noun number errors in the CoNLL data set, for instance, are three to four times more common than in the FCE and Illinois corpora. While FCE and Illinois have writings produces by speakers of multiple first-language backgrounds, the CoNLL data set contains data from a more homogeneous group of learners. We believe that there is a higher proportion of Chinese speakers among the

CoNLL ESL speakers; and the Chinese language does not mark singular/plural on nouns. This strongly suggests that for the group of speakers in the CoNLL data set noun number mistakes are more common than for non-native writers in general. Although there are also differences for the other types of mistakes, they are less striking.[4]

4.3 Annotating Learner Mistakes

Learner corpora annotated for errors are essential for the development and evaluation of an error correction system. Such corpora are also very expensive to create, since annotation for mistakes is time-consuming, especially because errors are very sparse since over 90% of the instances are used correctly. Thus, acquiring annotation for a reasonable number of mistakes requires tagging at least 10 times more data. In this section, we describe how learner corpora are annotated and provide more detail on the annotation of the *Illinois corpus*.

The three learner corpora presented above follow different annotation guidelines and utilize different error taxonomies. The FCE corpus has 88 error tags; the NUCLE tagset size is 27; the Illinois annotation schema specifies nine error categories. Thus for some types of errors, the error tags do not always correspond to the same mistakes both because of different granularity levels and because of how errors are defined. For instance, the Illinois corpus annotates all grammatical verb errors with one tag, while the FCE corpus uses seven different tags; verb errors that involve wrong tense formation such as "has wrote" are tagged as form errors in the CoNLL corpus and as tense errors in the FCE corpus. With regard to annotation guidelines, the Illinois corpus specifies alternative and optional corrections, while the other corpora do not. The importance of providing alternative corrections is discussed in Section 4.3.1.

4.3.1 Annotation of the Illinois Data Set

The Illinois data set (Rozovskaya and Roth, 2010a)[5] contains essays written by learners of English of nine first-language backgrounds: Bulgarian, Chinese, Czech, French, German, Italian, Polish,

[4]Some differences may also be due to differences in annotation schema. Also, in the Illinois data set, the preposition errors have higher error rates, because a portion of sentences for annotation was selected to maximize the occurrence of preposition errors (Rozovskaya and Roth, 2010a).

[5]The material in this section is part of the Master's thesis by the author and is included here only for completeness, since this data is used in the dissertation.

Russian, and Spanish. With the exception of the Chinese sub-corpus, all of the data for annotation was extracted from the International Corpus of Learner English (ICLE, Granger et al., 2002), which is a collection of essays written by European speakers of advanced level of proficiency. The Chinese data is a part of the Chinese Learners of English corpus (CLEC, Gui and Yang, 2003), which contains essays written by students of all levels of proficiency. The annotators annotated about 2,600 sentences (63,000 words). The average annotation rate for the three annotators varied between 30 and 40 sentences per hour.

Annotation Procedure

The annotation of the Illinois corpus is performed at the sentence level: the annotators corrected all mistakes in the sentence and classified them using the specified error schema. The annotation was performed by three native speakers of North American English – one undergraduate and two graduate students – specializing in foreign languages and Linguistics, with previous experience in natural language annotation. A sentence was presented to the annotator in the context of the essay from which it was extracted. Essay context is necessary as some words may seem acceptable in the context of a sentence but are incorrect in the context of the essay.

Providing alternative and optional corrections

An important distinction of the Illinois data set from the other learner corpora presented in this chapter is that it specifies alternative and optional corrections: the annotators were encouraged to propose more than one correction and to mark corrections that are optional, i.e. where the original text was acceptable but a better alternative existed. Providing alternative corrections is essential for this task; judging grammaticality is very difficult even for native speakers, as the agreement on what constitutes a mistake even among native English speakers can be pretty low (Madnani et al., 2011): while some errors are clear cut, others may be regarded by some native speakers as acceptable usage. It is thus important for a robust evaluation of an error correction system to include alternative and optional corrections. Then the system is not penalized as long as its suggestion is one of the possible labels.

Schütze (1996) addresses the question of obtaining reliable linguistic judgments and in doing so identifies factors that influence the judgment process and that have to be controlled for when

Error type	**Description**	**Examples**
Article	Any error involving an article	"Laziness is the engine of *$the/\varnothing$* progress."
Preposition	Any error involving a preposition	"He is an engineer with a passion **to/for* what he does."
Noun number	Errors involving plural/singular confusion of a noun	"Science is surviving by overcoming the mistakes not by uttering the **truths/truth*."
Verb	Errors in verb tense and verb inflections	"We **discusses/discuss* this every time."
Word form	Correct lexeme, but wrong suffix	"It is not **simply/simple* to make professional army."
Spelling	Error in spelling	"...if someone **commited/committed* a crime..."
Word insertion, deletion, or replacement	Other corrections that do not fall into any of the above categories; mostly lexical choice errors	"There is a **probability/possibility* that today's fantasies will not be fantasies tomorrow."

Table 4.5: **Error classification used in the annotation of the Illinois data set.**

collecting linguistic data. It would be interesting to see whether his ideas can be applied to improve agreement on what constitutes acceptable usage in error annotation but this question is out of the scope of this dissertation.

Annotation Schema

The annotation schema of the Illinois corpus includes specifications for the following types of mistakes: *articles*, *prepositions*, *noun number*, *spelling*, *verb*, *word form*, *word replacement*, *word deletion*, and *word insertion*. The majority of errors in the last three categories constitute word choice mistakes. Table 4.5 provides a description of each error type.

Annotation Statistics

Table 4.6 shows statistics for the annotated sentences by language group and error type. Because the sub-corpora differ in size, we show the number of errors per hundred words. Category *punctuation* was not specified in the annotation, but can be easily identified and includes insertion, deletion, and replacement of punctuation marks. The largest error category is *word replacement*; it combines mistakes involving deleted and inserted words, and word substitutions. This is followed by the *punctuation* category, which comprises 22% of all corrections. About 12% of all errors involve articles, and preposition errors comprise 17% of all mistakes (a portion of sentences containing preposition errors was selected on purpose by looking for constructions likely to contain such errors, thus the relative frequency of preposition mistakes is slightly higher compared to the other

Source lang.	**Total sent.**	**Total words**	**Errors per 100 words**	**Corrections by Error Type (%)**								
				Art.	**Prep.**	**Verb**	**Word form**	**Noun num.**	**Word order**	**Spell.**	**Word repl.**	**Punc.**
Bulg.	244	6,197	11.9	10.3	12.1	3.5	3.1	3.0	2.0	5.0	46.7	14.2
Chin.	468	9,327	15.1	12.7	27.2	7.9	3.1	4.6	1.4	5.4	26.2	11.3
Czech	296	6,570	12.9	16.3	10.8	5.2	3.4	2.7	3.2	8.3	32.5	17.5
French	238	5,656	5.8	6.7	17.4	2.1	4.0	4.6	3.1	9.8	12.5	39.8
German	198	5,086	11.4	4.0	13.0	4.3	2.8	1.9	2.9	4.7	15.4	51.0
Ital.	243	6,843	10.6	5.9	16.6	6.4	1.4	3.0	2.4	4.6	20.5	39.3
Polish	198	4,642	10.1	15.1	16.3	4.0	1.3	1.3	2.3	2.1	12.3	45.2
Russian	464	10,844	13.0	19.2	17.8	3.7	2.5	2.5	2.1	5.0	28.3	18.8
Spanish	296	7,760	15.0	11.5	14.2	6.0	3.8	2.6	1.6	11.9	37.7	10.7
All	**2,645**	**62,925**	**12.2**	**12.5**	**17.1**	**5.2**	**2.9**	**3.0**	**2.2**	**6.5**	**28.2**	**22.5**

Table 4.6: **Error statistics on the annotated data by source language and error type**.

corpora). Two other common categories are *spelling* and *verb* errors.

Based on the error statistics in Table 4.6, we can make observations regarding differences in errors based on the first language. Consider the top two error categories, other than punctuation and word substitution errors: articles and prepositions. It can be observed from the table that there is a significantly smaller proportion of article errors for the speakers of languages that have articles, e.g. French and German, although speakers of those language also make article mistakes. These statistics support the the widely established view in the area of second language acquisition that article usage depends on whether the native tongue has articles and how articles are used in the native language of the learner (Ionin and Montrul, 2009). Preposition mistakes, on the other hand, are more evenly distributed across speakers of various first-language backgrounds. This behavior with respect to article and preposition mistakes has been similarly observed by other researchers (Dalgish, 1985).

Inter-Annotator Agreement

Correcting and annotating non-native text is challenging and requires a number of decisions on the part of the annotator. Human language allows for many ways to express the same idea. Furthermore, it is possible that the corrected sentence, even when it does not contain clear mistakes, still does not sound like a sentence produced by a native speaker. The latter is complicated by the fact that native speakers differ widely with respect to what constitutes acceptable usage (Madnani et al., 2011).

To date, a common approach to annotating non-native text has been to use one rater (Izumi

Agreement set	**Rater**	**Judged correct**	**Judged incorrect**
Agreement set 1	Rater #2	37	63
	Rater #3	59	41
Agreement set 2	Rater #1	79	21
	Rater #3	73	27
Agreement set 3	Rater #1	83	17
	Rater #2	47	53

Table 4.7: **Inter-annotator agreement by category (*correct*, *incorrect*) on the Illinois corpus**. The number next to the agreement set denotes the annotator who corrected the sentences on the first pass. *Judged correct* denotes the proportion of sentences in the agreement set that the second rater did not change. *Judged incorrect* denotes the proportion of sentences, in which the second rater made corrections.

et al., 2004; Han et al., 2006; Nagata et al., 2006; Gamon et al., 2008; Yannakoudakis et al., 2011).[6] The output of human annotation is viewed as the gold standard when evaluating an error correction system. The question of reliability of using one rater for preposition errors has been raised by Tetreault and Chodorow (2008a) who show that native speakers do not always agree on whether a specific preposition constitutes acceptable usage.

Since the annotation of the Illinois corpus is performed at the sentence level and includes multiple language phenomena, measuring agreement at the word level is problematic. The solution that we adopt here is to ask an annotator whether a sentence previously corrected by another rater is correct. After all, the goal of the annotation is to make the sentence grammatical, without enforcing that errors are corrected in the same way. To compute agreement, one hundred sentences annotated by each rater were selected and the corrections were applied. This corrected set was mixed with new sentences and given to the other two annotators. In this manner, each annotator received two hundred sentences corrected by the other two annotators.

For each pair of the annotators, agreement is computed based on the 100 sentences on which they did a second pass after the initial corrections by the third rater. To compute agreement at the sentence level, the annotated sentences are classified into the categories *correct* or *incorrect*. A sentence is considered *correct* if a rater did not make any corrections in it on the second pass.[7] Table 4.7 shows for each agreement set the number of sentences that were corrected on the second

[6]With the exception of the corpus used in the HOO-2011 shared task (Dale and Kilgarriff, 2011), but it is quite small.

[7]Punctuation errors are ignored.

Agreement set	**Agreement (%)**	**kappa**
Agreement set 1	56	0.16
Agreement set 2	78	0.40
Agreement set 3	60	0.23

Table 4.8: **Inter-annotator agreement at the sentence level (Illinois corpus)**. *Agreement* shows how many sentences in each agreement set were assigned to the same category (*correct*, *incorrect*) for each of the two raters.

pass. On average, 40.8% of the agreement set sentences belong to the *incorrect* category, but the proportion of *incorrect* sentences varies across annotators.

We also compute agreement on the two categories, *correct* and *incorrect*. The inter-annotator agreement and the kappa values are shown in Table 4.8. Agreement on the sentences corrected on the second pass varies between 56% to 78% with kappa values ranging from 0.16 to 0.40. The low numbers reflect the difficulty of the task and the variability of the native speakers' judgments about acceptable usage. In fact, since the annotation requires looking at several phenomena, we can expect a lower agreement when compared to agreement rate on one language phenomenon. Suppose rater A disagrees with rater B on a given phenomenon with probability 1/4, then, when there are two phenomena, the probability that he will disagree with at least one of them is $1 - 9/16 = 7/16$. And the probability goes down with the number of phenomena.

Inter-Annotator Agreement on Article and Preposition Mistakes

It should be stressed that the inter-annotator agreement numbers above are computed at the sentence level: two annotators are considered to disagree, if the second rater makes at least one correction in the sentence on the second pass. While these numbers are low, what they actually reveal is that native speakers disagree widely on what constitutes acceptable usage. These numbers, however, do not set an upper bound on the performance of an automated system, where the performance is typically evaluated with respect to a specific error type.

To determine agreement at the level of individual phenomena, we also compute inter-annotator agreement on preposition and determiner errors – the two common errors that we look at in this corpus – and find that it is quite high: between 91% to 96% depending on the raters.

Data set	Experiments	Size (words)
WikiNYT	Algorithms (Chp. 5); Adaptation (Chp. 6)	300 million
Google	Algorithms (Chp. 5); Adaptation (Chp. 6); Open-class errors (Chp. 7); Illinois system at CoNLL (Chp. 8); Joint approaches (Chp. 9)	1 trillion

Table 4.9: **Statistics on the native English corpora.**

4.4 Corpora of Native English Data

In addition to the learner corpora, some of the experiments in this thesis also make use of corpora of native English data. We briefly introduce them here. The first corpus, *WikiNYT*, is a selection of texts from English Wikipedia and the New York Times section of the Gigaword corpus and contains a total of 300 million words.

To experiment with larger data sets, we use the Google Web1T 5-gram corpus (Brants and Franz, 2006), which consists of frequency counts for bigrams, trigrams, 4-grams, and 5-grams, extracted from one trillion words of English Web text. We refer to this corpus as *Google*. The Google corpus is more than 3,000 times larger than the WikiNYT data set. Table 5.6 shows the sizes of the two native data sets and the experiments in which they are used. It should be stressed that the sizes of the native corpora are orders of magnitude larger than the sizes of the annotated learner data sets. Notably, the WikiNYT and the Google data sets are 250 and 800,000 times larger, respectively, than the CoNLL corpus, the largest among the learner data sets. This advantage of native English corpora is explored in Chapters 6 and 8.

4.5 Evaluation Metrics

This section presents the evaluation metrics employed in error correction tasks.

4.5.1 F1 Score

The standard metric of Precision, Recall, and F1 score has been widely used for ESL error correction tasks since it has an intuitive appeal. This measure was adopted in the three shared tasks on error correction (Dale and Kilgarriff, 2011; Dale et al., 2012; Ng et al., 2013) and was also used by other researchers (e.g., Gamon, 2010; Tajiri et al., 2012). Below is the definition of *Precision* and *Recall*, as they are used in the error correction domain.

Definition 4.5.1. *Let* **Error** *denote the total number of errors in the gold test data. Let* **Predicted** *denote the number of instances that the system flagged as errors. Then* ***Precision*** *and* ***Recall*** *of the system on the gold data are defined as follows:*

$$Precision = \frac{Predicted \bigcap Error}{Predicted}$$

$$Recall = \frac{Predicted \bigcap Error}{Error}$$

Precision indicates the fraction of predicted mistakes that are true errors in gold annotation. *Recall* computes the fraction of gold errors that the system identifies. Note that we can also distinguish between precision and recall on *detection* and *correction.* In detection, an error needs to be identified by the system, while in correction, the suggestion proposed by the system in place of the original word also needs to match the gold correction.[8] Thus, evaluation for correction is stricter than evaluation for detection. In this thesis, we evaluate the systems with respect to correction, unless otherwise noted.

Definition 4.5.2. *Given precision and recall,* ***F1 score*** *can be computed, which is the harmonic mean of precision and recall and is defined as follows:*

$$F1 = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall}$$

The trade-off between precision and recall can be calibrated via a decision threshold imposed on the confidence of the classifier (Carlson et al., 2001). Given a threshold value, the correction proposed by the system will be accepted only when the score that the model assigns to the correction exceeds the threshold. A higher precision and a lower recall are obtained when the decision threshold is high, and vice versa. Thresholding allows for the system to be adapted to the needs of the user. By varying the threshold value, one can also obtain an optimal point where F1 is maximized. Since the F1 score is the evaluation metric used in the CoNLL-2013 competition, the Illinois system at

[8]These definitions of detection and correction differ slightly from the definitions in Dale and Kilgarriff (2011) who also distinguish between error *detection* and error *identification* depending on whether the boundaries of the mistake are correctly identified. We ignore these distinctions, as they are not directly relevant to the problem setting in this work.

CoNLL, presented in Chapter 8, is optimized for this metric.

Precision/Recall Curves

By varying the decision threshold, it is possible to measure precision at multiple recall points and generate a graph that describes the behavior of the system in more detail. We believe that Precision/Recall curves provide a better picture on the performance than the F1 measure, which just reports performance at one point. Thus, for many experiments, Precision/Recall curves are shown.

4.5.2 AAUC

Average Area Under Curve (AAUC, Hanley and McNeil, 1983) is a measure commonly used to generate a summary statistic, computed as an average precision value over a range of n recall points and is another method of measuring system performance along the threshold continuum:

$$AAUC = \frac{1}{n} \cdot \sum_{i=1}^{n} Precision(i)$$

The advantage of the AAUC metric is that it provides an average precision over multiple recall points, as opposed to one point reflected in the F1 score. On the other hand, as a summary, it is more compact than Precision/Recall curves.

4.5.3 Accuracy: Measuring Success in Error Correction Tasks

The measures described above do not address one important characteristic that an error correction system should possess. As shown in Table 4.3, only a small percentage of words in each category have mistakes, while over 90% are used correctly. The low error rates are the key reason the error correction task is so difficult: it is quite challenging for a system to improve over a writer that already performs at the level of over 90%. Indeed, very few NLP tasks already have systems that perform at that level, even when the data is not as noisy as ESL data is.

To guarantee that the overall quality of the corrected text is not reduced, practical error correction systems should be tuned to minimize recall. Indeed, error sparsity makes it very challenging to identify mistakes accurately. In the CoNLL-2013 shared task, for example, no system attains a precision over 50%. Note that once precision drops below 50%, the system introduces more mistakes than it identifies.

Accuracy Clearly, optimizing for the F1 measure does not ensure that the system improves the quality of the text. In Rozovskaya and Roth (2010c), we proposed a different evaluation metric based on the *accuracy* of the data before and after running the system. The accuracy of the data before running the system is the percentage of correct instances. Based on the error rates in Table 4.3, the error rate on article errors in the Illinois data set is 8.4%, thus the accuracy of the data, or the *baseline*, is 91.6%. Given precision, recall, and F1 values, we can show how performance translates to accuracy.

Precision/Recall and Accuracy Conversion Using the definitions of precision and recall and the task baseline, we can relate the resulting accuracy of the classifier to the precision and recall on an error correction task as follows: Let *Precision* and *Recall* denote the precision and recall, respectively, of the system on an error correction task, and *ErrorRateLearner* denote the fraction of errors in the data. Then the task baseline (i.e. the accuracy of the data before running the system) is:

$$Baseline = 1 - ErrorRateLearner$$

It can be shown that the error rate after running the classifier is:

$$ErrorRateSystem = \frac{ErrorRateLearner \cdot (Precision + Recall - 2 \cdot Recall \cdot Precision)}{Precision}$$

It follows that the *accuracy* of the system on the task is $1 - ErrorRateSystem$.

4.6 Summary

This chapter presents the errors addressed in the thesis, the learner corpora, and statistics on the types of errors contained in these data sets. We also discuss the annotation process and provide detail on the annotation of one of the learner corpora, the Illinois data set, which was corrected and annotated for errors within the framework of this project. Finally, we describe the native English corpora that are used in this thesis and present the evaluation metrics.

Chapter 5

Correcting Closed-Class Errors: Algorithmic Perspectives

This dissertation views the problem of error correction from a machine learning perspective, which, as argued in Chapter 3, is the most suitable approach for this task. In this chapter, we consider an important issue at the core of the machine learning approach, the choice of the learning algorithm for ESL correction tasks. Following the standard method of training a classifier on native English data, we compare several state-of-the-art machine learning algorithms applied to two types of closed-class errors – articles and prepositions. These experiments shed light on which algorithms work better for the task. In particular, we show that there are significant differences in the performance of these algorithms. Importantly, our results reverse conclusions from earlier evaluations, where comparisons were performed between incomparable data sets.

5.1 Introduction

A variety of learning algorithms have been applied to correct ESL mistakes. Although the choice of a particular learning algorithm differs, with the exception of decision trees (Gamon et al., 2008), all algorithms used in ESL error correction are linear learning algorithms: some discriminative (e.g., Han et al., 2006; Felice and Pulman, 2008; Tetreault and Chodorow, 2008b; Tetreault et al., 2010), some probabilistic (Gamon et al., 2008; Gamon, 2010), or "count-based" (Bergsma et al., 2009).

While model comparison has not been the goal of the earlier studies, it is quite common to compare systems, even when they are trained on different data sets and use different features. Several conclusions have been made when comparing systems developed for ESL correction tasks. A *language model* was found to outperform a *maximum entropy classifier* (Gamon, 2010). However, the language model was trained on the Gigaword corpus (Linguistic Data Consortium, 2003), a corpus several orders of magnitude larger than the corpus used to train the maximum entropy classifier. Similarly, web-based models built on Google Web 1T 5-gram Corpus (Bergsma et al., 2009) were found to achieve better results when compared to a maximum entropy model that uses a corpus $10,000$ times smaller (Chodorow et al., 2007).[1]

To investigate the effect of the choice of the learning algorithm, we select two ESL correction tasks that received a significant amount of attention in the error correction literature – article and preposition usage errors – and compare four popular learning methods applied to the problem of correcting these mistakes: two probabilistic approaches – *Naïve Bayes* and *language modeling*; a discriminative algorithm – *Averaged Perceptron*; and a count-based method – *SumLM* (Bergsma et al., 2009), which, as we show, is very similar to Naïve Bayes, but with a different free coefficient. We train our models on native data from several sources, varying training sizes and feature sets, and show that there are significant differences in the performance of these algorithms.

Contributions

The main contribution of this chapter is an extensive, fair comparison of four state-of-the-art learning methods for the task of error correction that clears up problems and contradictory results from earlier evaluations. Contrary to previous results (Bergsma et al., 2009; Gamon, 2010), we find that when trained on the same data with the same features, Averaged Perceptron achieves the best performance, followed by Naïve Bayes, then the language model, and, finally, the count-based approach. In fact, we show that Averaged Perceptron performs better than SumLM when SumLM is trained with a *10 times* larger corpus; better than a language model that uses *5 times* more data; and comparably to NB trained with a corpus *2 times* larger. The results that we obtain, which hold for different training sets, genres, and feature sets, contradict conclusions from earlier evaluations that were not made on comparable data sets.

The remainder of this chapter is structured as follows. Section 5.2 presents the task setting. Section 5.3 introduces all linear learning algorithms from a unified perspective. We then compare the four algorithms using word n-gram features on the Illinois data set (5.5.1) and using both word n-gram and more sophisticated features on the FCE data set (5.5.2). Section 5.6 concludes the chapter.

[1]These two models also use different features.

5.2 Task Setting and Features

As this chapter focuses on article and preposition usage errors, we consider these mistakes in more detail. Examples of these errors can also be found in Table 4.1 but we illustrate them here for convenience:

1. "They listen to $*\varnothing/the$ lecture carefully."
2. "Laziness is the engine of $*the/\varnothing$ progress."
3. "He is an engineer with a passion $*to/for$ what he does."

In (1), the definite article is incorrectly omitted. (2) is an example of an unnecessary article. In (3), the writer uses an incorrect preposition.

5.2.1 Pre-processing

Both the article and the preposition models take as input the corpus documents pre-processed with a part-of-speech tagger[2] and shallow parser[3] (Punyakanok and Roth, 2001). The error modules rely on the pre-processing information for feature generation and for candidate generation, e.g. to identify missing article contexts.

5.2.2 Confusion Sets and Candidates

Preposition Errors

We distinguish three types of preposition mistakes: choosing an incorrect preposition; using a superfluous preposition; and omitting a preposition. For learners of many first-language backgrounds, the majority of the preposition errors are replacements, i.e., where the author correctly recognized the need for a preposition, but chose the wrong one to use. Table 5.1 shows the distribution of the three types of preposition errors in the FCE corpus.

Based on the error distribution, the preposition model targets only preposition replacement mistakes, which is a strategy followed in previous studies (e.g., Gamon, 2010). It is common in preposition error correction to include the top n most frequent English prepositions. The confusion

[2]`http://cogcomp.cs.illinois.edu/page/software_view/POS`
[3]`http://cogcomp.cs.illinois.edu/page/software_view/Chunker`

Error type	Example
Replacement 57.9%	"I can see **at*/*on* the list a lot of interesting sports."
Omission 24.0%	"I will be waiting *∅/*for* your call."
Unnecessary 18.1%	"Despite **of*/∅ being tiring , it was rewarding."

Table 5.1: **Distribution of preposition errors in the FCE corpus.**

Error type	Example
Replacement 15.7%	"Can you send me **the*/*a* letter back writing what happened to you."
Omission 57.5%	"Nowadays *∅/*the* Internet makes us closer and closer."
Unnecessary 26.8%	"One of my hobbies is **the*/∅ photography."

Table 5.2: **Distribution of determiner errors in the FCE corpus.**

set for prepositions includes the top ten prepositions, which accounts for 82% of all preposition errors (Leacock et al., 2010): {*on, from, for, of, about, to, at, in, with, by*}.

Determiner Errors

Similar to preposition mistakes, there are three types of determiner error: omitting a determiner; choosing an incorrect determiner; and adding a spurious determiner. Table 5.2 shows the distribution of determiner errors in the FCE corpus. In contrast to preposition mistakes, most of the determiner errors are omissions and insertions (over 80% of all determiner errors). Although the majority of determiner errors involve article mistakes, about 14% of these errors also involve personal and possessive pronouns[4] (Rozovskaya et al., 2012).

The system that we develop in this thesis focuses on article errors, with the confusion set defined as follows: {*a*, *the*, ∅}; *an* and *a* are collapsed into one class and can be disambiguated later using simple rules.[5] Article candidates include all articles and contexts, where an article is likely to be omitted.

Because the majority of article errors are missing articles, it is essential to design techniques for targeting these errors. Specifically, detecting missing article errors requires the identification of contexts where an article might have been incorrectly omitted. On the one hand, we wish to include as many such contexts as possible, since otherwise we have no chance of correcting such an error. Therefore, one possibility is to include every space as a potential article insertion point. However, this method is likely to produce a lot of noise, which will negatively affect the performance of the

[4]e.g., "Pat apologized to me for not keeping *the**/*my* secrets."

[5]We thank Josh Gioja for the code that performs phonetic disambiguation of the indefinite article.

system. It is thus customary to filter the contexts using pre-processing tools (e.g., Han et al., 2006). In this work, missing article contexts include spaces at the beginning of a noun phrase as well as spaces that follow a preposition or a verb. Noun-phrase-initial contexts are identified with the help of the part-of-speech tagger and shallow parser. Spaces following a preposition or a verb are added as likely contexts for omitted articles even when such contexts are not identified by the shallow parser as noun-phrase-initial, to offset the poor performance of the pre-processing tools on ESL data (Nagata et al., 2011). The problem of the pre-processing errors is addressed in more detail in Chapters 7 and 8.

5.2.3 Features

We use two groups of features for each task: (1) word n-gram (surface) features only and (2) more sophisticated features that also include POS and syntactic information. The word n-gram features are defined in the same way for all of the models; the higher-level features are task-specific. In the first part of the experiments in Section 5.5.1, we only use word surface features, in order to have a fair comparison with language models. In Section 5.5.2, we experiment with more sophisticated features. These enhanced features were used successfully in the systems developed for three text correction shared tasks, where our models achieved state-of-the-art results (Rozovskaya et al., 2011, 2012, 2013).

5.2.3.1 Word N-gram Features

As an example for the process of feature extraction, consider the target preposition *to* in the following sentence: "He is an engineer with a passion *to* what he does." (target articles are handled in a similar way). Let *preposition context* denote the preposition and the window around it. "A passion *to* what he" is a context for window size 2. We define three feature sets, varying window size from 2 to 4 words on each side (see Table 5.3). We use p to refer to a candidate preposition from the confusion set. All feature sets consist of word n-grams of various lengths spanning p and are of the form $s^{-k}ps^{+m}$, where s^{-k} and s^{+m} denote k words before and m words after p; we show two 3-gram features for illustration:

Feature set	Preposition context	N-gram lengths
Window2	a passion [to] what he	2,3,4
Window3	with a passion [to] what he does	2,3,4
Window4	engineer with a passion [to] what he does .	2,3,4,5

Table 5.3: **Description of the three feature sets based on word n-grams**. All feature sets consist of word n-grams of various lengths spanning the preposition (article) and vary by n-gram length and window size.

Feature type	Feature group	Features
Word n-gram		wB, w_2B, w_3B, wA, w_2A, w_3A, $wBwA$, w_2BwB, wAw_2A, w_3Bw_2BwB, w_2BwBwA, $wBwAw_2A$, wAw_2Aw_3A, $w_4Bw_3Bw_2BwB$, w_3w_2BwBwA, $w_2BwBwAw_2A$, $wBwAw_2Aw_3A$, $wAw_2Aw_3w_4A$
POS		pB, p_2B, p_3B, pA, p_2A, p_3A, $pBpA$, p_2BpB, pAp_2A, $pBwB$, $pAwA$, p_2Bw_2B, p_2Aw_2A, p_2BpBpA, $pBpAp_2A$, pAp_2Ap_3A
Chunk	NP_1	*headWord*, *npWords*, *NC*, *adj&headWord*, *adjTag&headWord*, *adj&NC*, *adjTag&NC*, *npTags&headWord*, *npTags&NC*
	NP_2	*headWord&headPOS*, *headNumber*
	wordsAfterNP	*headWord&wordAfterNP*, *npWords&wordAfterNP*, *headWord&2wordsAfterNP*, *npWords&2wordsAfterNP*, *headWord&3wordsAfterNP*, *npWords&3wordsAfterNP*
	wordBeforeNP	$wB\&f_i \ \forall i \in NP_1$
	Verb	$verb$, $verb\&f_i \ \forall i \in NP_1$
	Preposition	$prep\&f_i \ \forall i \in NP_1$

Table 5.4: **Features used in the article error correction system.** wB and wA denote the word immediately before and after the target, respectively; and pB and pA denote the POS tag before and after the target. *headWord* denotes the head of the NP complement. NC stands for noun compound and is active if second to last word in the NP is tagged as a noun. *Verb* features are active if the NP is the direct object of a verb. *Preposition* features are active if the NP is immediately preceded by a preposition. *Adj* feature is active if the first word (or the second word preceded by an adverb) in the NP is an adjective. *NpWords* and *npTags* denote all words (POS tags) in the NP.

1. a passion p
2. passion p what

5.2.3.2 Enhanced Features

The enhanced feature sets include features that use POS and shallow parser information. The article features are listed in Table 5.4. Table 5.5 lists the preposition features.

5.3 Datasets

All of the models in the algorithm comparison experiments are trained on native English data and evaluated on learner data.

Feature Type	Description
Word n-ngram	wB, w_2B, w_3B, wA, w_2A, w_3A, $wBwA$, w_2BwB, wAw_2A, w_3Bw_2BwB, w_2BwBwA, $wBwAw_2A$, wAw_2Aw_3A, $w_4Bw_3Bw_2BwB$, w_3w_2BwBwA, $w_2BwBwAw_2A$, $wBwAw_2Aw_3A$, $wAw_2Aw_3w_4A$
Preposition complement	$compHead$, $wB\&compHead$, $w_2BwB\&compHead$

Table 5.5: **Features used in the preposition error correction system.** wB and wA denote the word immediately before and after the target, respectively; the other features are defined similarly. *compHead* denotes the head of the preposition complement. $wB\&compHead$, $w_2BwB\&compHead$ are feature conjunctions of *compHead* with wB and w_2BwB, respectively.

5.3.1 Training Data

Using the corpora of native English data presented in Section 4.4, we generate training data of several sizes. The *size* of the data set for a specific classifier denotes the total number of candidate words that are input to the classifier (i.e. articles or prepositions) selected as described above.

From the WikiNYT dataset, training corpora of several sizes are generated, with the largest comprising 10 million training examples. To experiment with larger data sets, we use the Google corpus. This corpus is used in the preposition models only. Table 5.6 shows the number of training examples in the two native data sets.

5.3.2 Test Data

The algorithm comparison experiments are evaluated on the Illinois and the FCE datasets presented in Chapter 4. Because the FCE corpus is also used in the next chapter, where part of it is used in training, we divide the FCE data into training and test according to the split used in the HOO-2012 shared task (Dale et al., 2012): in this split, 1,000 files of the FCE corpus are reserved as training data, and the remaining 244 documents – for testing. The performance on the FCE dataset is reported on the test part of the data both in this chapter and in Chapter 6. Table 5.7 shows the sizes of the two ESL datasets. The number of examples denotes the total number of prepositions and articles in each corpus. We also show the number of errors and the error rate (the percentage of mistakes with respect to the total number of examples).

Data set	Total examples	
	Articles	Prepositions
WikiNYT	3M	1M 5M 10M
Google	-	26,000M

Table 5.6: **Training data sizes for the native English corpora in the algorithm comparison experiments.** Each entry indicates the number of preposition/article instances.

Data set	Total examples		Errors/Error rate	
	Art.	Prep.	Art.	Prep.
Illinois	4,185	4,731	352 (8.41%)	418 (8.84%)
FCE (Train)	60,743	31,274	2,892 (4.76%)	2,113 (6.76%)
FCE (Test)	15,398	7,673	787 (5.11%)	496 (6.46%)

Table 5.7: **Statistics on articles and prepositions in the ESL data sets.** *Error rate* denotes the percentage of examples (articles or prepositions in the data) that are mistakes.

5.4 The Models

We study four linear learning models: the discriminative method *Averaged Perceptron* (AP); two probabilistic methods – a *language model* (LM) and *Naïve Bayes* (NB); and a "count-based" method, *SumLM* (Bergsma et al., 2009).

Each model produces a score for a candidate in the confusion set. Since all of the models are linear, the hypotheses generated by the algorithms differ only in the weights they assign to the features (Roth, 1998, 1999). Thus a score computed by each of the models for a preposition p in the context S can be expressed as follows:

$$g(S,p) = C(p) + \sum_{f \in F(S,p)} w_a(f), \tag{5.1}$$

where $F(S,p)$ is the set of features active in context S relative to preposition p, $w_a(f)$ is the weight algorithm a assigns to feature $f \in F(S,p)$, and $C(p)$ is a free coefficient. Predictions are made using the winner-take-all approach: $p^* = \arg\max_p g(S,p)$. The algorithms make use of the same feature set $F(C,p)$ and differ only by how the weights $w_a(f)$ and $C(p)$ are computed. Below we explain how the weights are determined in each method. Table 5.8 summarizes the four approaches.

Method	Free Coefficient	Feature weights
AP	bias parameter	mistake-driven
LM	$\lambda \cdot prior(p)$	$\sum_{v_l \circ v_r} \lambda_{v_r} \cdot log(P(u\|v_r))$
NB	$log(prior(p))$	$log(P(f\|p))$
SumLM	$\|F(S,p)\| \cdot log(C(p))$	$log(P(f\|p))$

Table 5.8: **Summary of the learning methods**. $C(p)$ denotes the number of times preposition p occurred in training. $F(S,p)$ is the set of features active in context S relative to preposition p. λ is a smoothing parameter, u is the rightmost token in f, $v_l \circ v_r$ denotes all concatenations of left token substring v_l and right token substring v_r of feature f without u.

5.4.1 Averaged Perceptron

Discriminative classifiers represent the most common learning paradigm in error correction. AP (Freund and Schapire, 1999) is a discriminative mistake-driven online learning algorithm, a slight variation of the standard Perceptron. Perceptron maintains a vector of feature weights w and processes one training example at a time, updating w if the current weight assignment makes a mistake on the training example. The $C(p)$ coefficient is the bias parameter (see Table 5.8).

We use the regularized version of AP in Learning Based Java[6] (LBJ, Rizzolo and Roth, 2007). While classical Perceptron comes with a generalization bound related to the margin of the data, Averaged Perceptron also comes with a PAC-like[7] generalization bound (Freund and Schapire, 1999). This linear learning algorithm is known, both theoretically and experimentally, to be among the best linear learning approaches and is competitive with SVM and Logistic Regression, while being more efficient in training. It also has been shown to produce state-of-the-art results on many natural language applications, (e.g., Punyakanok et al., 2008).

5.4.2 Language Modeling

Given a feature $f = s^{-k}ps^{+m}$, let u denote the rightmost token in f and $v_l \circ v_r$ denote all possible concatenations of left token substring v_l and right token substring v_r of feature f without u. The language model computes several probabilities of the form $P(u|v_r)$. If $f =$"with a passion p what", then $u =$"what", and $v_r \in \{$"with a passion p", "a passion p", "passion p", "p" $\}$. In practice, these probabilities are smoothed and replaced with their corresponding log values, and the total weight contribution of f to the scoring function of p is $\sum_{v_l \circ v_r} \lambda_{v_r} \cdot log(P(u|v_r))$.

[6]LBJ can be downloaded from `http:cogcomp.cs.illinois.edu/page/software_view/LBJ`.

[7]*PAC* stands for *probably approximately correct.* The framework was introduced by Valiant (1984).

In addition, this scoring function has a coefficient that only depends on p: $C(p) = \lambda \cdot prior(p)$ (see Table 5.8). The *prior* probability of a candidate p is:

$$prior(p) = \frac{C(p)}{\sum_{q \in ConfSet} C(q)}, \tag{5.2}$$

where $C(p)$ and $C(q)$ denote the number of times p and q occurred in the training data.

We implement a count-based LM with Jelinek-Mercer linear interpolation as a smoothing method[8] (Chen and Goodman, 1996), where each n-gram length, from 1 to n, is associated with an interpolation smoothing weight λ. Weights are optimized on a held-out set of ESL sentences.

Window2 and *Window3* feature sets correspond to 4-gram LMs and Win4 to 5-gram LMs. Language models are trained with SRILM (Stolcke, 2002).

5.4.3 Naïve Bayes

NB is another linear model and is often competitive with more sophisticated approaches. NB architecture is also particularly well-suited for adapting the model to learner errors (Section 6.2.3). Weights in NB are determined, similarly to LM, by the conditional feature counts and the *prior* probability of each candidate p (Eq. (5.2)). For each candidate p, NB computes the joint probability of p and the feature space F. Assuming that the features are conditionally independent given p, this results in:

$$\begin{aligned} g(S,p) &= log\{prior(p) \cdot \prod_{f \in F(S,p)} P(f|p)\} \\ &= log(prior(p)) + \\ &+ \sum_{f \in F(S,p)} log(P(f|p)) \end{aligned} \tag{5.3}$$

NB's weights and its free coefficient are also summarized in Table 5.8.

Training Naïve Bayes for Deletions and Insertions

There is one key issue that needs to be addressed and that concerns training NB on the Google

[8]Unlike other LM methods, this approach allows us to train LMs on very large data sets. Although we found that backoff LMs may perform slightly better, they still maintain the same hierarchy in the order of algorithm performance.

corpus for mistakes that involve word deletions and insertions. Since the Google corpus contains only n-gram counts, training NB using the counts presents a problem when errors that involve omissions or insertions are being addressed (and the majority of article mistakes are of this type). The reason is that it is difficult to accurately estimate the prior counts for $\varnothing$ article when we do not have complete text but only n-gram counts.[9] We solve this problem by treating the article and the word following it as one target. Then, in the case of deletions, the following word alone acts as the target. This approach solves the problem of estimating the count of the $\varnothing$ article. This method can also be applied to other problems with deleted or inserted words. In particular, in addition to article errors, we also use this method, when training NB for errors involving extraneous prepositions (Chapter 8).

5.4.4 SumLM

For candidate p, SumLM (Bergsma et al., 2009)[10] produces a score by summing over the logs of all feature counts:

$$\begin{aligned} g(S,p) &= \sum_{f \in F(S,p)} log(C(f)) \\ &= \sum_{f \in F(S,p)} log(P(f|p)C(p)) \\ &= |F(S,p)|C(p) + \sum_{f \in F(S,p)} log(P(f|p)) \end{aligned}$$

where $C(f)$ denotes the number of times n-gram feature f was observed with p in training. It should be clear from equation 5.3 that SumLM is very similar to NB, with a different free coefficient (Table 5.8).

[9]When we have access to complete sentences, we apply a shallow parser and estimate the prior of $\varnothing$ by counting the number of NPs with $\varnothing$ article.

[10]SumLM is one of several related methods proposed in this work; its accuracy on the preposition selection task on native English data nearly matches the best model, SuperLM (73.7% versus 75.4%), while being much simpler to implement.

5.5 Comparison of Algorithms

Using the two native corpora – WikiNYT and Google – we build models of several training sizes. Note that the Google corpus does not contain complete sentences, but only n-gram counts. Thus we can use this data set with two limitations. First, we can only make use of word n-gram features, since we cannot pre-process it with a POS tagger or a chunker. Second, when training on the Google corpus, the simplest approach given the type of its data is to train an LM or NB (if we train a discriminative model such as AP we have to restrict the context features to the window of two words around the target). The other three models can be evaluated with the n-gram counts available. For example, we compute NB scores by obtaining the count of each feature independently, e.g. the count for left context 5-gram "engineer with a passion *p*" and right context 5-gram "*p* what he does .", due to the conditional independence assumption that NB makes. On Google, we train NB, SumLM, and LM with three feature sets: *Window2*, *Window3*, and *Window4*.

5.5.1 Algorithm Comparison Results

Key Results The key results of the fair comparison of the four algorithms are shown in Figure 5.1 and summarized in Table 5.9. The table shows that AP trained on 5 million prepositions performs as well as NB trained with twice as much data; the performance of LM trained on 10 million examples is better than that of AP trained with 10 times less data (1 million), but not as good as that of AP trained with half as much data (5 million); AP outperforms SumLM when the latter uses 10 times more data. Figure 5.1 demonstrates the performance results reported in Table 5.9; it shows the behavior of different systems with respect to *precision* and *recall* on the error correction task. We generate the curves by varying the decision threshold on the confidence of the classifier (Carlson et al., 2001) (see Section 4.5 for a description of evaluation metrics).

Performance for Various Features, Data, and Training Sizes

We now show results for different window sizes, training data and training sizes. Figure 5.2 compares the models trained on *WikiNYT-10M* corpus for *Window4*. AP is the superior model, followed by NB, then LM, and finally SumLM.

Results for other training sizes and feature set configurations show similar behavior and are

Algorithm Comparison: Key results
$AP > NB > LM > SumLM$
$AP \sim 2 \cdot NB$
$5 \cdot AP > 10 \cdot LM > AP$
$AP > 10 \cdot SumLM$

Table 5.9: **Key results on the comparison of algorithms**. $2 \cdot NB$ refers to NB trained with twice as much data as AP; $10 \cdot LM$ refers to LM trained with 10 times more data as AP; $10 \cdot SumLM$ refers to $SumLM$ trained with 10 times more data as AP. These results are also shown in Figure 5.1. Results are on the Illinois data set.

Training corpus	**Training size**	**Feature set**	**Performance (F1)**			
			AP	**NB**	**LM**	**SumLM**
WikiNYT	5M	*Window3*	**20.16**	18.93	18.50	16.78
WikiNYT	10M	*Window4*	**23.00**	20.97	19.30	17.83
Google	26,000M	*Window4*	-	**28.60**	27.46	23.96
Google	26,000M	*Window2*	-	24.23	**24.93**	20.84
WikiNYT (Article)	3M	*Window3*	**34.87**	31.82	-	32.92

Table 5.10: **Performance comparison of the four algorithms on the Illinois data set for different training data, training sizes, and window sizes.** Each row shows results for training data of the same size. The last row shows performance on the *article correction* task. All other results are for prepositions.

reported in Table 5.10, which provides model comparison in terms of best F1 obtained by each model. F1 is optimized using a decision threshold. The table also shows results on the *article correction task*. We do not evaluate the LM approach on the article correction task here, but the LM performance on articles in Chapter 8 exhibits similar behavior with respect to the other learning models.

5.5.2 Experiments with Enriched Features

We showed in the previous section that AP significantly outperforms the other models, and that NB is the second best model. In order to include LM in the comparison and keep the feature set identical across the models, the feature sets consisted only of word n-grams. Other algorithms, though, can make use of richer features that are not restricted to word n-grams. In Section 5.2.3, we defined additional features separately for articles and prepositions that make use of POS and shallow parser information. We denote them here *Window4-enhanced*, as they include all word n-gram features in that window. The next experiment shows that models indeed gain from using

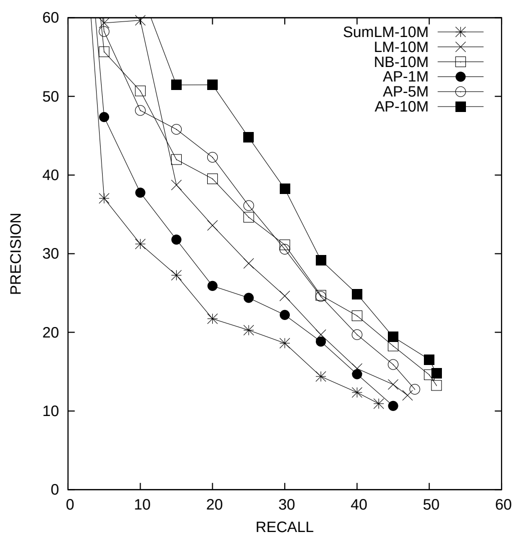


Figure 5.1: **Prepositions: algorithm comparison across different training sizes.** (*WikiNYT, Window3*). AP (1 million training examples) outperforms SumLM, when the latter is trained with 10 times more data; AP also outperforms LM when the latter is trained with 5 times more data. NB requires at least twice as much data to achieve the performance of AP. Results are on the Illinois data set.

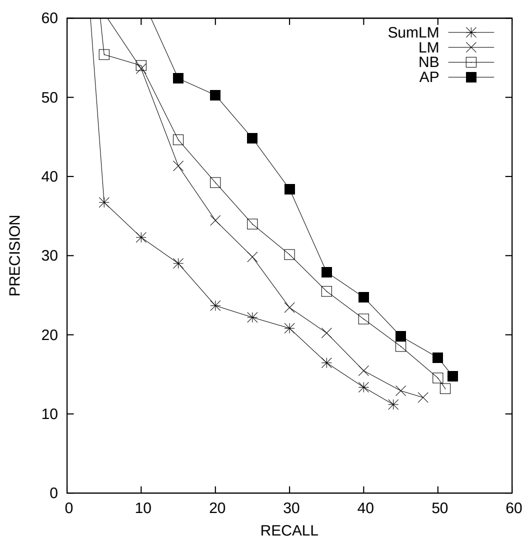


Figure 5.2: **Prepositions: algorithm comparison for training data of the same size.**: Performance of models for feature set *Window4* trained on 10 million examples from the *WikiNYT* data set. Results are on the Illinois data set.

Task	Training size	Feature set	Performance (F1)			
			AP	NB	LM	SumLM
Articles	3M	*Window4*	22.77	**24.75**	-	17.64
Prepositions	5M	*Window4*	**20.44**	18.24	15.08	15.83

Table 5.11: **Algorithm comparison on the FCE data set**. All models are trained on WikiNYT with word n-gram features.

Task	Training size	Feature set	Performance (F1)	
			AP	NB
Articles	3M	*Window4*	22.77	24.75
		Window4-enhanced	**30.76**	25.04
Prepositions	5M	*Window4*	20.44	18.24
		Window4-enhanced	**24.32**	20.45

Table 5.12: **Algorithm comparison for AP and NB on the FCE data set.** *Window4-enhanced* features use information beyond word n-grams. All models are trained on WikiNYT.

richer feature sets. Moreover, as we enhance the features with more knowledge, performance of the models improves and the gap between AP and NB grows.

Because these features are also used in the adaptation experiments in Chapter 6, we evaluate them on the FCE data set. Table 5.11 compares the four algorithms on the *FCE* test data for feature set *Window4*. Table 5.12 compares the performance of AP and NB models trained with *Window4* and *Window4-enhanced* features on the two tasks. These experiments confirm the earlier results about the relative performance of the four models; the only exception is that the NB model that uses word n-gram features has a better result on articles than the AP model. However, we find that with better features AP improves significantly and increases its performance gap compared to NB.

5.6 Summary

In this chapter, we conduct a fair comparison of four state-of-the-art machine learning approaches on article and preposition error correction tasks, using the selection training paradigm. We show that there are significant differences in the performance of these algorithms. In particular, discriminative classifiers outperform other machine learning methods on this task. Moreover, while a language modeling approach may seem like a natural choice when it comes to error correction, we find that, under the same conditions, language models are not as effective as other machine learning techniques, such as discriminative classifiers and Naïve Bayes. The results reverse conclusions from earlier evaluations that showed that language models outperform discriminative classifiers. These findings pave the way for the development of models that utilize knowledge about error regularities, presented in the next chapter.

Chapter 6

Adaptation to Learner Errors

We now move to the question of developing an appropriate training paradigm for error correction tasks. In particular, this chapter will focus on developing models that utilize knowledge about error regularities *with minimal annotation costs*. The proposed solution is based on the key observation that parameters relating to error regularities are simple. This observation motivates a two-stage approach that first extracts statistics on error patterns from learner data and then "injects" these into a model trained on native data. We describe a general method of obtaining error patterns from a small sample of annotated learner data. Next, building on the results of the experiments in the preceding chapter, we describe methods that allow us to incorporate these error statistics within the frameworks of the two top-performing learning algorithms, thereby providing these models with knowledge about error regularities.

6.1 Introduction

In the previous chapter, we compared four state-of-the-art learning methods trained in the selection training paradigm. An important advantage offered by the selection paradigm is that classifiers can be learned from large data sets with no need for annotation. The problem is that the selection training paradigm produces models that have no knowledge of the types of errors learners make since it does not use the source word in prediction. We argued in Section 3.2.3 that learner errors are systematic and that using the source word in prediction is better than just using the context.

A straightforward way to use the source word is to train on annotated learner data, but generally we have a very limited amounts of the latter. We thus wish to develop models that can be trained on native data but will utilize knowledge about the types of errors learners make. We address the following question:

- Suppose we have a small amount of annotated learner data. Is there a way to develop models trained on native English data that can utilize ESL data in prediction even when very little of it is available?

We describe an approach that does this by exploiting large amounts of native data, in addition to small amounts of ESL data. We call this approach *adaptation* to learner errors. The rest of this chapter describes the adaptation algorithms that we develop for this purpose.

Contributions

We propose an approach that "injects" knowledge about error regularities into the model trained on native data *without expensive data annotation* that would be required to train on annotated data alone:

- To obtain knowledge about error patterns, *error statistics* are collected over a sample of annotated ESL data. Importantly, the error statistics are very simple, and thus we only need a small annotated set to estimate them.

- To incorporate these error statistics into training, we propose adaptation methods for the two top-performing learning approaches identified in the previous chapter: the discriminative method AP and the NB algorithm. This allows us to train in the same way as when training on annotated data, using the *source* word as a feature, which, in turn, provides the models with knowledge about typical mistakes.

Section 6.2 presents the general method of extracting error patterns from a small annotated sample and describes the two adaptation algorithms. Section 6.3 presents results for the key experiments that show that the adapted models are superior to models trained in the selection paradigm. Section 6.4 presents analysis of the adaptation methods. Advantages of each of the adaptation algorithms are discussed in Section 6.5. Section 6.6 summarizes the chapter.

6.2 Adaptation with Small Amount of Annotation

Our goal is to utilize learner data to obtain a distribution of labels that reflect the types of errors learners make. Unlike when training on native data, where the distribution of the labels reflects

Label	**Sources**		
	a (6,681)	∅ (32,825)	the (15,055)
a (7,118)	**0.890**	0.094	0.016
∅ (32,239)	0.007	**0.972**	0.021
the (15,204)	0.008	0.054	**0.938**

Table 6.1: **Sample confusion matrix for article errors**. Based on the training data from the FCE corpus (Section 5.3). The left column shows the correct article. Each row shows the author's article choices for that label and $Prob(source|label)$. The numbers next to the targets show the count of the label (or source) in the data set.

the relative frequencies of the words, we would like it to be realistic; that is, we are interested in the bias of the labels as a function of the word chosen by the writer.

6.2.1 Obtaining Error Statistics from Annotated Data

Given a specific task, we collect all source/label pairs, where both the source and the label belong to the confusion set, and generate two confusion matrices: one where each cell represents $Prob$(source=s|label=l) (used in the artificial errors method in Section 6.2.2), and the other – inverse matrix – where each cell shows $Prob$(label=l|source=s), which is used by the priors adaptation method for NB in Section 6.2.3.

Tables 6.1 and 6.2 show sample confusion matrices of the first type for article and preposition mistakes based on the FCE data set (see Section 4.2.2). Consider, for example, the preposition matrix: it contains 10 rows and columns – one for each preposition – and each entry shows $Prob(p_i|p_j)$, the probability that the author chose preposition p_i given that the correct preposition is p_j. The matrix also shows the preposition count for each source and label in the data set.

Given the matrix and the counts, it is possible to generate the matrix in the other direction and obtain $Prob(p_j|p_i)$, the probability that the correct preposition is p_j given that the author's preposition is p_i.

The confusion matrices reveal that the distribution of alternatives for each source preposition is very different from that of the others. This strongly suggests that errors are systematic. Additionally, most articles and prepositions are used correctly, so the *error rate* is very low (the error rate can be estimated by looking at the matrix diagonals in the tables); for example, the error rate for the preposition *about* is lower than for *on*, since 94.7% of the occurrences of the label *about* are

Label	**Sources**									
	on (1,897)	about (1,952)	from (1,468)	for (3,995)	of (5,945)	to (3,897)	at (2,726)	in (6,367)	with (2,011)	by (625)
on (1,829)	**0.852**	0.003	0.005	0.020	0.014	0.008	0.012	0.076	0.008	0.002
about (1,946)	0.003	**0.947**	0.005	0.017	0.017	0.001	-	0.004	0.006	-
from (1,450)	0.002	0.002	**0.961**	0.005	0.010	0.001	0.003	0.010	0.002	0.004
for (4,034)	0.002	0.010	0.002	**0.939**	0.019	0.018	0.001	0.007	0.001	0.001
of (5,795)	0.002	0.004	0.002	0.004	**0.974**	0.001	0.001	0.009	0.002	0.001
to (3,990)	0.004	0.001	0.001	0.015	0.010	**0.936**	0.007	0.018	0.008	-
at (2,898)	0.011	0.001	0.002	0.002	0.007	0.009	**0.889**	0.076	0.002	0.001
in (6,274)	0.040	0.002	0.001	0.003	0.011	0.003	0.013	**0.924**	0.002	0.001
with (2,046)	0.002	0.010	0.003	0.008	0.009	0.009	0.002	0.015	**0.933**	0.009
by (621)	0.008	0.003	0.016	0.010	0.011	-	0.002	0.008	0.008	**0.934**

Table 6.2: **Sample confusion matrix for preposition errors**. Based on the training data from the FCE corpus. The left column shows the correct preposition. Each row shows the author's preposition choices for that label and $Prob(source|label)$. The numbers next to the targets show the count of the label (or source) in the data set.

correct, but only 85.2% of the label *on* are correct.

We now show how to incorporate these error statistics into models trained on native data. Two methods are described: the *artificial errors* method and the *priors* method. The first method is a general adaptation technique applicable to any discriminative model and implemented here within the AP algorithm; the second method is a special adaptation technique for NB. Sections 6.2.2 and 6.2.3 show how to modify the two learning algorithms to use these statistics.

6.2.2 Adaptation for a Discriminative Model

In this section, we propose an adaptation technique for the model that demonstrated the best performance in the algorithm comparison experiments in Chapter 5, the AP algorithm. In the selection paradigm, training examples are generated from native data by extracting all words that are members in the confusion set; this is an arbitrary distribution that merely depends on which sentences are chosen, and it dictates the distribution of examples presented to the training algorithm. Instead, here we show how to present examples in a way that reflects ESL error patterns. The method, which we call *artificial errors*, uses error statistics described above (Tables 6.1 and 6.2) to generate artificial errors in correct native English training data.

We proposed and evaluated the original version of the artificial errors approach in Rozovskaya and Roth (2010c). However, it suffers from the low recall problem, as discussed below. In this chapter, we describe a more general approach that improves over the original method by solving the low recall problem; the original method is a special case of the approach proposed here. First, we describe how artificial errors are generated in the original method.

6.2.2.1 Generating Artificial Errors with Error Statistics

Consider Tables 6.1 and 6.2. The top row of Table 6.2 shows the probabilities of a non-native writer incorrectly choosing each of the ten prepositions when the intended preposition should be *on*. For example, label *on* corresponds to source *on* in 85.2% of the cases and corresponds to source *in* in 7.6% of the cases. In other words, in 7.6% of the cases when the preposition *on* should have been used, the writer instead used *in*.

These probabilities are used to generate artificial errors in the training data. Thus, for each example in the training data (which are from native text and therefore assumed to be correct), with probability 7.6% we flip *on* in the training data to *in*. Then in the sense of the types of errors and the candidate distributions, the resulting text looks more like ESL text, because the native training examples now behave analogously to ESL data – the prepositions will now be incorrect with the same frequency and distribution as those from the corresponding target ESL user.

Note that when the artificial errors are generated, the labels of these examples are not changed, only the surface forms are replaced. We refer to the replacement as *source*, and it can be used as a feature, which we call the *source* feature. In other words, *source* denotes the form that the classifier sees as a feature (which could be an error) and *label* denotes the correct word. For ESL data, the *source* corresponds to the word chosen by the author, while for native data, the *source* corresponds to the artificially produced (and possibly erroneous) surface form.

Error Sparsity and Low Recall

The *artificial errors* approach described above generates errors with the frequency of naturally-occurring mistakes. This approach creates a low-recall model that tends to abstain rather than flag a possible error and thus does not identify many errors. This happens because of the low error rates in the learner data. The values of the *source* feature thus tend to have a very skewed label distribution, i.e. most of the time the source feature value corresponds to the label. The system will learn that in the majority of cases the source feature corresponds to the label, and will tend to over-predict it, which will result in very few mistakes being flagged. Note that the low recall problem is also observed when models are trained on annotated learner data (Section 6.4), for the same reason that is due to error sparsity.

The Error Inflation Method

So far, we have seen two extreme choices for training error correction models: (1) training without the *source word* feature (the selection paradigm) or (2) training with this feature, where the model relies on it too much and thus suffers from extremely low recall.

While maintaining strong precision is important, we wish to have the flexibility of increasing the recall. To this end, we reduce the confidence that the system has in the source word, while preserving the knowledge the model has about likely confusions (typical errors). This is achieved by decreasing the proportion of correct examples (where the *source* feature and the *label* are the same) and distributing the extra probability mass among the typical errors in an appropriate proportion by generating additional error examples. This inflates the proportion of artificial errors in the training data and, hence, the error rate, while keeping the probability distribution among likely corrections the same. Increasing the error rate improves the recall, while the typical error knowledge ensures that high precision is maintained. As we show in Section 6.5, this method causes the classifier to rely on the *source* feature less and increases the contribution of the features based on context. The learning algorithm therefore finds a more optimal balance between the *source* feature and the context features.

Algorithm 1 shows the pseudo-code for generating artificial errors in training data; it takes as input training examples, the confusion matrix CM, as shown in Tables 6.1 and 6.2, and the inflation constant C. The inflation constant specifies the proportion of correct instances that remain in training compared to the number of correct instances in data with naturally-occurring errors (e.g., 0.9 indicates that after artificial errors with the original error rate have been added, only 90% of the correct instances remain in the data, while 10% of those are converted to additional artificial mistakes). An inflation constant value of 1.0 corresponds to the *original* adaptation method that simulates learner mistakes without inflation, using the error rates of naturally-occurring mistakes. Table 6.3 shows the proportion of artificial errors created in training using the inflation method for different inflation values.

Inflation constant					
1.0 (Original)	0.9	0.8	0.7	0.6	0.5
7.20%	16.48%	25.76%	35.04%	44.32%	53.60%

Table 6.3: **Artificial errors: proportion of generated artificial preposition errors in training using the inflation method for various inflation values.**

Algorithm 1 Data Generation with Inflation

```
Input: Training examples $E$ with correct sources, confusion matrix $CM$, inflation constant $C$
Output: Training examples $E$ with artificial errors
for Example $e$ in E do
  Initialize $lab \leftarrow e.label$, $e.source \leftarrow e.label$
  Randomize $targets \in CM[lab]$
  Initialize $flag \leftarrow False$
  for target $t$ in $targets$ do
    if flag equals True then
      Break
    end if
    if t equals lab then
      $Prob(t) = CM[lab][t] \cdot C$
    else
      $Prob(t) = \frac{1.0 - CM[lab][lab] \cdot C}{1.0 - CM[lab][lab]} \cdot CM[lab][t]$
    end if
    $x \leftarrow Random[0, 1]$
    if $x < Prob(t)$ then
      $e.source \leftarrow t$
      $flag \leftarrow True$
    end if
  end for
end for
return E
```

6.2.3 Naïve Bayes Adaptation

We now describe another adaptation approach, also based on error statistics, for the second best performing model, NB. While NB does not perform as well as a discriminative classifier, the algorithm has advantages in several training scenarios. In particular, it is easy to train NB on the Google corpus (see Section 6.5.1). The adaptation for NB is also simpler to implement compared to the artificial errors method. Specifically, with the artificial errors approach, a separate classifier needs to be re-trained when adapting to a different group of learners. The NB adaptation presented here does not require re-training the model to adapt to different error statistics; it only requires changing one parameter for each classifier.

The method relies on the observation that error regularities can be viewed as a *distribution on priors over the correction candidates.* Consider the preposition correction task as an example. Given a preposition s in text, the *prior* for candidate p is the probability that p is the correct preposition for s. If a model is trained on native data without adaptation to learner mistakes, candidate priors correspond to the relative frequencies of the candidates in the native training data. More importantly, these priors remain the same regardless of the preposition used in text. From the model's perspective, it means that a correction candidate, for example *to*, is equally likely given that the author's preposition is *for* or *from.* This is clearly incorrect and disagrees with the error statistics, which show that not all confusions are equally likely.

We use annotated ESL data and define *adapted* candidate priors that are dependent on the author's preposition.[1] Let s be a preposition appearing in text, and p a correction candidate. Then the *adapted* prior of p given s is:

$$prior(p, s) = P(p, s|s) = \frac{C(s, p)}{C(s)},$$

where $C(s)$ denotes the number of times s appeared in the ESL data, and $C(s, p)$ denotes the number of times p was the correct preposition when s was used.

Using the information from Table 6.1, we compute adapted priors, i.e $Prob(label|source)$, for article mistakes. They are shown in Table 6.4. Similarly, adapted priors for preposition errors can be generated from Table 6.2. Table 6.5 shows sample adapted candidate priors for two author's choices – when an ESL writer used *on* and *by*.

A key distinction of the adapted priors is the high probability assigned to the author's preposition: the new prior for *on* given that it is also the preposition found in text is 0.817, versus the 0.07 prior based on native data. The adapted prior of preposition p, when p is used, is always high, because the majority of prepositions are used correctly. Furthermore, higher probabilities are also assigned to those candidates that are most often observed as corrections for the author's preposition. For example, the adapted prior for *in* when the writer chose *on* is 0.139, since *on* is frequently incorrectly chosen instead of *in.* Similarly, the most likely confusion for the source *by* is *with.* Note also how the distribution of adapted priors differs from that of the source-independent

[1]Note that the priors can also depend on the author's first language or the author's specific error patterns

Label	**Sources**		
	a	∅	the
a (0.130)	**0.948**	0.020	0.008
∅ (0.591)	0.033	**0.955**	0.045
the (0.279)	0.019	0.025	**0.947**

Table 6.4: **Inverse confusion matrix for article errors**. Based on the training data from the FCE corpus. The left column shows the correct article. Each row shows the author's article choices for that label and $Prob(label|source)$. The numbers next to the labels show the *relative frequency* of that label in the data set. Note this matrix can also be constructed from the one shown in Table 6.1.

priors based on native data: the most likely candidate, according to the native priors is *of*; it is four times more likely than *with*. However, now we generate a distinct family of priors for each source preposition. To summarize, source-independent priors correspond to the relative frequencies of the prepositions in native training data; adapted priors, on the other hand, reflect error rates and how the likelihood of each preposition confusion.

To determine a mechanism to inject the adapted priors into a model, we note that the NB algorithm makes use of two types of conditional probabilities: probability of a feature f given the label (correct preposition, or article) and the priors (Section 5.4.3). The first family of probabilities does not change and can be computed from native data. The second type is what we are changing here: rather than having a uniform source-independent prior that depends on the probability of observing a label, we now split it to multiple classifiers, and compute the prior as the probability for a label to appear when a specific source appears. We thus train NB in a traditional way, on native data, and then replace the prior component in Eq. (5.3) with the adapted preposition-dependent prior to get the score for p of the *NB-adapted* model:

$$g(S,p) = log\{prior(p,s) \cdot \prod_{f \in F(S,p)} P(f|p)\}.$$

We stress that in the new method there is no need to re-train the model to adapt to a different set of error statistics, as is the case with the artificial errors approach. Only one model is trained, and only at decision time do we change the prior probabilities of the model.

Candidate	**Source-indep.**	**Adapted prior**			
	prior	**author's choice**	**prior**	**author's choice**	**prior**
of	0.25		0.005		0.003
to	0.22		0.007		0.001
in	0.15		0.139		0.013
for	0.10		0.005		0.005
on	0.07	on	**0.817**	by	0.006
by	0.06		0.003		**0.928**
with	0.06		0.002		0.028
at	0.04		0.017		0.006
from	0.04		0.002		0.010
about	0.01		0.003		0.000

Table 6.5: **Examples of *adapted* candidate priors ($Prob(label|source)$) for two author's choices – *on* and *by* – based on preposition errors in the FCE training data**. *Source-independent prior* or native prior denotes the probability of the candidate in the standard model and is based on the relative frequency of the candidate in native training data. *Adapted priors* are dependent on the author's preposition. Adapted priors for the author's choice are very high. Other candidates are allocated more probability mass if they often appear as corrections for the author's choice.

6.3 Adaptation Experiments

The main goal of the experiments is to evaluate the adaptation methods proposed in Section 6.2. We described two adaptation methods that depend on the choice of the learning model (discriminative classifier or NB), both of which allow the models trained on native data to use the author's word as a feature.

The goal of the experiments is to evaluate the adapted models and compare them with models trained on native data in the standard selection paradigm. The main research question is the following:

- Is it possible to improve error correction models trained on native data by using a little bit of ESL annotation (6.3.2)?

6.3.1 Experimental Setup

We train three types of models in the experiments:

- *Native-trained models* – trained on native data in the selection paradigm; the weights for all features are estimated on native data.

- *ESL-trained models* – trained on ESL training data in the correction paradigm; the weights for all features are estimated on ESL data.
- *Adapted models* – trained on native data and adapted using a little bit of ESL data; the weights for context features are estimated on native data, while the source feature parameters are learned on learner data.

Native training data To train AP and NB models in the selection paradigm and with adaptation, we use the native corpora presented in Section 4.4. For articles and prepositions, models are trained on the WikiNYT corpus, and the training size is fixed to 3 million examples for articles and to 5 million for prepositions, with the exception of those experiments that compare models trained on native data of different sizes. For the preposition models, we also train NB on the Google corpus (26,000 million preposition examples).

ESL training data The FCE training partition (Table 5.7) in Section 5.3.2 is denoted as ESL training data. The ESL training data is also used to *adapt* native-trained models. In the key experiments, we adapt models trained on native data with 10% of the ESL training data; this portion contains about 6,000 and 3,000 article and preposition examples, respectively. The entire ESL training data contains 60,743 articles and 31,274 prepositions (Table 5.7).

Features As was shown in Section 5.5.2, enhancing features with POS tags and the shallow parser output improves the performance of the models. Thus, for the adaptation experiments, we use the enhanced feature sets for all models except for the models trained on Google; the Google models are trained on the word n-gram features *Window4*, i.e. word n-grams in the 4-word window around the target word.

Parameter tuning Part of the training data (10% of the files) is reserved for development. The results on the test partition of the FCE corpus use parameters optimized on the development data. The following parameters are optimized: the inflation constant for the AP-adapted models (0.8 for articles and 0.8 for prepositions), and the smoothing parameter for NB (-15 for the WikiNYT models and -21 for the Google models). The results on the development set are presented in Section 6.4.1.

Note on evaluation Results are reported using F1, which is optimized via a decision threshold (see Section 4.5). All models, with the exception of AP-adapted models, use a decision threshold.

Model	Training size		Infl.	Performance		
	Native	ESL	constant	P	R	F1
Native-trained (AP)	3M	-	-	28.57	33.33	30.76
Adapted (AP)	3M	6K	0.8	29.71	34.96	**32.11**
Native-trained (NB)	3M	-	-	20.17	33.03	25.04
Adapted (NB)	3M	6K	-	22.84	30.08	**25.96**

Table 6.6: **Articles: Key adaptation experiments.** All models are trained on WikiNYT data. Adaptation uses 100 files (10%) of the FCE training data for error statistics. Inflation constant is optimized on the development set. All differences are statistically significant (McNemar's test, $p < 0.0001$).

Model	Training size		Infl.	Performance		
	Native	ESL	constant	P	R	F1
Native-trained (AP)	5M	-	-	26.07	22.80	24.32
Adapted (AP)	5M	3K	0.8	31.27	22.27	**26.01**
Native-trained (NB)	5M	-	-	17.75	24.13	20.45
Adapted (NB)	5M	3K	-	21.22	27.47	**23.93**
Native-trained (NB, *Google*)	26,000M	-	-	28.47	27.87	28.16
Adapted (NB, *Google*)	26,000M	3K	-	31.42	30.00	**30.69**

Table 6.7: **Prepositions: Key adaptation experiments.** Models are trained on WikiNYT data or Google. Adaptation uses 100 files (10%) of the FCE training data for error statistics. Inflation constant is optimized on the development set. All differences are statistically significant (McNemar's test, $p < 0.0001$).

The AP-adapted models do not use a threshold, since the inflation parameter functions as a "knob" that controls recall.

6.3.2 Key Adaptation Results

Tables 6.6 and 6.7 compare the AP and NB selection models with AP-adapted and NB-adapted models for articles and prepositions, respectively, on the FCE test data. Adapted models always improve F1 compared to their selection counterparts.

A Two-Dimensional Summary of Adaptation Figure 6.1 displays a two-dimensional summary of the key adaptation results on the preposition correction task. The figure summarizes the adaptation idea of combining large amounts of native data with small amounts of ESL data. We can view adaptation to learner mistakes by considering a graph, with an x-axis representing the amount of ESL data and the y-axis – the amount of native training data used in training (see Figure 6.1). Each dot on the graph corresponds to a model, and the numbers in boxes show F1 results. All dots on the x-axis indicate models that are trained exclusively on ESL data (different amounts), and a dot on the y-axis represents a native-trained model in a selection paradigm. The dots in the middle are *adapted* models that are trained on a lot of native data and adapted using different amounts of ESL data. The x-axis is expensive, and the y-axis is cheap. The data sizes are shown for each axis in *log* scale, since the sizes of the native data are several orders or magnitude larger than the sizes of the ESL training data: the entire ESL data set contains 31,000 preposition examples, and the 10% shown on the graph corresponds to 3,000 prepositions. The native data size varies from 1 million to 13 million prepositions.

We can start by looking at models trained on ESL data: an ESL-trained model that uses 3,000 examples obtains an F1 score of 14.62. We can obtain an F1 of 26.73 if we use all of the ESL training data available, 10 times more. However, this is still not a lot of data, and in most cases, we will not have even this much annotated data available, so we really would like to train using native data. A native-trained model that uses 1 million examples obtains an F1 score of 22.01, and we can improve further, if we use 5 million and 13 million training examples. This demonstrates the importance of using native data for learning weights for context features.

The main lesson that is learned from the graph is that increasing the amount of training data is important, but one type of data is a lot more costly than that other. The adaptation methods are shown to gain a lot from a small increase in the costly dimension: the key result (marked as "1" on the graph and also shown in Table 6.7) is that a 5 million model *adapted* with 3,000 ESL prepositions outperforms the model trained in a *selection* paradigm (26.01 versus 24.32 F1 points), demonstrating that even a little bit of ESL data can make a big difference. This improvement is consistent for adapted models that use 1 million and 13 million prepositions from native data.

The second comparison (marked as "2") demonstrates the importance of using native data for learning the weights of context features: an ESL-trained model that uses 3,000 prepositions performs very poorly, since there is not enough evidence to learn the feature weights reliably. We can improve significantly to 23.92 and 29.48 by adding 1 million and 13 million native examples, respectively.

Finally, result marked as "3" shows that the model adapted with just 3,000 preposition instances is very close to the performance of the model adapted with 10 times more data. This shows that

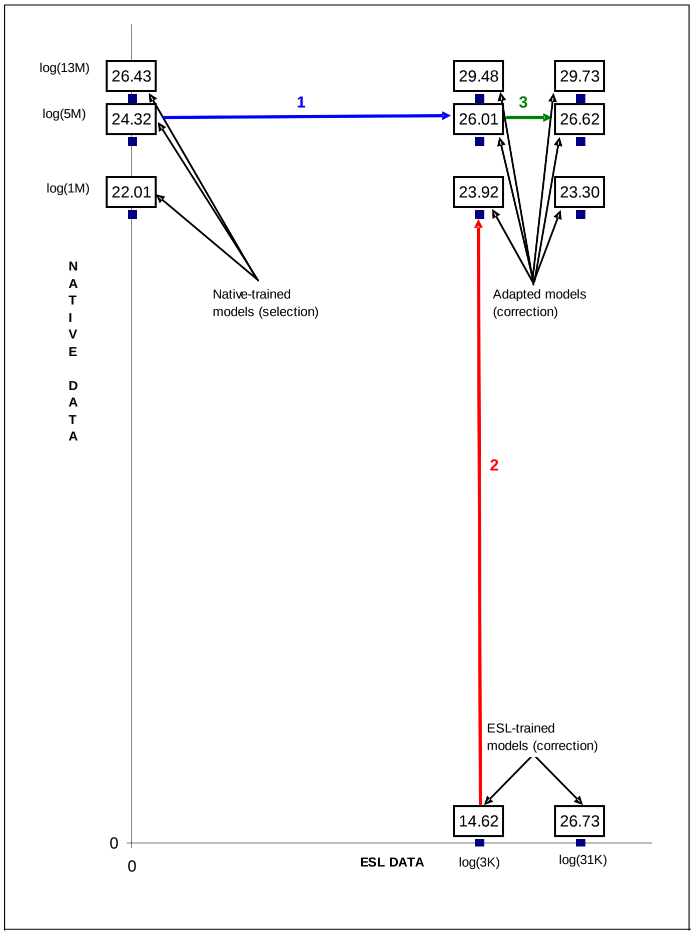


Figure 6.1: **Adaptation: a two-dimensional summary. Increasing the amount of training data is important, but one type of data is a lot more costly than that other. Adaptation methods are shown to gain a lot from a small increase in the costly dimension.**The x-axis shows the ESL data continuum as a percentage of the annotated training examples. The y-axis shows the native training data. The numbers in boxes show **F1 results** on the preposition correction task. Note how the native data size is in fact several orders of magnitude larger than the ESL data.

by using just a small amount of annotated ESL data, it is possible to estimate the parameters of the *source* feature reliably. Finally, we also note that the model trained on 5 million native examples and adapted with 3,000 examples is close in performance to an ESL-trained model that uses 10 times more data, while the model that uses more native data (13 million) outperforms the ESL-trained model by almost 3 F1 points (from (26.73 to 29.48).

6.4 Analysis of the Adaptation Approaches

Along with the main question addressed in the previous section, we are also interested to answer the following questions, as part of the analysis of the adaptation techniques:

1. *Size of annotation.* How much annotated data is needed for estimating error statistics robustly? Does the size depend on the error type and the size of the confusion set (Sec-

tion 6.4.2)?

2. The *error inflation method*. Can the error inflation method be used to improve recall for ESL-trained models (Section 6.4.3)?

3. How do adapted and ESL-trained models compare? In particular, if we have a reasonably large annotated learner corpus (like FCE), is it still better to train on native data with adaptation (Section 6.4.4)?

4. *Genre*. Is there a difference in the genres of the ESL and native data and what effect does the genre have on the learned models (Section 6.4.5)?

Before this analysis, we consider adaptation experiments on the development data that led to the key adaptation results in Section 6.3.2.

6.4.1 Adaptation Experiments on the Development Data

Tables 6.8 and 6.9 show the performance of AP and NB systems for articles and prepositions, respectively. For the AP-adapted models, results with several values for the inflation parameter are shown to examine how the parameter value affects the model behavior. Note that the AP-adapted models with higher parameter values have lower recall, since the error rate in the training data is lower. For example, models with value of 1.0 attain a recall of 10.47 and 8.74 for articles and prepositions, respectively. We also note that any value between 0.9 and 0.5 will give a boost in F1 compared to the rate of 1.0 that corresponds to the error rates of naturally-occurring mistakes.

Comparing the adaptation effect on AP and NB models, we find that AP-adapted models exhibit very similar behavior on both tasks, while NB-adapted models on the article task show only modest improvement. We believe that this is because the preposition problem has a larger confusion set, and thus the models benefit more from learning about the error patterns.

6.4.2 How Much Annotation is Required for Estimating Error Statistics

In this section, we investigate how much annotated ESL data is needed to estimate error statistics robustly and to improve performance over the native-trained models. We vary the ESL data size used for error estimation between 0.5% to 100% of the ESL training data available.

Model	Training size		Infl.	Performance		
	Native	**ESL**	**constant**	**P**	**R**	**F1**
Native-trained (AP)	3M	-	-	21.35	23.84	22.52
Adapted (AP)	3M	6K	1.0	39.56	10.47	16.55
			0.9	28.17	23.26	25.47
			0.8	23.50	30.81	**26.66**
			0.7	20.13	34.88	25.53
			0.6	19.22	40.12	25.98
			0.5	16.86	41.86	24.04
Native-trained (NB)	3M	-	-	14.17	25.87	18.31
Adapted (NB)	3M	6K	-	14.24	26.74	18.58

Table 6.8: **Articles: adaptation for native-trained models on the development data**. All models are trained on WikiNYT data. Adaptation uses 100 files (10%) of the FCE training data for error statistics.

Model	Training size		Infl.	Performance		
	Native	**ESL**	**constant**	**P**	**R**	**F1**
Native-trained (AP)	5M	-	-	21.38	22.01	21.69
Adapted (AP)	5M	3K	1.0	55.10	8.74	15.08
			0.9	31.72	14.89	20.26
			0.8	30.34	22.98	**26.15**
			0.7	24.42	23.95	24.18
			0.6	20.80	26.86	23.43
			0.5	18.50	29.45	22.72
Native-trained (NB)	5M	-	-	13.68	20.71	16.47
Adapted (NB)	5M	3K	-	20.88	22.98	**21.87**
Native-trained (NB, Google)	26,000M	-	-	20.89	25.89	23.12
Adapted (NB, Google)	26,000M	3K	-	27.46	26.21	**26.82**

Table 6.9: **Prepositions: adaptation for native-trained models on the development data**. Models are trained on WikiNYT or Google. Adaptation uses 100 files (10%) of the FCE training data for error statistics.

Model	**Training size**		**Performance**		
	Native	**ESL**	**P**	**R**	**F1**
Native-trained (AP)	3M	-	28.57	33.33	30.76
Adapted (AP)	3M	0.5%	27.98	33.43	30.46
		1%	28.77	33.33	30.88*
		5%	30.13	34.45	**32.14***
		10%	29.71	34.96	32.11*
		50%	30.09	35.57	32.60*
		100%	29.61	35.06	32.11*
Native-trained (NB)	3M	-	20.17	33.03	25.04
Adapted (NB)	3M	0.5%	27.55	29.37	**28.43***
		1%	27.22	29.88	28.48*
		5%	23.78	29.67	26.40*
		10%	22.84	30.08	25.96*
		50%	22.93	30.08	26.02*
		100%	22.99	30.28	26.14*

Table 6.10: **Articles: varying the ESL data size for adaptation**. All AP-adapted models use a value of 0.80 for inflation. The adapted models that outperform the corresponding native-trained model are marked with an asterisk. The size of the ESL data for adaptation is shown as a percentage of the ESL training data (100% corresponds to 60,743 articles).

Table 6.10 shows the performance of AP and NB models for the article correction task where the amount of annotation used for adaptation is varied. The adapted models that outperform the corresponding native-trained models are marked with an asterisk. As we add more data for error estimation, the performance improves. At 5% of the ESL data, we approach a saturation point, after which the performance remains about the same. This point is highlighted in the table.

It is interesting to note that NB models adapted with the smallest amounts of data get the best results. The reason for this behavior is that when 2% or less of the annotated data is used, we do not get examples of some less common article errors,[2] specifically confusions between *a* and *the* and vice versa, and these confusions thus get 0 probabilities in such a model. . We hypothesize that NB benefits from not suggesting these corrections.

Table 6.11 shows the performance on preposition mistakes. We find that, in contrast to article errors, more data is needed in order for the model to achieve a saturation point (10%). One reason for this is that one needs about twice as much annotated data to obtain a comparable number of examples for each preposition as we obtain for each article. With less than 2% of training data

[2]We refer the reader to Table 6.4 in Section 6.2.3 for statistics on article mistakes.

Model	**Training size**		**Performance**		
	Native	**ESL**	**P**	**R**	**F1**
Native-trained (AP)	5M	-	26.07	22.80	24.32
Adapted (AP)	5M	0.5%	20.65	10.13	13.59
		1%	24.19	10.00	14.15
		5%	31.62	19.73	24.30
		10%	31.27	22.27	**26.01***
		50%	31.03	22.13	25.83*
		100%	31.73	22.93	26.62*
Native-trained (NB)	5M	-	17.75	24.13	20.45
Adapted (NB)	5M	0.5%	20.70	9.47	12.99
		1%	22.29	10.13	13.93
		5%	22.84	20.80	21.77*
		10%	21.22	27.47	**23.93***
		50%	21.96	28.40	24.76*
		100%	22.51	27.07	24.57*
Native-trained (NB, Google)	26,000M	-	28.47	27.87	28.16
Adapted (NB, Google)	26,000M	0.5%	24.23	11.47	15.56
		1%	24.86	12.13	16.30
		5%	30.98	25.73	28.11
		10%	31.42	30.00	**30.69***
		50%	32.05	28.93	30.41*
		100%	31.33	29.20	30.22*

Table 6.11: **Prepositions: Varying the ESL data size for adaptation**. All AP-adapted models use value of 0.80 for inflation. The adapted models that outperform the corresponding native-trained model are marked with an asterisk. The size of the ESL data for adaptation is shown as a percentage of the ESL training data (100% corresponds to 31,000 prepositions).

used for error estimation, the models perform very poorly, since many of the confusions are not observed.

To summarize, the evaluation above suggests that we need about 3,000 examples to estimate error statistics reliably (5% of the training data for articles, and 10% for prepositions). For prepositions, the performance continues to improve slightly with more data but for articles, it does not change much. This is to be expected, since preposition errors have more parameters due to a larger confusion set. This is also confirmed for systems that are trained entirely on ESL data (Gamon, 2010).

Task	Model	Performance		
		P	R	F1
Articles	ESL-trained model (natural errors)	73.42	16.86	27.42
Prepositions	ESL-trained model (natural errors)	81.82	8.74	15.78

Table 6.12: **Performance of ESL-trained AP models.** All models are trained on 100% of ESL training data with naturally-occurring mistakes.

Model	Training size		Adapt.	Performance		
	Native	ESL	rate	P	R	F1
ESL-trained (natural errors)	-	90%	-	81.82	8.74	15.78
ESL-trained (inflation)	-	90%	1.0	66.67	11.65	19.83
			0.9	54.02	15.21	23.73
			0.8	38.57	17.48	24.05
			0.6	26.44	22.33	**24.21**

Table 6.13: **Prepositions: AP Inflation experiments for ESL-trained models.** All models are trained on 90% of data and evaluated on the development data (10% of all the training data).

6.4.3 Error Inflation for ESL-trained Models

We now discuss another application of the error inflation algorithm proposed in Section 6.2.2. When AP models are trained on annotated ESL data with naturally-occurring mistakes, they also suffer from low recall due to error sparsity, (e.g., Dahlmeier and Ng, 2011; Xiang et al., 2013). Table 6.12 shows the performance of AP models trained on ESL data with naturally-occurring mistakes. The recall of these models is very low – 16.86% for articles and 8.74%. The error inflation method can also be applied to such models to boost recall. In this case, we generate artificial errors in training, in addition to the naturally-occurring mistakes already in the data. Table 6.13 shows preposition models trained on the ESL data with and without error inflation. The first row in the table shows the results of training on natural errors. By inflating the error rate, we can boost the recall of the models significantly, from 8.75% to over 20% on the preposition correction task.

6.4.4 Comparison of Adapted Models and ESL-trained Models

We now evaluate ESL-trained models that use various amounts of data and compare these to adapted models. Table 6.14 compares adapted models that use 10% of all ESL training data available and ESL-trained models that use different amounts of data. The table demonstrates that due to adaptation we are able to train models that perform significantly better than ESL-

Task	Model	Training size		Performance		
		Native	ESL	P	R	F1
	Adapted	3M	10%	29.71	34.96	**32.11**
Articles	ESL-trained	-	10%	26.75	13.21	17.68
		-	20%	31.52	17.68	22.65
		-	50%	33.81	21.54	26.31
		-	100%	36.81	25.81	30.34
Prepositions	Adapted	13M	10%	31.27	22.27	**29.48**
	ESL-trained	-	10%	20.63	11.33	14.62
		-	20%	27.89	14.80	19.33
		-	50%	28.97	19.20	23.09
		-	100%	32.33	22.80	26.73

Table 6.14: **Comparison of adapted and ESL-trained models.** Adapted models use 10% of the ESL data (6,000 and 3,000 article and preposition examples) and are trained on *WikiNYT* with the AP algorithm. Inflation parameter is optimized on the development set. For the adapted models, we use 0.8 in all cases. For the ESL-trained models we use 0.9 and 0.7, for articles and prepositions, respectively. 100% of the ESL data contains about 60,000 articles and 31,000 prepositions.

trained models. Adapted models perform much better than ESL-trained models that use the same amounts of data: on the article correction task, the adapted model obtains an F1 of 32.11, while the ESL-trained model obtains 17.68; similarly, for prepositions, the F1 results are 29.48 and 14.62, respectively. Moreover, the adapted models require only a little bit of annotation and outperform significantly ESL-trained models that use considerably more data. In fact, on both tasks, the models adapted with 10% of the ESL data outperform ESL-trained models that use 10 times more ESL data for training. This is because adapted models are better at estimating parameters that do not depend on the source word but depend on the contextual information. Finally, note also that one can easily increase the training size of the adapted models, without incurring additional annotation costs.

6.4.5 Artificial Versus Naturally-Occurring Mistakes and the Genre Effect

As we showed in Section 6.4.4, the performance of adapted models that use several million examples from native English data is better than that of ESL-trained models that are built using 100 times less data (Table 6.14).

Even though we can significantly improve over ESL-trained models, this requires a lot of native data. There might be two reasons for this behavior. First, there might be a difference between how

Model	Articles			Prepositions		
	P	**R**	**F1**	**P**	**R**	**F1**
ESL-trained (natural errors)	73.42	16.86	27.42	81.82	8.74	15.78
ESL-trained (corrected with artificial errors)	60.00	5.23	9.62	82.61	6.15	11.44

Table 6.15: **Natural versus artificial errors for ESL-trained models**: *Natural errors* denotes models that are trained on data with naturally-occurring mistakes. *Corrected with artificial errors* denotes models that are trained on data where naturally-occurring errors are corrected and instead artificial errors are introduced using error statistics of the naturally-occurring mistakes. All models are trained on 100% of ESL training data.

Model	Articles			Prepositions		
	P	**R**	**F1**	**P**	**R**	**F1**
ESL-trained (natural errors)	73.42	16.86	27.42	81.82	8.74	15.78
ESL-trained (corrected with artificial errors)	60.00	5.23	9.62	82.61	6.15	11.44
Adapted	15.38	1.74	3.13	100.0	0.97	1.92

Table 6.16: **Genre**. Comparison of ESL-trained models and adapted models that use the same number of training examples. *Natural errors* denotes ESL-trained models that are trained on data with naturally-occurring mistakes. *Corrected with artifical errors* denotes ESL-trained models with naturally-occurring errors corrected and artificial errors introduced instead. *Adapted* models are trained on the same number of training examples from WikiNYT as the ESL-trained models (about 60,000 articles and 31,000 prepositions). Adapted models use 100% ESL training data for statistics.

naturally-occurring and artificial errors are distributed. Second, this ceiling in the performance of the adapted models might also be caused by genre differences between native and non-native texts.

To determine whether there exists a difference between naturally-occurring and artificial errors, we generate ESL training data, where all natural errors are corrected, and instead errors are added artificially, using the artificial errors approach. We then train AP models on corrected ESL data. Table 6.15 compares models trained on ESL data with naturally-occurring and artificial mistakes. We find that there is indeed a difference between these models; both precision and recall drop, for prepositions slightly, and more so for articles, when artificial errors are used instead of the naturally-occurring ones. We conjecture that contexts containing natural mistakes may have other cues that indicate the presence of an error: for example, many article errors have noun number mistakes on neighboring words. However, while the performance of the corrected text models drops slightly compared to the natural errors models, it remains quite good.

To investigate the genre effect, we control for the training size, when training on native and ESL data (Table 6.16). Note that the performance of these models is very poor, since the native-trained models use only 31,000 and 60,000 training examples. It is clear from this result that genre is a key factor affecting the performance of native-trained adapted models.

6.5 Discussion

We have presented two adaptation approaches based on error statistics: the *artificial errors* method, applicable for any discriminative model; and the *priors* method, a special adaptation approach for NB. We showed that the adaptation methods require significantly less annotation than training on annotated data, since estimating the *source* features parameters requires much less annotated data than learning all model parameters. Moreover, the training data size is not restricted by the amount of the error-tagged data available. The source word of the writer can be used in training as a feature, in the exact same way as with the models trained on error-tagged data.

We note that error regularities can be reflected at various levels, from specific authors, to writers of specific first language, to more general patterns across several languages. For first-language adaptation, we refer the reader to Rozovskaya and Roth (2010c,b, 2011); for adaptation to a small group of learners, see Rozovskaya et al. (2011).[3] Adaptation experiments in Chapter 8 can also be considered as first-language adaptation, as the majority of writers share the same first language. In this chapter, we applied the adaptation idea to ESL writers across several first-language backgrounds and showed that we only need a small ESL sample to estimate error statistics. But one would expect that even fewer annotated examples will be needed to adapt on first-language basis or to specific users, since writers make consistent mistakes. In fact, in Rozovskaya et al. (2011), we show that when the adaptation techniques are applied based on writing samples from a small group of learners, the improvements over the native-trained models are even more impressive, even though the data required for error estimation is much smaller.[4]

[3]Experiments in Rozovskaya and Roth (2010c,b) and Rozovskaya and Roth (2011); Rozovskaya et al. (2011) are not included in the thesis, as they are superseded by later works.

[4]Rozovskaya et al. (2011) describes the University of Illinois system in the HOO-2011 shared task. This system ranked first among six participating teams.

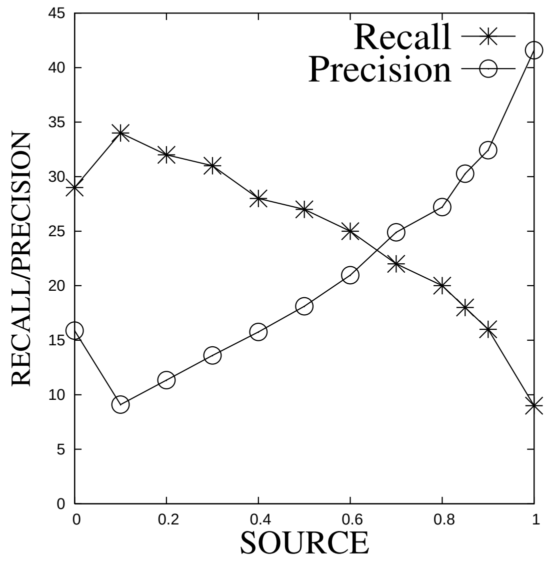
6.5.1 Advantages of the Two Adaptation Methods

In Chapter 5, we showed that NB and AP outperform both the language modeling approach and the counting method *SumLM* on the article and preposition error correction tasks. We have also shown that AP performs consistently better than NB with different feature sets on both tasks. Clearly, for a robust system, one would want to train a discriminative model and adapt it using the *artificial errors* approach. However, the NB adaptation has its own advantages. First, it does not require generating new training data and re-training in order to adapt to a different error distribution. One model is trained, and only at decision time are the prior parameters adapted. Second, in certain scenarios, it is much easier to train a NB model. Specifically, the Google corpus contains precomputed n-gram counts; these can be used directly to train NB; if we train a discriminative model, such as AP, on the Google corpus, we have to restrict the context features to the window of two words around the target. In Section 6.3 (Table 6.7) we showed that the Google model outperforms the AP classifier, when the latter is trained on a smaller data set (WikiNYT).

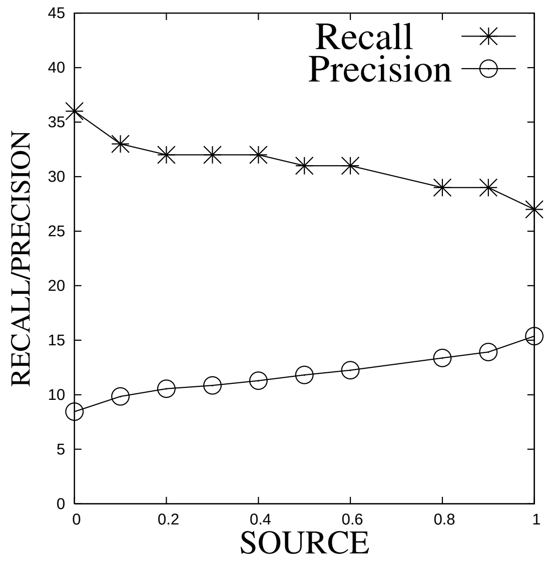
6.5.2 Supporting a Range of Recall Values

In Chapter 3, we discussed two training paradigms for error correction distinguished by whether the source word is used in training. Adaptation allows us to provide models with knowledge about error regularities. Both the *artificial errors* method and the *priors* method improve over the native-trained models; however, unlike with the NB adaptation method, the AP model adapted without error inflation suffers from low recall (Section 6.2.2), which can be fixed by inflating the error rate in the training data.

Supporting a range of recall values is important, since the desired recall may depend on the application of the error correction system and on the proficiency of the user. For example, more proficient users may be more interested in having more of their mistakes detected, since they will be able to determine the correct usage once an error is flagged. While our experiments have presented a large range of recall values for different inflation values, we believe that as recall is increased, good precision should be maintained. However, as shown in this Chapter and in will be shown in Chapter 8, models trained without error inflation exhibit very low recall. The inflation method helps boost the recall, while maintaining good precision.



(a) AP



(b) NB

Figure 6.2: **Source inflation continuum** for preposition errors (the FCE data set). The y axis shows the best recall achieved by the model and the corresponding precision, when predictions are done without thresholding them; this is the range available for a user to play with.

Analyzing the error inflation approach from the learning point of view, the method can be viewed as a middle ground that provides a range of degrees of the employment of the source text. In the *selection* model, the degree is 0, while in adaptation without inflation, the source text is fully used. By varying the degree, we can adapt the model to the needs of users with different proficiency levels. Figure 6.2(a) and Figure 6.2(b) illustrate how precision and recall change across the continuum for the AP and NB models. The y axis shows the maximum recall achieved (when predictions are done without thresholding – whenever the model suggests a replacement, we accept it) and the precision corresponding to this recall. Note that the recall range for AP varies from 8% to 35%.

Note also that the NB model is not severely affected by the low error rate – its recall ranges between 26% and 35%. However, the NB classifier never achieves the good precision of the AP classifier.

6.6 Summary

This chapter develops methods that provide models trained on native English data with knowledge about error regularities with minimal annotation costs. The proposed methods are based on error statistics and allow for training models in the *correction* training paradigm using the *source* word as a feature. The resulting *adapted* models combine large amounts of native English data (which is cheap and can be used to learn parameters for context features) with small amounts of annotated ESL data for estimating error patterns. This is possible because the parameters relating to error patterns are very simple and thus do not require a lot of annotated data for estimation.

The adapted models outperform native-trained models that use the same amount of native data for training. The adapted models also perform much better than ESL-trained models that use the same amounts of data or even 10 times more annotated data for training. This is because, unlike the *source* feature, context features have many more parameters and require a lot more data for estimation. It is thus essential to use native data for learning the parameters of the context features.

The implementations of the adapted models described in this chapter were part of the error correction systems in three text correction competitions; these systems ranked first or second on various metrics in the HOO-2012 shared task and ranked first on all metrics in the HOO-2011 and the CoNLL-2013 shared tasks.

Chapter 7

Errors on Open-Class Words: Nouns and Verbs

In this chapter, we study mistakes on open-class words and focus on noun number misuse and grammatical verb errors. Developing models for correcting errors that belong in the open-class category introduces additional challenges that are not discussed by previous research. We develop strategies for correcting these errors in a linguistically-motivated framework. It is demonstrated that addressing these challenges in the right way is crucial to building a state-of-the-art system for correcting these errors.

7.1 Introduction

Verb and noun errors are some of the most prevalent mistakes made by non-native writers of English but the least studied. The reason is that dealing with open-class mistakes requires a new paradigm; essentially all research done on correcting ESL errors assumes a closed set of triggers – e.g., correcting the use of prepositions or articles – but identifying mistakes on open-class words necessitates identifying potentially ambiguous triggers first: nouns and verbs cannot be identified in text based on their surface forms, since many of them are ambiguous with respect to part of speech. This is complicated by the fact that since ESL text is noisy, pre-processing tools tend to do poorly on learner data (Nagata et al., 2011). Thus the candidate extraction procedures for errors on open-class words need to address the problem of noisy data. Moreover, correcting verb errors presents an additional challenge because verbs fulfill many grammatical functions and are marked for different properties, which results in a variety of mistakes.

This chapter develops strategies for addressing mistakes on open-class words. The problem is considered in the context of correcting several prominent grammar and usage errors on two central parts of speech: nouns and verbs. For both verb and noun errors, we consider the problem of identifying the relevant candidates in noisy learner text. It is shown that a robust candidate selection method must take into account the problem of the noisy input and poor performance of the pre-processing tools.

Furthermore, we propose a linguistically-motivated approach to verb error correction that makes use of the notion of *verb finiteness* to identify types of verb errors, before using a statistical machine learning approach to correct these mistakes. We show that the linguistically-informed model significantly improves performance on correcting verb errors.

Contributions

- We evaluate several *candidate identification methods* for noun number errors, using the CoNLL-2013 data set. Comparison of our method with other approaches used in the competing systems reveals that the choice of the method is crucial for system performance.

- We present a holistic, linguistically-motivated framework to the problem of correcting grammatical verb mistakes; our approach "starts from scratch" without any knowledge on which mistakes should be corrected or on the mistake type; in doing that we show that the specific challenges of verb error correction are better addressed by first predicting the finiteness type of the verb in the error identification stage.

  1. Within the proposed model, we describe and evaluate several methods of *selecting verb candidates*, an algorithm for *determining the verb type*, and a type-driven *verb error correction system.*
  2. We annotate a subset of the FCE data set with gold verb candidates and gold verb type.

The rest of this chapter is devoted to developing models for grammatical verb errors (Section 7.2) and noun number mistakes (Section 7.3).

7.2 Correcting Grammatical Verb Errors

While verb errors occur more often than article and preposition mistakes, with a few exceptions (Lee and Seneff, 2008b; Gamon et al., 2009; Tajiri et al., 2012), there has been little work on verbs. There are two reasons why it is difficult to deal with verb mistakes. First, in contrast to articles and prepositions, verbs are more difficult to identify in text, as they can often be confused with other parts of speech, and pre-processing tools are known to make more errors on noisy ESL data (Nagata et al., 2011). Second, verbs are more complex linguistically: they fulfill several grammatical functions, and these different roles imply different types of errors.

The linguistic complexity of verb errors has led all previous work on verb mistakes to assume prior knowledge of the mistake type; however, identifying the specific category of a verb error is nontrivial, since the surface form of the verb may be ambiguous, especially when that verb is used incorrectly. Consider, for instance, the following examples of verb mistakes:

1. "We **discusses/discuss* this every time."
2. "I will be lucky if I *{*will find*}/*find* something that fits."
3. "They wanted to visit many places without **spend*/*spending* a lot of money."
4. "They arrived early to **organized*/*organize* everything".

These examples illustrate three grammatical verb properties – *Agreement*, *Tense*, and non-finite *Form* choice – that encompass the most common grammatical verb problems for ESL learners. The first two examples show mistakes on verbs that function as main verbs in a clause: sentence (1) shows an example of subject-verb *Agreement* error; (2) is an example of a *Tense* mistake, where the ambiguity is between {*will find*} (Future tense) and *find* (Present tense). Examples (3) and (4) display *Form* mistakes: confusing the infinitive and gerund forms in (3) and incorrectly adding an inflection on an infinitive verb in (4).

This section proposes a linguistically-informed approach to addressing the specific challenges of verb error correction that have not been addressed previously – identifying verb candidates in text and determining which class of errors is present, before proceeding to correct the error. The experimental results show that our linguistically-motivated approach benefits verb error correction. In particular, in order to determine the error type we build on the notion of *verb type* that distinguishes between *finite* and *non-finite* verbs (Quirk et al., 1985) and grammatical properties that are associated with each group, i.e. *Agreement* and *Tense* (examples (1) and (2) above) and *Form* choice (examples (3) and (4) above), respectively (see Section 7.2.1).

Tag	**Error type**	**Rel. freq. (%)**
TV	Tense (but see also Table 7.3)	40.0
FV	Form	22.3
AGV	Verb-subject agreement	11.5
MV	Missing verb	11.7
UV	Unneccesary verb	7.3
IV	Inflection	5.4
DV	Derivation	1.8
Total		6,640

Table 7.1: **Grammatical verb errors in FCE.**

7.2.1 Verb Errors in ESL Writing

We study verb errors using the FCE corpus (see Section 4.2.2). The corpus possesses several desirable characteristics for studying verb errors: it is large (500,000 words) and has special error tags for different error categories. Verb errors are more common in FCE than article and preposition mistakes: 5,056 determiner errors, 5,347 preposition errors, and 6,640 grammatical verb mistakes, marked using seven different tags (Table 7.1).

7.2.1.1 Classifying Verb Errors Using Verb Finiteness

There are many grammatical categories for which English verbs can be marked. The linguistic notion of verb finiteness or *verb type* (Radford, 1988; Quirk et al., 1985) distinguishes between verbs that function on their own in a clause as main verbs (finite) and those that do not (non-finite). Grammatical properties associated with each group are mutually exclusive: tense and agreement markers, for example, do not apply to non-finite verbs; non-finite verbs, on the other hand, are not marked for many grammatical functions but may appear in several forms in English.

The most common verb problems for ESL learners – *Tense*, *Agreement*, non-finite *Form* – involve verbs both in finite and non-finite roles, which is not surprising: the English verb tense system is more complex compared to other languages, and the presence and distribution of non-finite forms varies by language as well. Table 7.2 illustrates contexts that license finite and non-finite verb types.

Our intuition is that knowledge of verb finiteness should benefit models that perform verb error correction because properties associated with each verb type are mutually exclusive. An observed

Verb type	**Example**	**Verb properties**		
		Agreement	**Tense**	**Form**
Finite	"He *discussed* this with me last week"	-	Past Simple	-
	"He *discusses* this with me every week."	3rd pers., sing.	Present Simple	-
Non-finite	"He left without *discussing* it with me."	-	-	Gerund
	"They let him *discuss* this with me."	-	-	Infinitive
	"*To discuss* this now would be ill-advised."	-	-	to-Infinitive

Table 7.2: **Contexts that license finite and non-finite verbs and the corresponding active properties.**

verb error may be due to several grammatical phenomena, and knowing which phenomena are active depends on the function of the verb in the current context. Indeed, as we show, using verb type information is beneficial for verb error modeling. Note that *Agreement*, *Tense*, and *Form* errors account for about 74% of all grammatical verb errors in Table 7.1, but in fact, the verb type formulation is also relevant for the remaining mistakes but is out of the scope of this work.[1]

7.2.2 Annotation for Verb Type

In order to evaluate the quality of the *algorithm for verb type assignment* and the *candidate selection* methods, we annotated all verb errors in the corpus and 10% of correct verbs with the information about verb type. Correct verbs were annotated in a random set of 124 documents. We refer to these 124 documents that have gold annotation for all verbs as *gold subset*.

The annotation was performed by two students with formal background in Linguistics. The inter-annotator agreement is shown in Table 7.4 and is quite high.

Annotating Verb Errors

We collected all mistakes tagged as Tense (TV), Agreement (AGV), and Form (FV) in the FCE corpus (4864 mistakes). For each verb error, the annotators marked the type of the verb (finite or non-finite) and the type of error (Tense, Agreement, or Form) (Table 7.3). Note that the annotation was necessary because these error tags do not always correspond to the three error types we study here; for example, the FV tag sometimes marks errors on finite verbs, e.g. tense. Overall, about 7% of all annotated verb errors have to do with phenomena different from the three verb properties considered here, and we exclude them.

[1]For instance, the missing verb errors (MV, 11.7%) require an additional step for identifying contexts for missing verbs, and then appropriate verb properties need to be determined based on the verb type. The additional step, however, is orthogonal to the phenomena that we consider.

Errors by Verb Type	Error type	Example
Finite (67.7%)	Agreement (20%) Tense (80%)	"We *discusses*/discuss* this every time." "If you buy something, you {*would be*}*/{*will be*} happy."
Non-finite (25.3%)		"If one is famous he has to accept the disadvantages of *be*/being* famous." "I am very glad {*for receiving*}*/{*to receive*} it." "They arrived early to *organized*/organize* everything."
Other errors (7.0%)	Passive/Active (42.3%) Compound (40.7%) Other (16.8%)	"Our conference party {*is included*}*/*includes* dinner and dancing." "You ask me for some *informations*/information* – here *they*/it are*/is*." "Nobody {*has to be*}*/{*should be*} late."

Table 7.3: **Verb error classification** based on 4864 mistakes marked as TV, AGV, and FV errors in the FCE corpus.

Category	Agreement	Kappa	Random
Correct verbs	0.97	0.95	0.51
Erroneous verbs	0.88	0.81	0.41

Table 7.4: **Inter-annotator agreement** based on 250 verb errors and 250 correct verbs, randomly selected.

Annotating Correct Verbs

The annotators were presented with an automatically-generated list of candidate verbs. Each verb was shown in the context of the sentence. The candidates that were identified incorrectly due to mistakes by the part-of-speech tagger were marked as invalid. For valid verbs, the annotators specified the verb type.

7.2.3 The Computational Model for Verb Error Correction

The verb error correction problem is formulated as a classification task in the spirit of the machine learning paradigm to error correction that was also used for article and preposition error correction in Chapters 5 and 6, with the exception that the verb error correction model includes additional components. The complete system consists of the following components:

1. Candidate selection (7.2.3.1)
2. Determining the verb type (7.2.3.2)
3. Feature generation (7.2.3.3)
4. Error identification (7.2.3.4)
5. Correction component (7.2.3.5)

After verb candidates are selected, verb type is determined and features are generated for each candidate. The *verb type prediction* is used in the *error identification* component. Given the output of the error identification stage, the corresponding classifiers for each error type are invoked to propose an appropriate correction.

We use the FCE data both for training and evaluation. We randomly split the documents of the corpus into two equal parts, training on one part and evaluating on the other. Each component is developed using the training data.

7.2.3.1 Candidate Selection

This stage selects the set of words that are presented as input to the classifier. This is a crucial step for models that correct mistakes on open-class words, because it limits the performance of any system: those errors that are missed at this stage have no chance of being detected by the later stages. This is also a challenging step, as the class of verbs is open, with many English verbs being easily confused with other parts of speech. We implement four candidate selection methods. Method (1) extracts all verbs heading a verb phrase, as identified by a shallow parser (Punyakanok and Roth, 2001). Method (2) expands this set to also include words tagged with one of the verb tags {VB,VBN,VBG,VBD,VBP,VBZ}. However, generating candidates by selecting only those tagged as verbs is not good enough, since the POS tagger performance on ESL data is known to be suboptimal (Nagata et al., 2011). This especially concerns those verbs that contain errors. For example, verbs lacking agreement markers are likely to be mistagged as nouns (Lee and Seneff, 2008b). Erroneous verbs are exactly the cases that we wish to include. Methods (3) and (4) address the problem of pre-processing errors. Method (3) adds words that are on the list of valid English verb lemmas; the latter is compiled using a POS-tagged version of the NYT section of the Gigaword corpus. Method (4) also includes words not tagged as verbs but whose lemmas are in the list of valid verb lemmas. The difference between methods (3) and (4) is that (4) will also add words with inflections; for example, the former will add *shop* but not *shopping*, but the latter will add both.

Methods (3) and (4) require a procedure that given a possibly inflected verb can generate its lemma. To this end, we develop a tool called *verbMorph* that performs morphological analysis on verbs. Unlike lemmatizers, *verbMorph* both adds and removes morphological endings for a specific morphological form. The module assumes (1) a list of valid verb lemmas and (2) a list of irregular

Candidate identification method	**Error recall**	**Recall by error category**		
		A	**T**	**F**
(1) All verb phrases	83.00	86.62	93.55	59.08
(2) + tokens tagged as verbs	91.96	90.30	94.33	87.79
(3) + tokens that are valid verb lemmas	95.50	95.99	96.46	93.23
(4) + tokens with inflections whose lemmas are valid verb lemmas	**96.09**	**96.32**	**96.62**	**94.84**

Table 7.5: **Performance of the candidate selection methods on the FCE data set.** *Recall* denotes the percentage of gold errors identified by each method.

A verb is non-finite if any of the following hold:	**A verb is finite if any of the following hold:**	**Accuracy on**	
		Correct verbs	**Incorrect verbs**
(1) $[numTokens = 2] \wedge [firstToken = to]$ (2) $firstToken = be$ (3) $[numTokens = 1] \wedge [pos = VBG]$	(1) All verbs identified by shallow parser (2) *can;could* (3) $[numTokens = 1] \wedge [pos \in \{VBD, VBP, VBZ\}]$ (4) $[numTokens = 2] \wedge [firstToken! = to]$ (5) $numTokens > 2$	98.01	89.4

Table 7.6: **Algorithm for determining verb finiteness.** *numTokens* denotes the number of tokens in the verb instance, e.g. for the verb instance *to go*, $numTokens = 2$. Verbs not covered by the rules are not assigned any verb type. The last column shows algorithm accuracy on the gold subset separately for correct and incorrect verbs.

English verbs and their morphological variants. *VerbMorph* is also used to generate morphological variants for verbs (see Section 8.4.3).

The quality of the identification methods is evaluated by verifying how many verbs that have errors can be identified in text using each of the methods. Table 7.5 shows for each method its recall on gold verb errors. The last two methods that address pre-processing mistakes are able to recover more erroneous verb candidates in text. It is also interesting to note that across all methods, the highest recall is obtained for tense errors. In fact, method (2) that relies on the POS tagger misses 10% of all agreement mistakes and over 12% of non-finite errors but less than 6% of tense mistakes. This suggests that the POS tagger is more prone to failure due to errors in agreement and form. This makes sense since we would not typically expect errors in tense to disrupt the structure of the parse tree or the POS sequence. We stress that the evaluation in Table 7.5 shows the ability of each method to select as part of its candidates erroneous verbs but it does not evaluate performance on error detection and correction. In Section 7.2.4 we also evaluate the contribution of each method to error identification.

7.2.3.2 Automatic Prediction of Verb Type

Predicting verb finiteness is not trivial, as almost all English verbs can occur in both finite and non-finite form, and the choice depends on the function of the verb in the sentence. Moreover, the surface forms of a verb in finite and non-finite forms may be the same.

While we cannot learn verb type automatically due to lack of annotation, we show, however, that it is possible to reliably predict verb finiteness for the majority of verbs using linguistic knowledge. We implement a decision-list classifier that makes use of a small set of linguistically-motivated rules (Table 7.6). The algorithm covers about 92% of all verb candidates, abstaining on the remaining highly-ambiguous 8%.

The accuracy of the algorithm that predicts verb type is evaluated on the gold subset (shown in the last column of Table 7.6). Accuracy is the percentage of verbs for which the automatic type prediction is the same as gold. We note that despite its simplicity, overall, this method is highly effective, which indicates that it is possible to make use of linguistic knowledge in a straightforward way. The performance is better, however, for correct verbs than for verb errors, as errors downgrade performance.

7.2.3.3 Features

We define features based on context around the target, where the *target* refers to the verb, including its auxiliaries or the infinitive marker (e.g., *found*, *will find*, *to find*). The *baseline* features are word n-grams in the 4-word window before and after the target defined in the same way as for article in preposition error correction tasks in Section 5.2.3. Additional features are intended to characterize a given error type: for *Agreement* and *Form* errors, we use a syntactic parser (Marneffe et al., 2006) and define features that reveal relations between the target verb and the neighboring words. We denote these features by *syntax*. Syntactic knowledge via tree patterns has been shown useful for agreement mistakes (Lee and Seneff, 2008b). Features for *Tense* include temporal adverbs in the sentence and tenses of other verbs in the sentence and are similar to the features used in other verb classification tasks (Reichart and Rappoport, 2010; Lee, 2011; Tajiri et al., 2012). The features are shown in Table 7.7.

	Agreement	**Description**
(1)	subjHead, subjPOS	The surface form and the POS tag of the subject head
(2)	subjDet {those, this,..}	Determiner of the subject phrase
(3)	subjDistance	Distance between the verb and the subject head
(4)	subjNumber {*Sing, Pl*}	*Sing* – singular pronouns and nouns; *Pl* – plural pronouns and nouns
(5)	subjPerson {*3rdSing, Not3rdSing, 1stSing*}	*3rdSing* – she,he,it,singular nouns; *Not3rdSing* – we,you,they, plural nouns; *1stSing* – "I"
(6)	conjunctions	(1)&(3);(4)&(5)

	Tense	**Description**
(1)	verb phrase (VP)	verb lemma, negation, surface forms and POS tags of all words in the verb phrase
(2)	verbs in sentence (4 features)	tenses and lemmas of the finite verbs preceding and following the target verb
(3)	temporal adverbs (2 features)	temporal adverb before and after the target verb
(4)	bag-of-words (BOW)(8 features)	Includes the following words in the sentence: {if, when, since, then, wish, hope, when, since, after}

	Form	**Description**
(1)	closest word	surface form, lemma, POS tag, and distance of the closest open-class word to the left of the verb
(2)	governor	surface form, POS tag and dependency type of the target
(3)	preposition	if the verb is preceded by a preposition: preposition itself and the surface form, POS tag and dependency of the governor of the preposition
(4)	target pos and lemma	POS tag and lemma of the verb and their conjunctions with features in (2) and (3) and word ngrams

Table 7.7: **Features used, grouped by error type.**

7.2.3.4 Modeling Error Identification

This section describes the error identification system that makes use of the components presented above. The goal of this stage is to identify errors and for each error predict its type. The input to the system is the list of candidate verbs (identified as described in Section 7.2.3.1). A multiclass classifier is trained with the following label space: {*Correct*, *Form*, *Agreement*, *Tense*}.

To evaluate the contribution of verb type, we implement two error identification models: *combined* and *type-based*. The *combined* model is agnostic to the type of the verb. It makes use of *all* the features we have defined earlier for each verb. The *type-based* model, in contrast to the *combined*, uses the type of the verb predicted by the verb type classifier.

There are two ways to use the information about verb type in the type-based model. A *soft* way is to keep the same setting as the one used in the combined approach but add the predicted verb type as a feature. The other – *hard*-decision approach – is to use only a subset of the features depending on the predicted type: *Agreement* and *Tense* for the finite verbs, and *Form* features for non-finite.

7.2.3.5 Correction Component

The role of the correction component is to propose an appropriate correction for those verbs that are identified as errors. The correction component consists of three classifiers, each corresponding to one type of mistake. Given the output of the error identification model, an appropriate classifier is run for each instance predicted to be a mistake.[2] The verb type predictor is used to select finite instances for training the *Agreement* and *Tense* components and non-finite – for the *Form* component. The label space for *Tense* specifies tense and aspect properties of the English verbs (see Tajiri et al., 2012 for more detail); the *Agreement* component specifies the person and number properties; while the *Form* component includes the commonly confusable non-finite English forms (see Table 7.2). These components are trained as multiclass classifiers.

7.2.4 Experiments

The main goal of this study is to propose a unified framework for correcting verb mistakes. We do not focus on features and on the specific learning algorithm but instead on how to model verb error correction. Thus, our experimental study addresses the following research questions:

I. **Linguistic questions**:
   - (i) Candidate selection methods
   - (ii) Contribution of verb type to error identification

II. **Computational Framework**: Error identification versus correction

III. **Gold annotation**:
   - (i) Using gold candidates and gold verb type versus automatically predicted ones
   - (ii) Performance comparison by error type

All of the identification and correction models are trained using the discriminative learning framework with the SVM learning algorithm (Chang et al., 2010). The choice of the model in favor of a discriminative classifier is based on the results obtained in Chapter 5.

[2]We assume that each verb contains at most one mistake. The number of verbs that have both agreement and tense errors is negligible, less than 1% of all errors.

Candidate identification	**AAUC**	
method (error recall, %)	**Combined**	**Type-based**
(1) 83.00	73.38	79.49
(2) 91.96	80.36	86.48
(3) 95.50	**81.39**	**87.05**
(4) 96.09	81.27	86.81

Table 7.8: **Comparison of candidate selection methods as performance on error identification.** The first column shows the percentage of gold errors selected by each method. Type-based models are discussed in detail in Section 7.2.4.1.

	Correct verbs	**Erroneous verbs**	**Error rate (%)**
Training	41,721	1,981	4.75
Test	41,836	2,014	4.81

Table 7.9: **Number of verb candidates in training and test data.** Examples are selected using method (3).

Evaluation The key results are shown using Precision/Recall curves. For all other experiments, AAUC is used as a summary statistic (Section 4.5). AAUC is computed over a range of 15 recall points:

$$AAUC = \frac{1}{15} \cdot \sum_{i=1}^{15} Precision(i).$$

7.2.4.1 Linguistic Questions

Candidate Selection Methods To evaluate candidate selection, we compare models that perform error identification using the four candidate selection methods presented in Section 7.2.3.1. Table 7.8 presents the results. Overall, better performance is achieved by methods with higher recall, with the exception of method (4). Although method (4) has the highest recall on gold errors, it does not translate into better performance in verb error identification, due to the amount of noise that is also added (the last method includes all candidates whose lemmas are valid English verbs). Following these results, method (3) is used in the remainder of the verb experiments. Table 7.9 shows the number of verb instances in training and test selected using this method. The last column shows the error rate, or the percentage of erroneous verbs. The test data contains 2,014 errors and has the error rate of 4.81%. 57.5% of the errors are Tense errors, 27.4% are Form, and 15.1% are Agreement errors.

Contribution of Verb Type Here we address the following questions: does the information about

Model	**AAUC**
Combined	81.39
Type-based I (soft)	81.11
Type-based II (hard)	**87.05**

Table 7.10: **Verb type contribution with soft and hard verb-type decisions**.

verb type improve verb error identification and what is the best way to incorporate this knowledge into the system?

In order to demonstrate that verb finiteness is important, we compare the baseline *combined* model to two *type-based* models on error identification. In Section 7.2.3.4, we described two ways of adding verb type knowledge to the error identification system: adding the predicted verb type as a feature in the combined model and selecting only the relevant features depending on the verb type. Table 7.10 shows the results of adding verb type information in the error identification stage. We find that the first approach does not provide improvement over the baseline combined model, while the second method is very effective. We conjecture that because verb type prediction is quite accurate, the hard-decision approach is preferred, as it provides knowledge in a direct way. One immediate advantage of the second method is that since *Agreement* and *Tense* errors only occur on finite verbs, it restricts the choice of labels for finite verbs to one of {*Correct*, *Agreement*, *Tense*}; similarly, the choice of labels for non-finite verbs is restricted to {*Correct*, *Form*}. Doing so reduces ambiguity during training and prediction. A second advantage of this type-driven approach is that the type-based model will use the appropriate features, which also facilitates learning. Henceforth, we will use the hard-decision method in the type-based model.

Figure 7.1 evaluates the *combined* model and the hard-decision *type-based* model presented in Table 7.10. Precision/Recall curves are generated by varying the threshold on the confidence of the classifier. This graph provides more detail than the AAUC results, since it reveals the behavior of the systems at multiple recall points: we observe that at every recall point the *type-based* classifier has higher precision.

So far, the models used all features defined in Section 7.2.3.3. While features are not the focus of this work, we show that the type-driven approach is applicable with different feature sets (Table 7.11). In fact, the *type-based* approach improves more with more sophisticated features, which is to be expected, since more complex features are tailored toward relevant verb errors. Adding features

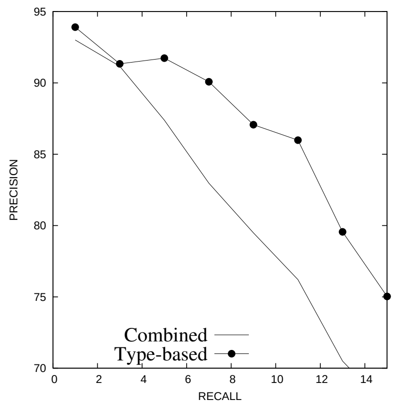


Figure 7.1: **Key result for verb type contribution**. AAUC shown in Table 7.10. The *combined* model uses no verb type information. In the hard-decision *type-based* model, verb type is assigned with the verb type algorithm and each verb uses the features for its type. The differences are statistically significant (McNemar's test, $p < 0.0001$).

Feature set	**AAUC**	
	Combined	**Type-based**
Baseline	46.62	49.72
All−Syntax	79.47	84.88
Full feature set	**81.39**	**87.05**

Table 7.11: **Verb type contribution for different features**.

specific to each error type significantly improves the performance over the baseline n-gram features. The rest of the experiments use all features (denoted *Full* feature set).

7.2.4.2 Identification Versus Correction

After running the error identification component, we apply the appropriate correction models to those instances identified as errors. The correction models are also multiclass classifiers trained on the relevant verb instances (finite or non-finite), as predicted by the verb type classifier. The error-identification model is a type-based hard-decision model. It is important to note that the correction components are also type-aware models, as they are trained on the relevant verbs according to the verb type prediction.

Correction components are evaluated by fixing a recall point in the error identification stage. The corresponding results on error identification and error correction are shown in Table 7.12.[3] We observe the low recall obtained by the models; it can be increased by modifying the decision

[3]The results on Agreement mistakes are identical for identification and correction, since Agreement errors are always binary decisions, unlike Tense and Form mistakes.

Error type	Correction			Identification		
	P	**R**	**F1**	**P**	**R**	**F1**
Agreement	90.62	9.70	17.52	90.62	9.70	17.52
Tense	60.51	7.47	13.31	86.62	10.70	19.05
Form	81.82	16.34	27.24	83.47	16.67	27.79
Total	71.94	10.24	17.94	85.81	12.22	21.20

Table 7.12: **Performance of the complete model after the correction stage.**

threshold but here we wish to ensure that a relatively high precision is maintained (see also verb experiments in Chapter 8 where models are optimized for F1 and use a different decision threshold). Note that the baseline (the percentage of correct verbs) for the verb error correction task is 95.19% (Table 7.9). The performance shown in Table 7.12 corresponds to an accuracy of 95.60% on error identification (error reduction of 8.7%) and 95.40% on correction (error reduction of 4.5%) over the baseline of 95.19%.

7.2.4.3 Analysis on Gold Data

We further analyze our model on the gold subset of the data. The set contains 7,784 gold verbs, including 464 errors, and we run experiments in 10-fold cross-validation, where on each run 90% of the documents are used for training and the remaining 10% are used for evaluation. Using the gold data we can show what the contribution of each module of the system is if the other components do not produce errors. The gold annotation can be used in two system components: (1) candidate selection and (2) verb type prediction.

Table 7.13 shows the performance on error identification when gold versus automatic settings are used. As expected, using the gold verb type is more effective than using the automatic verb type, both with automatic and gold candidate generation.

We also compare automatic and gold *candidate selection*. We find that 8.7% instances generated by the automatic candidate selection are noise, i.e. words that belong to other parts of speech that are incorrectly selected. Using gold candidates instead of automatically-generated ones demonstrably improves performance. The combined model improves by 14 AAUC points (from 55.90 to 69.86) when gold candidates are used. These results demonstrate that, while not an issue in other error correction tasks, candidate selection is an important component of an error correction

Candidate selection	**Verb type prediction**	**AAUC**
Automatic	None	55.90
	Automatic	74.72
	Gold	**89.45**
Gold	None	69.86
	Automatic	90.89
	Gold	**96.42**

Table 7.13: **Verb type contribution on the gold subset**. All models use the *Full* feature set. Value *None* for verb type prediction denotes the *combined* model.

Error type	**Combined**	**Type-based Automatic**	**Type-based Gold**
Agreement	86.80	88.43	**89.21**
Tense	18.07	25.62	**26.87**
Form	97.08	98.23	**98.36**

Table 7.14: **AAUC results for verb type contribution by error type using gold subset**.

system for open-class words.

Note that compared to the performance on the entire data set (Table 7.10), the performance of the models shown here that use automatic components is lower, which is to be expected, since the training sizes are different. On the other hand, because of the smaller training size, the gain due to the type-based approach is larger on the gold subset (19 AAUC points (rows 1 and 2) versus 6 AAUC points).

Finally, we evaluate the contribution of verb type for each error type separately. Table 7.14 compares the performance of the combined model to two type-based models: one where the type is predicted using the proposed verb type algorithm, and the other, where the gold type is used. It is clear from the results that all errors benefit form verb type information, although it is difficult to determine which error type benefits most, since performance varies by error type.

7.3 Correcting Errors in Noun Number

We now discuss mistakes in noun number. As shown in Table 4.4, frequency of noun number errors strongly depends on the first-language background; these errors are the second most common error type in the CoNLL corpus and account for 8.4% and 11.4% of all mistakes in the training and test sets, respectively, while in the other two data sets they occur three to four times less frequently.

Because the CoNLL data set has a large number of errors of this type, we make use of this data set for developing a model for correcting mistakes in noun number. This also gives us an opportunity to compare against approaches taken by other teams. Below we show an example of a noun number error:

- "Some countries are having difficulties in managing a place to live for their **citizen*/*citizens* as they tend to get overpopulated."

Among the system dimensions studied in the previous section that are relevant for verb error correction, here we address the problem of identifying noun candidates that are input to the classifier. The model itself is a binary NB classifier trained on the Google corpus (we discuss the choice of the training data and of the learning algorithm in Chapter 8).

7.3.1 Candidate Identification for Noun Number Mistakes

As discussed in Section 7.2, candidate identification is crucial for correcting mistakes on open-class words but is rarely carefully studied or evaluated. Several participating systems in the CoNLL-2013 competition mention how they identify noun candidates. Following these descriptions, we implement and compare three methods of candidate identification and show that the choice of a method has a considerable impact on error correction performance. The first method relies on the shallow parser and POS tagger and includes all common nouns, i.e. those tagged as *NN* or *NNS*, that head an NP. Two systems use the first method – Kao et al. (2013) and Xiang et al. (2013). The second method includes all words tagged as *NN* and *NNS* and is used in several other systems (e.g., Yoshimoto et al., 2013).

As is the case with the verb error correction system presented in Section 7.2, these procedures do not have a perfect result on the identification of all nouns that contain noun mistakes. To this end, we implement the third method that addresses the problem of pre-processing errors by adding words that end in common noun suffixes, e.g. "ment", "ments", and "ist".

The percentage of noun errors selected as candidates by each method and the impact of each method on error correction performance are shown in Table 7.15. The last method that addresses the problem of pre-processing mistakes recovers 50% and 43% of the candidates in training and

Candidate identification method	**F1**		**Error recall (%)**	
	Train	**Test**	**Train**	**Test**
NP heads	21.79	40.47	85.81	87.72
All nouns	21.73	41.08	87.80	89.50
Nouns+heuristics	**23.06**	**42.60**	93.00	92.84

Table 7.15: **Nouns: effect of candidate identification methods on the error correction performance.** All models are trained using the NB algorithm. *Error recall* denotes the percentage of noun errors that are identified in text by each method.

test data, respectively, missed by the first approach that only considers NP heads. Importantly, the last method also has the best result in error correction performance, improving F1 scores by over 1% absolute value on both training and test data. For noun number mistakes, performance is reported using the F1 score, as this is the metric adopted in the CoNLL-2013 shared task.

7.4 Summary

This chapter addresses special challenges of correcting mistakes on open-class words with an emphasis on errors in noun number and grammatical verb mistakes. It is shown that verb and noun errors are common for ESL writers but present additional challenges due to the fact that identifying verbs and nouns in (noisy) text may itself be difficult. Moreover, building a system that corrects verb errors is difficult due to the linguistic diversity of verb mistakes. We evaluate candidate identification methods for both types of mistakes and show that trigger identification must address the problem of the noisy input. This is important, since most errors made by non-native speakers cannot be identified by considering only closed classes (e.g., prepositions and articles), as most of the previous research assumes.

Furthermore, for verb error correction, a linguistically-inspired approach is developed that first identifies verb candidates, then makes use of the linguistic notion of *verb type* to characterize the type of mistake, and then proceeds to correct it. This approach thus provides a first step in considering more general algorithmic paradigms for correcting grammar and usage errors. Our approach integrates a statistical machine learning model with a rule-based system that encodes linguistic knowledge to yield the first general correction approach to verb errors (that is, one that does not assume prior knowledge of which mistake was made and where). The robustness of this approach is further validated in Chapter 8 within the framework of the CoNLL-2013 shared task.

Chapter 8

Building a State-of-the-Art Error Correction Model

So far, we have covered several prominent categories of ESL mistakes: article and preposition usage errors, and mistakes on nouns and verbs. This chapter describes a complete system that addresses these errors using the techniques proposed in the previous chapters. The system – referred to as *Illinois* – participated in the recent CoNLL-2013 shared task on grammatical error correction and placed first among 17 teams, winning by a large margin over the other submissions.

Four design principles that are relevant for correcting all of these errors are identified. These principles are closely aligned with the techniques presented in this thesis. Analysis of the system along these dimensions reveals how each of these principles contributes to performance. As a result, we also improve over the Illinois model.

8.1 Introduction

The recent growing interest in ESL error correction in the NLP community resulted in three error correction competitions. The Helping Our Own (HOO-2011, Dale and Kilgarriff, 2011) focused on correcting errors in papers published in NLP conferences and were authored by researchers who are not native English speakers. Even though the participants were free to address all types of mistakes present in the data set, all of the participating teams targeted only article and preposition usage errors, in addition to punctuation and spelling errors.[1] The HOO-2012 competition (Dale et al., 2012) focused on preposition and article mistakes in the FCE corpus. The CoNLL-2013 shared task, in addition to determiner and preposition usage errors, includes also mistakes in noun number, verb agreement, and verb form. The phenomena included in the CoNLL-2013 shared task correspond very closely to the mistakes considered in the previous chapters, with the exception of verb form

[1]Our system ranked first in the HOO-2011 competition (Rozovskaya et al., 2011) and ranked first on some of the metrics and second on the other metrics in the HOO-2012 shared task (Rozovskaya et al., 2012).

errors that are defined differently and do not correspond directly to mistakes on non-finite verbs in Chapter 7.

Seventeen teams that participated in the task built a wide array of approaches that include discriminative classifiers, language models, statistical machine translation systems, and rule-based modules. Many of the systems also made use of linguistic resources such as additional annotated learner corpora, and defined high-level features that take into account syntactic and semantic knowledge. In spite of the fact that the submitted systems incorporated similar resources and employed sophisticated models, the scores varied widely. The system presented in this chapter (henceforth, the *Illinois* system; in the CoNLL-2013 shared task overview paper it is referred to as *UI*) obtained an F1 score of 31.20, while the second team scored only 25.01, and the median result was 8.48 points. These results suggest that there is not enough understanding of what works best and what elements are essential for building a state-of-the-art error correction system.

Contributions

By analyzing the Illinois system, we identify key design principles and show their importance for building a robust error correction model. The following system dimensions are considered: choice of the learning algorithm; choice of training data (native or annotated learner data); model adaptation to typical mistakes; and the use of linguistic knowledge. For each dimension, several implementations are compared, including approaches chosen by other teams that ranked among the top five systems in the competition.

The rest of this chapter is structured as follows. Section 8.2 describes the model design principles that will be examined in this chapter, and Section 8.3 presents the CoNLL-2013 task. Section 8.4 gives an overview of the Illinois system and describes its error-specific components. Section 8.5 presents an experimental analysis that focuses on the four dimensions, and Section 8.6 discusses the results from the perspective of choosing an evaluation metric. Section 8.7 summarizes the chapter.

8.2 Model Dimensions

Based on the analysis of the Illinois system and comparing it with other participating teams, we identify key principles for building a state-of-the-art error correction system:

1. Learning algorithm: Most of the teams, including the Illinois team, built statistical models. We show that the choice of the learning algorithm is very important and affects the performance of the system. The results reinforce the findings in Chapter 5.

2. Adaptation to learner errors: Chapter 6 showed that adaptation, i.e. developing models that utilize knowledge about error patterns of the non-native writers, is extremely important. Several adaptation techniques proposed in Chapter 6 are examined and their impact on the performance of the system is evaluated.

3. Linguistic knowledge: It is essential to use linguistic knowledge when developing error correction modules, e.g. to identify which type of verb errors occurs in a given context, before the appropriate correction module is employed. We describe and evaluate the contribution of these components.

4. Training data: The question of whether to train on learner data (if available) or on native English data is very important. The advantages of both approaches in the context of the shared task and in broader context are discussed.

8.3 Task Description

The CoNLL-2013 shared task (Ng et al., 2013) focuses on the following five common mistakes made by ESL writers: *article/determiner*; *preposition*; *noun number*; *subject-verb agreement*; *verb form*. Errors outside this target group are annotated in the task corpora but are not evaluated.

The CoNLL data set is described in Section 4.2.3, but we briefly review it here. The training data of the shared task is the NUCLE learner corpus (Dahlmeier et al., 2013) (we also refer to it as *learner data* or *shared task training data*). The test data for the task consists of an additional set of 50 student essays from the same first-language background. The training and the test data contain 1.2 million and 29,000 words, respectively.

Table 8.1 illustrates the mistakes considered in the task. With the exception of verb form errors, all other mistakes directly correspond to the phenomena considered in the preceding chapters. Errors marked as *verb form* include multiple grammatical phenomena that may characterize verbs, with the exception of agreement and tense error that are marked separately, and are not restricted to errors on non-finite verbs discussed in Chapter 7. The number of errors in the CoNLL data

Error type	**Examples**
Article	"It is also important to create **a/∅* better material that can support **the/∅* buildings despite any natural disaster like earthquakes."
Preposition	"As the number of people grows, the need **of/for* habitable environment is unquestionably essential."
Noun number	"Some countries are having difficulties in managing a place to live for their **citizen/citizens* as they tend to get overpopulated."
Verb agreement	"Therefore, the equipments of biometric identification **tend/tends* to be inexpensive."
Verb form	"...countries with a lot of deserts can terraform their desert to increase their habitable land and **using/use* irrigation." "It was not **surprised/surprising* to observe an increasing need for a convenient and cost effective platform." "With surveillance technology, terrorists could be more easily **identify/identified*..."

Table 8.1: **Sample errors considered in the CoNLL-2013 shared task.** Note that only the errors exemplifying the relevant phenomena are marked in the table; the sentences may contain other mistakes. Errors marked as *verb form* include multiple grammatical phenomena that may characterize verbs and are not restricted to errors on non-finite verbs discussed in Chapter 7.

	Classifier				
	Article	**Preposition**	**Noun number**	**Agreement**	**Form**
Train	254.0K	103.0K	240.0K	75.0K	175.0K
Test	6.0K	2.5K	2.6K	2.4K	4.8K

Table 8.2: **Number of candidate words by classifier type in training and test data.**

set and the error rates are listed in Table 4.3. It should be noted that the test data contains a much larger proportion of annotated mistakes. For example, while only 2.4% of noun phrases in the training data have determiner errors, in the test data 10% of noun phrases have mistakes.

8.4 The Illinois System

The Illinois system consists of five machine-learning models, each specializing in correcting one of the errors described above. Table 8.2 shows the total number of candidates for each classifier.

Pre-processing All components take as input the corpus documents pre-processed with a part-of-speech tagger[2] and shallow parser[3] (Punyakanok and Roth, 2001). Note that the shared task data contains comparable pre-processing information but we chose to run our own tools.

Learning In the Illinois submission, modules are trained somewhat differently and on different data – for some, the learner data is used and for others, native data is used.

[2]http://cogcomp.cs.illinois.edu/page/software_view/POS
[3]http://cogcomp.cs.illinois.edu/page/software_view/Chunker

Post-processing The Illinois submission includes a post-processing step, where corrections that always result in a false positive on the training data are ignored. In order to focus on the key characteristics of the system, here the post-processing step is not included; removing post-processing improves the result slightly from 31.20 to 31.43.

Below we describe the five error modules of the Illinois system.

8.4.1 Determiner Errors

Determiner errors in the CoNLL corpus are defined in the same way as described in Section 5.2.2 and involve all kinds of determiners: articles, and possessive and demonstrative pronouns. Most of the determiner errors in the data set are article omissions and article insertions. This is consistent with the error patterns observed in other learner corpora (see Chapter 4).

The Illinois system addresses only article errors. Candidates are identified using the method described in Section 5.2.2. Following the results on the comparison of the learning algorithms in Chapter 5, the article classifier is trained in a discriminative fashion, making use of the AP algorithm (see Section 5.4) and is trained on the training data of the shared task with rich features. The features, in addition to the surface form of the context around the target article, also encode POS and chunk properties of the surrounding context and are described in Section 5.2.3 (see Table 5.4). The contribution of the features is evaluated in Section 8.5.3.

Because the article system is trained on learner data with naturally-occurring errors, the *source* article can also be used as a feature. As discussed in Chapter 6, the source word is an important piece of information, and using it as a feature improves the performance of an error correction system. However, since learner errors are sparse, the *source* feature encourages the model to abstain from flagging a mistake, which results in low recall. To address the low recall problem, the article system implements an adaptation technique called "error inflation" presented in Section 6.2.2. The contribution of the error inflation method is evaluated in Section 8.5.2.

8.4.2 Preposition Errors

In contrast to determiners, for learners of many first-language backgrounds, the majority of the preposition errors are replacements (Section 5.2.3). However, learner errors heavily depend on the first-language background, and even though most of the preposition mistakes in the NUCLE corpus (62%) are replacements, spurious prepositions are also quite common and occur more frequently than in other learner corpora: in the NUCLE corpus, 29% of all preposition mistakes are spurious errors, while both in the Illinois corpus and the FCE corpus spurious errors account for 18% of all preposition mistakes (Table 5.1).

The preposition model is a NB classifier trained on the Google corpus. The preposition model, as well as all other models trained on the Google corpus, uses word n-gram features in the 4-word window around the target word. Since we wish to use the 4-word window context, we cannot train a discriminative model on the Google corpus (see Section 5.5). The model is adapted to likely preposition confusions using the NB adaptation method (see Section 8.5.2). The original Illinois preposition model targets replacement errors of the twelve most common English prepositions.[4] To improve coverage and to account for preposition errors that commonly occur in the CoNLL corpus, in this chapter we augment the preposition model and also identify spurious prepositions. Section 5.4.3 describes a procedure developed for training NB on the Google corpus for errors that involve deletions and insertions.

8.4.3 Verb Errors

The CoNLL-2013 shared task includes verb errors that are marked as agreement and form. Form errors are not limited to errors on verbs in non-finite form but cover all types of mistakes on verbs, with the exception of agreement and tense errors that are marked separately (see Table 8.1).[5]

The verb modules are implemented in the spirit of the linguistically-motivated approach based on verb finiteness proposed in Chapter 7. While Chapter 7 focuses on a high-level description of the approach to verb error correction, here we give a more detailed description that also includes the modifications that are due to how verb errors are defined in the CoNLL data set. The verb modules consist of the following components: (1) candidate identification; (2) determining the set of instances that should be handled by agreement and form modules based on verb type (verb

[4]The shared task includes mistakes that cover 36 prepositions but we found that once the confusion set becomes too large the model performance drops, and thus chose the twelve most common prepositions. Also note that the preposition models in Chapters 5 and 6 include replacements of the top ten prepositions. The CoNLL confusion set is expanded to improve the coverage for the CoNLL corpus. The confusion set for prepositions is {*on*, *from*, *for*, *of*, *about*, *to*, *at*, *in*, *with*, *by*, *into*, *during*}.

[5]The training data also marks some tense errors as a subset of form mistakes, while the test data does not.

“Hence, the environmental factors also ***contributes/contribute** to various difficulties, ***included/including** problems in nuclear technology.”	
Error	**Confusion set**
Agreement	{INF=contribute, S=contributes}
Form	{INF=include, ED=included, ING=including, S=includes}

Table 8.3: **Confusion sets for agreement and form classifiers.** Note that for irregular verbs, the first candidate in the confusion set for *Verb form* is the past participle.

finiteness); (3) training two classifiers – for agreement and form error correction.

Generating Morphological Forms for Verbs

In contrast to the article and preposition systems, where the *confusion sets* consist of a closed list of words, here the confusion set depends on the target word and includes its morphological variants. Table 8.3 illustrates confusion sets for verbs. To generate morphological variants, we make use of the tool called *verbMorph* that was developed as part of the verb error correction system introduced in Chapter 7 and performs morphological analysis on verbs (see Section 7.2.3).

Candidate Identification

This stage selects the set of words that are presented as input to the classifier. As discussed in Chapter 7, this is a crucial step for a system that corrects errors on open-class words. The candidate identification procedure follows the method proposed in Section 7.2.3. Note that while in Chapter 7 verb instances correspond to verb phrases, here each verb instance corresponds to one token. If a verb phrase contains multiple verb tokens (e.g., *has gone*), each of the tokens may be extracted as a separate candidate.

Using Verb Finiteness to Correct Verb Errors

The Illinois system includes two verb classifiers – agreement and form – that are inspired by the approach to verb error correction based on the linguistic notion of verb finiteness introduced in Section 7.2. The idea is to process verbs that fulfill different grammatical functions and thus are marked for different grammatical properties separately (see Section 8.5.3 for the contribution of this method).

Verb Correction Modules The agreement and form modules are implemented as a binary and 4-class classifiers, respectively.[6] Note that while the form classifier does not specifically target tense errors, it considers confusions, such as *contributed* and *contribute*, that on the surface might look like tense errors but could also be mistakes related to infinitives (e.g., "He may **contributed/contribute*."), so we specifically do not exclude these. Importantly, these do not represent all confusions related to tense errors, which are not addressed here, as they are not part of the task.

The Illinois system trains both classifiers on the Google corpus but analysis in Section 8.5.4 reveals that it is possible to improve over this model by training instead on learner data with more sophisticated syntax-based features developed in Chapter 7 and presented in Table 7.7.

8.4.4 Noun Errors

The Illinois noun number module is a NB classifier trained on the Google corpus with word n-gram features. Similar to verbs, *candidate identification* is an important step in the noun classifier. Our approach relies on POS information and heuristics that address the problem of pre-processing errors. It is described and contrasted with methods used by competing systems in Section 7.3.

8.5 System Analysis

In this section, we analyze the Illinois model, compare its components to alternative configurations implemented by other teams that ranked among the top five in the shared task, and present additional experiments that provide insight into the key principles that should be considered when developing a state-of-the-art system. In the experiments below, we focus on the following four dimensions identified in Section 8.2: choice of the learning algorithm (Section 8.5.1); adaptation to learner errors (Section 8.5.2); use of linguistic knowledge (Section 8.5.3); and choice of the training data (Section 8.5.4).

The evaluation metric used in the shared task is F1, therefore all of the modules are optimized for F1. Performance on both the training and the test set is reported. The results on the training set reflect performance of 5-fold cross-validation. The performance reported on the test data reflects the system that makes use of the entire NUCLE training corpus. We note that the performance

[6] For the verb *to be*, the agreement distinction is ternary – {*am*,*is*,*are*} – but the form *am* is rarely misused, and we ignore it.

on the test data is significantly higher than on the training data. This behavior is exhibited by all systems in the CoNLL-2013 shared task. It is due to the significantly lower error rates observed in the training data. Recall that the error rates for all of the phenomena in the test data are several times higher than in the training set (see Table 4.3 and discussion in Section 4.2.4).

The shared task made available two sets of test annotations: originally annotated data and with additional revisions based on the input from the participants. We show results *before* revisions, as the revised data may be biased (Ng et al., 2013).

Results by error type are generated based on the output of individual modules. Note that these are not directly comparable to error-specific results in the CoNLL overview paper, since the latter are approximated based on POS (Ng et al., 2013).

The complete system includes the union of corrections made by each of these modules, with the corrections generated by the individual models applied in order. Since some of the candidates for individual classifiers overlap (e.g., some verbs are processed by both the agreement and form modules), ordering overlapping candidates might potentially affect the final output, when both modules correctly identify an error but propose different corrections, but this does not happen in practice.

8.5.1 Dimension 1: Learning Algorithm

In Chapter 5, it was shown that the choice of a machine learning algorithm is extremely important and affects the performance of an error correction system. Following these results, the modules trained on the NUCLE corpus use the discriminative approach AP. Most of the other teams that train on the NUCLE corpus also use a discriminative method.

Our goal here is to compare learning models trained on the Google corpus. This is one of the largest native data sets available, and it clearly benefits the systems. Several participating teams make use of the Google corpus, either by training a language model (Xing et al., 2013) or a count-based model (Kao et al., 2013).[7]

In Chapter 5, we study several algorithms trained on the Google corpus and observe that NB performs better than other models, including LMs and count-based methods. Recall that training a

[7]For count-based methods, see Bergsma et al. (2009).

discriminative model on the Google data would limit the surrounding context features to a two-word window).

Our comparison of LM and NB models shows the impact of the algorithm choice in the context of the shared task (we do not implement count-based methods since they fall behind even the language modeling approach (see Chapter 5). Both NB and LM models use word n-grams spanning the target word in the 4-word window. As in Chapter 5, we train LMs with SRILM (Stolcke, 2002) using Jelinek-Mercer linear interpolation as a smoothing method (Chen and Goodman, 1996). Interpolation weights are optimized on a held-out set of for the NUCLE training data. Table 8.4 compares LM and NB trained for several error types. On the training data, NB outperforms LM on three error types, sometimes by a large margin (article errors), and on the test data – on all errors.

Error	**Model**	**F1**	
		Train	**Test**
Article	LM	11.46	21.11
	NB	**18.28**	**32.45**
Preposition	LM	**13.49**	12.09
	NB	09.03	**14.04**
Noun number	LM	22.37	40.72
	NB	**23.06**	**42.60**
Agreement	LM	15.91	20.65
	NB	**16.72**	**26.46**
Form	LM	**12.44**	13.40
	NB	11.93	**14.50**

Table 8.4: **Comparison of the learning models.** All models are trained on the Google corpus and use the same word n-gram features.

8.5.2 Dimension 2: Adaptation to Learner Errors

In the previous section, models were trained on native English data. These models have no notion of the error patterns of the learners. The second question that we address is model adaptation to learner errors. The adaptation idea is based on the observation that learners make mistakes in a systematic manner and is studied in Chapter 6. Superfluous preposition usage and noun number confusions, for instance, are mistakes that are typical for the writers of the CoNLL corpus (see Section 8.4.2 and Table 4.4). In the shared task, both the training and the test data are produced by learners from the same linguistic background, and the adaptation techniques allow us to expose the models to the types of errors common for these writers.

In Chapter 6, we study different adaptation methods that depend on the type of training data (learner or native) and the choice of the learning model. The key application of adaptation is for models trained on native English data. Adaptation is especially important in this case, because the learned models do not know anything about the errors learners make. Adaptation techniques allow the models trained on native data to use the source word as a feature and to propose a correction based on what the author originally wrote.

Adapting the Naïve Bayes Classifier

The first adaptation approach that we evaluate is the *priors method* (Section 6.2.3), which is a special adaptation technique for a NB model trained on native English data, and is based on changing the distribution of priors over the correction candidates. When a model is trained on native data, candidate priors correspond to the relative frequencies of the candidates in the native corpus and do not provide any information on the real distribution of mistakes and the dependence of the correction on the word used by the author.

In the NB adaptation, candidate priors are changed using the error confusion matrix based on the learner data that specifies how likely each confusion pair is to occur. As an illustration, Table 8.5 shows the confusion matrix that is used to replace the native priors for verb form errors, computed on the training data of the shared task. The new candidate priors are dependent on the author's original verb form used: the probability of candidate *INF* when the source form is *ED*, is more than twice higher than when the source form is *S*.

Table 8.6 compares the performance of NB and NB-adapted models. We found that adaptation always helps on the training data (except noun errors), but on the test data it only improves performance on article and verb form errors. We conjecture that this is because the error rates of the test data are quite different from those in the training data (see Table 4.3), and the adaptation methods rely on similar error distributions. Indeed, when priors are estimated on the test data, the performance improves significantly. For example, the preposition module attains an F1 of 18.05, which is a 4-point improvement over the non-adapted NB model.

Source	**Candidates**			
	ED	INF	ING	S
ED	0.99675	0.00192	0.00103	0.00030
INF	0.00177	0.99630	0.00168	0.00025
ING	0.00124	0.00447	0.99407	0.00022
S	0.00054	0.00544	0.00132	0.99269

Table 8.5: **Priors confusion matrix for verb form errors used in NB adaptation.** Each entry shows $Prob(candidate|source)$, where *source* denotes the verb form chosen by the author, and each verb is mapped to its grammatical form. Based on the CoNLL training data.

Error	**Model**	**F1**	
		Train	**Test**
Article	NB	18.28	32.45
	NB-adapted	**19.18**	**34.49**
Preposition	NB	09.03	**14.04**
	NB-adapted	**10.94**	12.14
Noun number	NB	**23.06**	**42.60**
	NB-adapted	22.89	42.31
Agreement	NB	16.72	**26.46**
	NB-adapted	**17.62**	23.46
Form	NB	11.93	14.50
	NB-adapted	**14.63**	**18.35**

Table 8.6: **Model adaptation for NB.** All models are trained on the Google corpus and use the same word n-gram features.

Adapting the Averaged Perceptron

The AP algorithm is used in models trained on the shared task data. AP models are online models and do not use priors on the set of candidates. In order to reflect our estimate of the error distribution, the AP algorithm is adapted differently, by introducing into the native data artificial errors, in a rate that reflects the errors made by the ESL writers (Section 6.2.2).

The *error inflation* method (Section 6.2.2) takes the idea of artificial mistakes further and can be used to improve the recall of discriminative models that are trained either on native data or learner data with naturally-occurring mistakes. Here, the error inflation method is applied to models trained on learner data. We note that although these models already have knowledge of typical confusions, we consider the error inflation approach part of the adaptation paradigm since it does not simply increase the error rates in the training data but in doing so it respects the error distributions, i.e. the likely confusions.

Table 8.7 shows the results of adapting the AP classifier using the error inflation method. Noun and form results are omitted, as these errors are better addressed using the Google corpus (Section 8.5.4).

Inflation Versus Sampling To further analyze the contribution of error inflation, we compare it against *sampling*, an approach used by other teams, (e.g., Xiang et al., 2013), that improves recall by removing correct examples in training. The article model by Xiang et al. (2013) is similar to the Illinois model but scored 3 F1 points below. Table 8.8 shows that the sampling approach falls behind the inflation method, since it significantly reduces the training size to achieve similar error rates. The proportion of errors in training in each row is identical: sampling achieves the error rates by removing correct examples, whereas the inflation method converts some positive examples to artificial mistakes. Recall that the *inflation constant* specifies the proportion of the correct instances remaining in training; smaller values correspond to higher error rates; the sampling approach, correspondingly, removes more positive examples. Note that the gap between error inflation and sampling grows as the error rates increase, since the sampling approach shrinks the training set.

To conclude discussion on adaptation, we stress that the adaptation approaches presented here make use of all the training data available. In practice, however, a lot less annotated data will be

Error	Model	F1	
		Train	Test
Article	AP (natural errors)	09.99	07.06
	AP (infl. const. 0.9)	**16.90**	**24.61**
Preposition	AP (natural errors)	0.25	0.0
	AP (infl. const. 0.7)	**11.17**	**07.37**
Agreement	AP (natural errors)	0.39	0.0
	AP (infl. const. 0.8)	**13.46**	**17.06**

Table 8.7: **Model adaptation for AP using error inflation.** All models are trained on the NUCLE corpus and use the same word n-gram features and the source word as a feature. *Inflation constant* shows how many correct instances remain in training (e.g., 0.9 indicates that 90% of the correct instances remain in the data, while 10% are converted to artificial mistakes.)

	F1	
Infl. constant	**Sampling**	**Inflation**
0.90	23.22	24.61
0.85	27.75	29.29
0.80	30.04	33.47
0.70	33.02	35.52
0.60	32.78	35.03

Table 8.8: **Comparison of the inflation and sampling methods for boosting recall on article errors.** Results on the test data. In both the sampling and the adaptation methods, the proportion of errors in training in each row is identical.

available, but these methods only require very little annotation for error statistics to be effective. For instance, in the HOO-2011 shared task, where the Illinois system also placed first, article and preposition models are adapted using only 4860 article instances and less than 2500 prepositions (Rozovskaya et al., 2011).

8.5.3 Dimension 3: Linguistic Knowledge

The use of linguistic knowledge is important in several components of the error correction system: candidate identification, features engineering, and special techniques for correcting verb errors.

Candidate Identification

In Section 7.3, we describe several candidate identification methods for noun number errors based on the approaches followed by other teams and contrast these with the approach used in the Illinois system. The Illinois candidate identification method addresses the problem of the pre-processing

Error	**Features**	**F1**	
		Train	**Test**
Article	n-gram	16.90	24.61
	n-gram+POS+chunk	**19.24**	**33.50**
Agreement	n-gram	13.46	17.06
	n-gram+POS	17.79	24.14
	n-gram+POS+syntax	**22.22**	**27.93**

Table 8.9: **Feature evaluation for article and verb agreement models.** All models are trained on the NUCLE corpus and are optimized using the error inflation method.

mistakes. It is shown that developing strategies to address pre-processing errors in the candidate selection method for mistakes on open-class words is important.

Features

Because of the format of the Google corpus, models trained on that data can make use of n-gram features only but for the NUCLE corpus we have several layers of morphological and syntactic annotation. Models for two error types – articles and agreement – especially benefit from training with features that use morphology and syntax (Table 8.9). The article features are defined based on the POS and shallow parser output and are presented in Table 5.4. The agreement features are listed in Table 7.7.

Using Verb Finiteness to Correct Verb Errors

As shown in Table 8.3 (Section 8.4.3), the surface realizations that correspond to the two agreement candidates are a subset of the four possible surface realizations of the form classifier. One natural approach, thus, is to train one classifier to predict the correct surface form of the verb, without considering the grammatical function of the verb in the sentence. However, as discussed in Section 7.2, the same surface realization may correspond to multiple grammatical properties.

The verb models of the Illinois system address agreement and form errors separately based on the *verb finiteness* approach proposed in Section 7.2. Using the verb-finiteness distinction, each verb is directed to the appropriate verb classifier. The candidates for the agreement module are the finite surface forms of the be-verbs (*is*, *are*, *was*, and *were*) and other finite verbs that have explicit subjects and require agreement markers. The form candidates are all other verbs. Recall that the verb form errors in the CoNLL data set include various types of grammatical verb misuse.

Training method	**F1**	
	Train	**Test**
Combined classifier	15.37	16.43
Type-based training (I)	16.76	18.59
Type-based training (II)	**17.10**	**21.08**

Table 8.10: **Improvement due to separate training based on verb finiteness.** All models are trained using the AP algorithm. When training separately we can take advantage of the fact that different errors benefit from different types of features. Type-based training (I) uses exactly the same POS features for agreement and form. Type-based training (II) optimizes POS features for each classifier individually.

Table 8.10 compares the performance on verb errors (agreement and form) for two approaches: when all verbs are handled together in a single classifier; and when verbs are processed separately based on verb type. Both of the approaches are trained on the NUCLE data with the AP learning algorithm. Note that when training separately we can take advantage of the fact that different errors benefit from different types of features, which is not possible, when one model is trained: in Table 8.10, classifier (I) uses exactly the same POS features for agreement and form; classifier (II) optimizes the set features for each classifier individually. The results in Table 8.10 are on the AP models but similar improvements due to separate training are observed for NB models trained on the Google data. Interestingly, another team (Kao et al., 2013) also corrects all verb errors using a model trained on the Google data but they handle all these errors together and score 8 F1 points below the Illinois verb module (they also train a count-based model, while the Illinois system uses NB classifiers).

8.5.4 Dimension 4: Training Data

The training data of the shared task is a large annotated corpus of learner essays that is of the same genre as the test data and produced by learners of the same linguistic background. In many scenarios, there is not going to be enough annotated learner data available for training the models (for example, the HOO-2011 shared task released a small annotated development corpus, and the participants had to train their systems on native data). However, the NUCLE corpus is relatively large (1.2 million words). The learner training data also possesses other desirable characteristics: it has multiple levels of annotations, which allows us to design richer features that reflect similar distributions occurring in the test data; finally, we are not restricted in the choice of a learning algorithm. For the above reasons, training models on this corpus may seem a natural choice. Indeed, many participating teams follow this approach and train the entire system on this data set (e.g., Xiang et al., 2013). We also show here that a model trained on the NUCLE data is hard to beat in many cases, even though the alternative model trained on native English data is orders of magnitude larger than the size of the NUCLE data. But we also show that some errors especially benefit from training on a larger corpus of native data.

Based on the analysis of the other model dimensions, in Table 8.11, we compare models trained on NUCLE and Google data in their best configurations on the training data. We find that three error types are especially affected by the choice of the training data: noun and form mistakes benefit from training on native data, while verb agreement errors, on the other hand, benefit from a model trained on the learner data with richer features. Specifically, when verb agreement models are trained on learner data with n-gram features, they do not perform as well as the models trained on the Google corpus. However, when using the learner data, we can add features that use POS and syntactic knowledge. Adding these features is extremely helpful.

For noun number mistakes, on the other hand, there is an important boost in performance when training on the native data as opposed to the learner data. This is because noun number usage strongly depends on the surface form of the noun. For example, certain nouns in English tend to be used exclusively in singular or plural form. Thus, a lot more data compared to other types of mistakes is needed to learn the model parameters. For verb form errors we observe a smaller but also a significant advantage to training on a lot of data; unlike agreement mistakes that are more structural and thus benefit from using more expressive features, form errors, similar to noun errors, benefit from training on more data.

To summarize, choice of the training data is an important consideration for building a robust error correction system. Learner data possesses several desirable characteristics but it is expensive and in many scenarios there will not be enough of it to train a robust model. Moreover, even when there is a fair amount of learner data, some errors may still benefit from a model that is trained on a native corpus whose size is orders of magnitude larger. Finally, adaptation methods described in Chapter 6 can improve models trained on native data with just a small amount of annotation.

Error	**Model**	**Features**	**F1**	
			Train	**Test**
Article	NB-adapted (Google)	n-gram	19.18	**34.49**
	AP-adapted with inflation (NUCLE)	+POS+chunk	**19.64**	33.50
Preposition	LM (Google)	n-gram	**13.49**	**12.09**
	AP-adapted with inflation (NUCLE)	n-gram	11.17	10.26
Noun number	NB (Goolge)	n-gram	**23.06**	**42.60**
	AP-adapted with inflation (NUCLE)	+POS	11.03	19.22
Agreement	NB-adapted (Google)	n-gram	17.62	23.46
	AP-adapted with inflation (NUCLE)	+POS+syntax	**22.22**	**27.93**
Form	NB-adapted (Google)	n-gram	**14.63**	**18.35**
	AP-adapted with inflation (NUCLE)	+POS	11.99	12.32

Table 8.11: **Choice of training data: learner versus native.** Best configuration of features and adaptation for each data type is chosen.

8.6 Discussion

In the previous section, we analyzed the Illinois system along four dimensions. In Table 8.12, we show the overall results and the performance of each of the components of the Illinois system and the improved results that are based on the best modules for each error type. The best modules are selected according to the performance on the training data (Table 8.11). The Illinois system at CoNLL-2013 uses post-processing and obtains an F1 score of 31.20. Here, the Illinois system performance is reported without the post-processing step for a fair comparison. The overall performance on the test data is a 0.32% absolute improvement over the Illinois system without the post-processing step, and a 0.55% improvement over the original system that includes post-processing. The improvement is due to enhancing the preposition module to address spurious mistakes; adding adaptation to the form classifier; and replacing the agreement module with a model trained on learner data. The improved modules are emphasized in the table.

Note that in the CoNLL-2013 competition, systems were compared on F1 performance and, consequently, this is the metric we optimize here. The F1 measure evaluates systems only at one Precision/Recall point. Moreover, this evaluation metric does not ensure that the quality of the text improves as a result of running the system (see Section 4.5), while in practical error correction systems, recall is minimized to guarantee that the overall quality of the corrected text does not go down.

To examine the behavior of the system at multiple recall points and to determine whether

Error	Illinois				Illinois-improved			
	Model	**P**	**R**	**F1**	**Model**	**P**	**R**	**F1**
Art.	AP-adapted (NUCLE)	51.04	24.93	33.50	AP-adapted (NUCLE)	51.04	24.93	33.50
Prep.	NB-adapted (Google)	23.64	04.18	07.10	LM (Google)	27.91	07.72	**12.09**
Noun	NB (Google)	48.24	38.13	42.60	NB (Google)	48.24	38.13	42.60
Agr.	NB (Google)	44.23	18.55	26.14	AP-adapted (NUCLE)	31.63	29.00	**27.93**
Form	NB (Google)	13.57	15.57	14.50	NB-adapted(Google)	20.83	16.39	**18.35**
All		45.15	24.10	31.43		45.61	24.35	**31.75**

Table 8.12: **Results on Test of the Illinois system (without post-processing) and the Illinois-improved system.** For the improved system, the best modules for each error type are selected according to the results on the training data obtained in Section 8.5 (see Table 8.11). The Illinois system at CoNLL-2013 uses post-processing and obtains an F1 score of 31.20. Here, the Illinois system performance is reported without the post-processing step for a fair comparison.

the quality of the text improves as a result of running the system, in Figure 8.1, we show Precision/Recall curves for individual classifiers that are part of the Illinois-improved system in Table 8.12. We observe that performance varies widely by error type. The easiest are noun and article usage errors: for nouns, we can do pretty well at the recall point 20% (with the corresponding precision of over 60%); for articles, the precision is also around 50% at the recall value of 20%. For agreement errors, we can get a precision of 55% with a very high threshold (identifying only 5% of mistakes). Finally, on two mistakes – preposition and verb form – the system never achieves a precision over 50% and thus cannot improve the quality of the text for these errors.

It is also interesting to see by how much the quality of the text can be improved if the system is tuned so that precision is sufficiently high. Using the definition of *accuracy* (Section 4.5), the noun module, for instance, at the recall point 20% achieves a precision of 63.93%. This translates into an accuracy of 94.46%, while the baseline on the noun errors (i.e. the accuracy of the data before running the system) is 94.0%. This means that the system improves the quality of the data. Importantly, once precision drops below 50%, the system introduces more mistakes than it identifies.

On the other hand, it should be noted that the results above in fact underestimate the performance of the system. This is because the original annotations do not provide alternative and optional corrections. The *after* revisions set adds error annotations missed in the original stage. Annotation errors make a big difference in this task, where the error rates are so low. Both the recall and especially precision of the systems go up when evaluated on the after revisions data. The

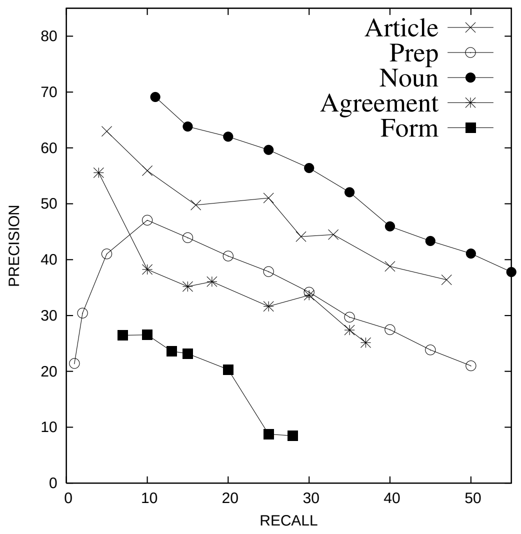


Figure 8.1: **Precision/Recall curves by error type.**

precision of the original Illinois system goes up from 46.45 to 62.19 on revised annotations. While the revised annotations may be biased (Ng et al., 2013), they do demonstrate the importance of providing alternative and optional corrections.

8.7 Summary

In this chapter, we build a complete error correction model that implements the techniques proposed in this thesis. This is a state-of-the-art error correction model – referred to as Illinois – that placed first in the CoNLL-2013 shared task. Through an analysis of the Illinois system and by comparing its components to implementations by other teams, we identify key design principles for building a state-of-the-art error correction system and show why it is doing so much better than the others. The key dimensions are closely aligned with the methods proposed in this thesis and concern the choice of a learning algorithm, adaptation to learner mistakes, use of linguistic knowledge, and choice of the training data. It is shown that the decisions in each case depend both on the type of mistake being targeted and the specific setting, e.g. how much annotated learner data is available. As a result of the analysis, we also improve over the Illinois model.

Chapter 9

Joint Learning and Inference for Error Correction

The previous chapters focused on developing classifiers targeting a specific type of mistake. The state-of-the-art Illinois system presented in Chapter 8 is based on a collection of these independently-trained models. Such models ignore linguistic interactions at the sentence level and thus do poorly on mistakes that involve grammatical dependencies among several words. In this chapter, we go beyond error correction that operates at the word level: we identify linguistic structures with interacting grammatical properties and propose to address such dependencies via joint inference and joint learning.

9.1 Introduction

The Illinois system, which is a state-of-the-art system presented in the previous chapter, consists of a collection of classifiers where each one specializes in correcting a specific error type. The final model includes the union of the corrections proposed by the individual classifiers. This standard approach considers each word in a sentence independently and thus assumes that there are no interactions between errors and between grammatical phenomena. For this reason, such models tend to make predictions that are inconsistent with the grammatical rules of the languages and, moreover, fail to identify errors in contexts that require multiple changes to be done simultaneously.

Even though it is quite clear that grammatical errors interact, for various conceptual and technical reasons, this issue has not been addressed in a significant way in the literature. We believe that the reasons for this are three-fold: (1) Lack of data: until very recently we did not have data that jointly annotates sufficiently many errors of interacting phenomena (see Section 9.2). (2) Conceptual: correcting errors in interacting linguistic phenomena requires that one identifies those phenomena and, more importantly, can recognize reliably the interacting components (e.g., given a verb, identify the subject to allow agreement to be enforced). The perception has been that this cannot be done reliably, but see Section 9.4. (3) Technical issues: the NLP community has only recently begun to better understand joint learning and inference and apply it to various phenomena (Roth and Yih, 2004a; Punyakanok et al., 2008; Martins et al., 2011; Clarke and Lapata, 2007; Sutton and McCallum, 2007) .

We show that it is possible to identify interactions well enough to facilitate a joint approach and, consequently, that joint methods correct incoherent predictions that independently-trained classifiers tend to produce. Furthermore, because the joint learning model considers interacting phenomena during training, it is able to identify mistakes that require making multiple changes simultaneously and that standard approaches miss. Overall, our joint model significantly outperforms the Illinois system, increasing the F1 score by 2 and 4 points in different evaluation settings.

Contributions

- We identify two pairs of interacting phenomena, *subject-verb* and *article-NPhead* agreements; we show how to reliably identify these pairs in noisy ESL data, thereby facilitating the joint correction of these phenomena.

- We propose two joint approaches: (1) a *joint inference* approach implemented on top of individually learned models using an Integer Linear Programming formulation (Roth and Yih, 2004a), and (2) a model that *jointly learns* each pair of these phenomena. We show that each of these methods has its advantages, and that both solve the two challenges outlined above: the joint models exclude inconsistent predictions that violate linguistic constraints. The joint learning model exhibits superior performance, as it is also able to overcome the problem of the noisy context encountered by the individual models and to identify errors in contexts where multiple changes need to be applied at the same time.

The rest of this chapter is structured as follows. Section 9.2 motivates the choice of the data set for this study, and Section 9.3 briefly presents the baseline system. Section 9.4 describes the interacting structures. Section 9.5 presents the joint algorithms. Experiments demonstrating the utility of the joint approaches are presented in Section 9.6. Section 9.7 discusses the results and presents error analysis. Section 9.8 concludes the chapter.

9.2 Choice of the Learner Data Set

To illustrate the utility of jointly addressing interacting grammatical phenomena, we consider the corpus of the CoNLL-2013 shared task (see Section 4.2.3); this data set is particularly well-suited for addressing interactions between grammatical phenomena. The shared task focuses on the following five common mistakes made by ESL writers: *article*, *preposition*, *noun number*, *subject-verb agreement*, and *verb form* (see Table 8.1 for examples of mistakes). Two pairs of grammatical interactions that are relevant for the types of errors included in the task are addressed – *article-NPhead* and *subject-verb*.

We note that the CoNLL-2013 data set is the first annotated collection that makes a study like ours feasible. The presence of a common test set that contains a good number of interacting errors – article, noun, and verb agreement mistakes – makes the data set well-suited for studying which approach works best for addressing interacting phenomena. The HOO 2012 competition (Dale et al., 2012) only addresses article and preposition mistakes, while the HOO-2011 shared task collection (Dale and Kilgarriff, 2011) contains a very small number of noun and agreement errors (41 and 11 in test, respectively). Indeed, in parallel to the work presented here, Wu and Ng (2013) attempted the ILP-based approach of Roth and Yih (2004b) on the HOO-2011 data set but did not show any improvement because of the small number of interacting phenomena and because they applied constraints in an indiscriminate manner. In contrast, we show how to identify the interacting structures' components in a reliable way, and this plays a key role in the joint modeling improvements.

9.3 The Baseline System

The joint approaches are applied to a system that consists of independently-trained classifiers. We take as the baseline the original Illinois system built in Chapter 8, as it is implemented in the CoNLL-2013 shared task.[1] As discussed in Chapter 8, this is a state-of-the-art system that ranked first in the CoNLL-2013 shared task. The Illinois system implements five machine-learning independently-trained classifiers each targeting one specific type of mistake. For system descrip-

[1]The Illinois system in the shared task also includes a post-processing step (Section 8.4).

Example	**Predictions made by the Illinois system**
"They believe that *such situation* must be avoided."	such situation $\rightarrow$ such a situations
"Nevertheless, electric *cars is* still regarded as a great trial innovation."	cars is $\rightarrow$ car are
"Every *students have* appointments with the head of the department."	*No change*

Table 9.1: **Examples of predictions of the Illinois system that combines independently-trained models.**

tion, we refer the reader to Chapter 8.

In Chapter 8, it was also found that the Illinois system can be further improved by replacing several of its components with alternative models. In particular, the agreement NB classifier trained on the Google corpus is replaced with a classifier trained on the learner data that makes use of syntactic features. The key experiments in this chapter are applied to the original Illinois system. One reason for this is to make the comparison fair, since the *joint learning* component is trained on the Google corpus, similar to the corresponding classifiers of the Illinois model, while the improved system in Chapter 8 replaces the agreement classifier with a model trained on learner data. For completeness, we also show *joint inference* results applied to the Illinois-improved model.

9.4 Interacting Mistakes

The approach of addressing each type of mistake individually is problematic when multiple phenomena interact. Consider the examples in Table 9.1 and the predictions made by the Illinois system. In the first and second sentences, there are two possible ways to correct the structures "such situation" and "cars is". In the former, either the article or the noun number should be changed; in the latter, either the noun number or the verb agreement marker.[2] In these examples, each of the independently-trained classifiers identifies the problem because each system makes a decision using the second error as part of its contextual cues, and thus the individual systems produce *inconsistent predictions.*

The second type of interaction concerns cases that require multiple simultaneous corrections be made; the last example in Table 9.1 requires making changes both to the verb and the subject.

[2]Both of these solutions will result in grammatical output and the specific choice between the two depends on the wider essay context.

Since each of the independent classifiers (for nouns and for verb agreement) takes into account the other word as part of its features, they both infer that the verb number is correct and that the grammatical subject *student* should be plural. This results in *missed errors.*

We refer to the words whose grammatical properties interact as *structures.* The independently-trained classifiers tend to fail to provide valid corrections in contexts where it is important to consider both words of the structure together.

9.4.1 Structures for Joint Modeling

We address two linguistic structures that are relevant for the grammatical phenomena considered: *article-NPhead* and *subject-verb.* In the *article-NPhead* structures, the interaction is between the head of the noun phrase (NP) and the article that refers to the NP (first example in Table 9.1). In particular, the model should take into account that the article *a* cannot be combined with a noun in plural form. For subject-verb agreement, the subject and the verb should agree in number.

We now need to identify all pairs of candidates that form the relevant structures. *Article-NPhead* structures are pairs of words, such that the first word is a candidate of type article, while the second word is a noun candidate. Given an article candidate, the head of its NP is determined using the POS information (this information is obtained from the article feature vector because the NP head is a feature used by the article system).[3] *Subject-verb* structures are pairs of noun-agreement candidates. Given a verb, its subject is identified with a dependency parser (Marneffe et al., 2006).

To evaluate the accuracy of subject and NP head predictions, a random sample of 500 structures of each type from the training data was examined by a human annotator with formal training in Linguistics. The human annotations were then compared against the automatic predictions. The results of the evaluation for subject-verb and article-NPhead structures are shown in Tables 9.2 and 9.3, respectively. Although the overall accuracy is above 90% for both structures, the accuracy varies by the distance between the structure components and drops significantly as the distance increases. For article-NPhead structures, *distance* indicates the position of the NP head with respect to the article, e.g. distance of 1 means that the head immediately follows the article. For

[3]Some heads are not identified or belong to a different part of speech.

subject-verb structures, *distance* is shown with respect to the verb: a distance of -1 means that the subject immediately precedes the verb. Although in most cases the subject is located to the left of the verb, in some constructions, such as existential clauses and inversions, it occurs after the verb.

Based on the accuracy results for identifying the structure components, we select those structures where the components are reliably identified. For article-NPhead, valid structures are those where the distance is at most three words. For subject-verb, we consider as valid those structures where the identified subject is located within two words to the left or three words to the right of the verb.

The valid structures are selected as input to the joint model (Section 9.5). The *joint learning* model considers only those valid structures whose components are *adjacent*. In adjacent structures the NP head immediately follows the article, and the verb immediately follows the subject. *Joint inference* is not restricted to adjacent structures.

The last column of Table 9.2 shows that valid subject-verb structures account for 67.5% of all verbs whose subjects are common nouns (51.7% are cases where the words are adjacent). Verbs whose subjects are common nouns account for 57.8% of all verbs that have subjects (verbs with different types of subjects, most of which are personal pronouns, are not considered here, since these subjects are not part of the noun classifier).

Valid article-NPhead structures account for 98.0% of all articles whose NP heads are common nouns (47.5% of those are adjacent structures), as shown in the last column of Table 9.3. 71.0% of articles in the training data belong to an NP whose head is a common noun; NPs whose heads belong to different parts of speech are not considered.

Note also that because a noun may belong both to an article-NPhead and a subject-verb structure, the structures contain an overlap.

9.5 The Joint Model

In this section, we present the *joint inference* and the *joint learning* approaches. In the joint inference approach, we use the independently-learned models from the Illinois system, and the interacting target words identified earlier are considered only at inference stage. In the joint

Distance	**Accuracy**	**% of all subj. predictions**	**Cumul.**
-1	97.6%	51.7%	51.7%
1,2,3	100.0%	8.9%	60.6%
-2	88.2%	6.9%	67.5%
Other	80.8%	32.5%	100.0%

Table 9.2: **Accuracy of subject identification on a random sample of subject-verb structures from the training data.** The overall accuracy is 91.52%. For each distance, we show the following statistics: accuracy based on comparison with human evaluation; the percentage of all predictions that have this distance; the cumulative percentage.

Distance	**Accuracy**	**% of all head predictions**	**Cumul.**
1	94.8%	47.5%	47.5%
2	94.4%	44.0%	91.5%
3	92.3%	6.5%	98.0%
Other	89.1%	2.0%	100%

Table 9.3: **Accuracy of NP head identification on a random sample of article-NPhead structures from training data.** The overall accuracy is 94.45%. For each distance, we show the following statistics: accuracy based on comparison with human evaluation; the percentage of all predictions that have this distance; the cumulative percentage.

learning method, we jointly learn a model for the interacting phenomena.

The label space in the joint models corresponds to sequences of labels from the confusion sets of the individual classifiers: {*a-sing*, *a-pl*, *the-sing*, *the-pl*, ∅*-sing*, ∅*-pl*} and {*sing-sing*, *sing-pl*, *pl-sing*, *pl-pl*} for article-NPhead and subject-verb structures, respectively.[4] Invalid structures, such as *pl-sing* are excluded via hard constraints (when we run joint inference) or via implicit soft constraints (when we use joint learning).

9.5.1 Joint Inference

In the individual model approach, decisions are made for each word independently, ignoring the interactions among linguistic phenomena. The purpose of joint inference is to include linguistic (i.e. structural) knowledge, such as "plural nouns do not take an indefinite article", and "agreement consistency between the verb and the subject that controls it". This knowledge should be useful

[4]*sing* and *pl* refer to the grammatical number of noun and verb agreement candidates. The candidates themselves are the surface forms of specific words that realize these grammatical properties. Note that a subject in subject-verb structures is always third person, since all subjects in subject-verb structures are common nouns; other subjects, including pronouns, are excluded. Thus the agreement distinction is singular versus plural.

for resolving inconsistencies produced by individual classifiers.

The inference approach we develop in this thesis follows the one proposed by Roth and Yih (2004b) that combines individually-trained models at decision time via joint inference. The advantage of this method is that it allows us to build upon any existing independently-learned models that provide a distribution over the local outcomes and produce a coherent global output that respects our declarative constraints. We formulate our component inference problems as ILP instances as in Roth and Yih (2004b).

The inference takes as input the individual classifiers' confidence scores for each prediction, along with a list of constraints. The output is the optimal solution that maximizes the linear sum of the confidence scores, subject to the constraints that encode the interactions. The joint model thus selects a hypothesis that both obtains the best score according to the individual models and satisfies the constraints that reflect the interactions among the grammatical phenomena at the level of linguistic structures, as defined in Section 9.4.

Inference The joint inference is enforced at the level of structures, and each structure corresponds to one ILP instance. All structures consist of two or three words: when an article-NPhead structure and a subject-verb structure include the same noun, the structure input to the ILP consists of an article-noun-verb triple. We formulate the inference problem as follows: given a structure s that consists of n words, let w_i correspond to the i^{th} word in the structure. Let h denote a hypothesis from the hypothesis space H for s, and $g(w_i, h, l^i)$ denote the score assigned by the appropriate error-specific model to w_i under h for label l from the confusion set of word w_i. We denote by $e_{w,l}$ the Boolean variable that indicates whether the prediction on word w is assigned the value l ($e_{w,l} = 1$) or not ($e_{w,l} = 0$).

We assume that each independent classifier returns a score that corresponds to the likelihood of word w_i under h being labeled l^i. The softmax function (Bishop, 1995) is used to convert raw activation scores to conditional probabilities for the discriminative article model. The NB scores are also normalized and correspond to probabilities. Then the inference task is solved by maximizing the overall score of a candidate assignment of labels l to words w (this set of feasible assignments is denoted H here) subject to the constraints C for the structure s:

$$\hat{h} = \arg\max_{h \in H} g(h) =$$

$$= \arg\max_{h \in H} \sum_{i=1}^{n} g(w_i, h, l^i) e_{w_i, l^i}$$

subject to $C(s)$.

Constraints In the $\{0, 1\}$ linear programming formulation described above, we can encode linguistic constraints that reflect the interactions among the linguistic phenomena. The inference enforces the following structural and linguistic constraints:

1. The indefinite article *a* cannot refer to an NP headed by a plural noun.
2. Subject and verb must agree in number.

In addition, we encode "legitimacy" constraints, that make sure that each w is assigned a single label. All constraints are encoded as hard constraints.

9.5.2 Joint Learning

We now describe how we *learn* the subject-verb and article-NPhead structures jointly. The joint model is implemented as a NB classifier and is trained in the same way as the independent models on the Google corpus with word n-gram features. Unlike the independent models, where the target corresponds to one word, here the target corresponds to two words that are part of the structure and the label space of the model is modified accordingly. Since we use features that can be computed from the small windows in the Google corpus, the joint learning model handles only adjacent structures (Section 9.4.1). Furthermore, because targets consist of two words and the Google corpus contains counts for n-grams of length at most five, the features are collected in the three word window around the target.[5]

In contrast to joint inference, the joint learning framework does not explicitly encode linguistic constraints. One reason for this is that the NP head and subject predictions are not 100% accurate, so input structures will have noise. However, the joint model learns these constraints through evidence seen in training.

[5]Also note that when the article is $\varnothing$, the surface form of the structure corresponds to the NP head alone; this does not present a problem because in the NB model the context counts are normalized with the prior counts.

9.6 Experiments

In this section, we describe our experimental setup and evaluate the performance of the joint approach. In the joint approach, the joint components presented in Section 9.5 handle the interacting structures described in Section 9.4. The individual classifiers of the Illinois system make predictions for the remaining words. The research question addressed by the experiments is the following: Given independently-trained systems for different types of errors, can we improve the performance by considering the phenomena that interact jointly? To address this, we report the results in the following settings:

1. *Joint Inference*: we compare the Illinois system that is a collection of individually-trained models that are applied independently with a model that uses joint inference encoded as declarative constraints in the ILP formulation and show that using joint inference results in a strong performance gain.

2. *Joint Learning*: we compare the Illinois system with a model that incorporates jointly-trained components for the two linguistic structures that we described in Section 9.4. We show that joint training produces an even stronger gain in performance compared to the Illinois model.

2. *Joint Learning and Inference*: we apply joint inference to the output of the joint learning system to account for dependencies not covered by the joint learning model.

As in Chapter 8, we report F1 performance on the original gold data.

9.6.1 Joint Inference Results

Table 9.4 shows the results of applying joint inference to the Illinois system. Both the article-NPhead and the subject-verb constraints improve the performance. The results for the joint inference are shown in two settings – adjacent structures and all structures – so that later we can compare joint inference with the joint learning model that handles only adjacent structures. It is also interesting to note that the key improvement comes from considering structures whose components are adjacent. This is not surprising given that the accuracy of identifying structure components drops as the distance increases.

To demonstrate the effect of the joint inference model, for subject-verb constraints, we also implement a naïve approach that looks for contradictions and changes either the verb or the subject if they do not satisfy the number agreement. These two heuristics are denoted as *NaïveVerb* and *NaïveNoun*. The heuristics differ from the joint inference in that they enforce agreement by always changing either the noun (*NaïveNoun*) or the verb (*NaïveVerb*), while the joint inference uses the scores produced by the independent models to enforce consistency. In other words, the key distinction is the objective function, while the components of the objective function are the same in the heuristics and the joint inference. The results in Table 9.4 show that simply enforcing agreement does not work well, while the ILP formulation is indeed effective and improves over the independently-trained models in all cases.

Recall that valid structures include only those whose components can be identified in a reliable way (Section 9.4.1). To evaluate the impact of that filtering, we perform two experiments with subject-verb structures (long-distance dependencies are more common in those constructions than in the article-NPhead structures): first, we apply joint inference to all subject-verb structures. We obtain an F1 score of 31.61; this is significantly worse than the results on subject-verb structures in Table 9.4 (31.97) and only slightly better than the baseline performance of the Illinois system. Furthermore, when we apply joint inference to those structures which were excluded by filtering in Section 9.4.1, we find that the performance degrades compared to the Illinois system, yielding an F1 of 30.85. These results demonstrate that the joint inference improvements are due to structures whose components can be identified with high accuracy and that identifying structure components accurately is essential.

The experiments above apply joint inference to the Illinois model. Recall that in Chapter 8 we found that it is possible to improve over the Illinois model by replacing some of its components. Notably, it was found that agreement errors benefit from a richer model that takes into account syntactic information. Table 9.5 shows the results of applying joint inference to this improved model, denoted here as *Illinois-improved*. Recall also that several components of this model are replaced, and the post-processing step is removed. For ease of comparison, joint inference results of the Illinois system are also shown. These results reveal that adding joint inference over the improved model that addresses agreement errors via a feature-rich classifier is also beneficial. We find that the gap between the baseline systems that do not use inference and those that do is generally the same in both case, with the exception of inference for adjacent subject-verb structures: the

System	**F1**	
	Adj. structures	**All structures**
Illinois (no inference)	31.20	31.20
NaïveVerb	31.19	31.13
NaïveNoun	31.03	30.91
Joint inference over the Illinois system		
Interacting structures	**F1**	
	Adj. structures	**All structures**
Subject-verb	31.90	31.97
Article-NPhead	31.63	31.79
Subject-verb + article-NPhead	**32.35**	**32.51**

Table 9.4: **Joint inference results.** All results are on the CoNLL-2013 test data using the original annotations. *Adjacent* denotes a setting, where the joint inference is applied to structures with consecutive components (article-NPhead or subject-verb). *All structures* denotes a setting, where the constraints are applied to all valid structures, as described in Section 9.4.1. *Illinois* denotes the result obtained by the top CoNLL-2013 shared task system. In all cases, the candidates that are not part of the structures are handled by the respective components of the Illinois system. *NaïveVerb* and *NaïveNoun* denote heuristics, where a verb or subject are changed to ensure agreement. All improvements over the Illinois system are statistically significant (McNemar's test, $p < 0.01$).

improvement due to the inference decreases slightly for the Illinois-improved model, which can be attributed to the contribution of syntactic features in the Illinois-improved agreement classifier. An important result of applying inference is that the predictions are coherent, which cannot be achieved in a model that combines independently-trained modules.

9.6.2 Joint Learning Results

Now we show experimental results of the joint learning model (Table 9.6). Note that the joint learning component considers only those structures where the words are adjacent. Because the Illinois system makes use of a discriminative article model, while the joint model uses NB, we also show results where the article model is replaced by a NB classifier trained on the Google corpus. In all cases, joint learning demonstrates a strong performance gain.

9.6.3 Joint Learning and Inference Results

Finally, we apply joint inference to the output of the joint learning system in Section 9.6.2. Table 9.7 shows the results of the Illinois model, the model that applies joint inference and joint learning separately, and both. Even though joint learning performs better than joint inference, the joint

No joint inference	**F1**	
	Illinois	**Illinois-improved**
	31.20	31.75

Interacting structures	**Joint inference**			
	Illinois (F1)		**Illinois-improved (F1)**	
	Adj. struct.	**All struct.**	**Adj. struct.**	**All struct.**
Subject-verb	31.90*	31.97*	31.97	32.35*
Article-NPhead	31.63*	31.79*	32.18*	32.35*
Subject-verb + article-NPhead	**32.35***	**32.51***	**32.41***	**32.90***

Table 9.5: **Joint inference results for the improved Illinois system.** All results are on the CoNLL-2013 test data using the original gold annotations. *Illinois-improved* denotes the system developed in Chapter 8. *Adjacent* denotes a setting, where the structure components are consecutive (article-NPhead or subject-verb), as described in Section 9.4.1. *Illinois* denotes the result obtained by the top CoNLL-2013 shared task system. In all cases, the candidates that are not part of the structures are handled by the respective components of the systems. Statistically significant improvements (McNemar's test, $p < 0.01$) over the no-inference systems are marked with an asterisk (*).

No joint learning	**F1**	
	Illinois system	**Illinois-NBArticle**
	31.20	31.71

Interacting structures	**Joint learning (F1)**	
Subject-verb	32.64	33.09
Article-NPhead	33.89	33.16
Subject-verb + article-NPhead	**35.12**	**34.41**

Table 9.6: **Joint learning results.** All results are on the CoNLL-2013 test data using the original annotations. *Illinois-NBArticle* denotes the Illinois system, where the discriminative article model is replaced with a NB classifier. *Adjacent* denotes a setting, where the structure components are consecutive (article-NPhead or subject-verb), as described in Section 9.4.1. *Illinois* denotes the result obtained by the top CoNLL-2013 shared task system. In all cases, the candidates that are not part of the structures are handled by the respective components of the Illinois system. All improvements are statistically significant over the Illinois and Illinois-NBArticle systems that do not use joint components (McNemar's test, $p < 0.01$).

System	**F1**
Illinois	31.20
Joint inference	32.51
Joint learning	35.12
Joint learn. + inf.	**35.21**

Table 9.7: **Joint Learning and Inference.** All results are on the CoNLL-2013 test data using the original annotations. Results of the joint models that include the joint inference component are shown for structures of all distances. *Illinois* denotes the result obtained in the CoNLL-2013 shared task. All joint systems demonstrate a statistically significant improvement over the Illinois system; joint learning improvements are also statistically significant compared to the joint inference results (McNemar's test, $p < 0.01$).

Example	**Illinois system**	**JL and JI**
"Moreover, the increased **technologies help** people to overcome different natural disasters.	*No change*	technology helps
"At that time,... **there are surveillances** in everyone's heart and criminals are more difficult to hide."	there are* surveillance*	there is surveillance
"In **such situation**, individuals will lose their basic privacy."	such a* situations*	such a situation
"In supermarket **monitor is** needed because we have to track thieves."	*No change*	monitors are

Table 9.8: **Examples of mistakes that are corrected by the joint model but not by the Illinois model.** *Illinois* denotes the result obtained by the top CoNLL-2013 shared task system from the University of Illinois. *JL* and *JI* stand for joint learning and joint inference, respectively. Inconsistent predictions are starred.

learning covers only adjacent structures. Furthermore, joint learning does not address overlapping structures of triples that consist of article, subject, and verb (6% of all structures). Joint inference allows us to ensure consistent predictions in cases not addressed by the joint learning model. Indeed, we can get a small improvement by adding joint inference on top of the joint learning.

9.7 Discussion and Error Analysis

In the previous section, we evaluated the proposed joint inference and joint learning models that handle interacting grammatical phenomena. We showed that the joint models produce significant improvements over the state-of-the-art Illinois that consists of independently-trained classifiers: the joint approaches increase the F1 score by 4 points (Table 9.7).

These results are interesting from the point of view of developing a practical error correction system. However, recall that the errors in the interacting structures are only a subset of mistakes in the CoNLL-2013 data set but the evaluation in Section 9.6 is performed with respect to all of these errors. From a scientific point of view, it is interesting to evaluate the impact of the joint models more precisely by considering the improvements on the relevant structures only. Table 9.9 shows how much the joint learning approach improves on the subset of relevant mistakes.

Error Analysis

To better understand where the joint models have an advantage over the independently-trained classifiers, we analyze the output produced by each of the approaches. In Table 9.8 we show examples of mistakes that the model that uses joint learning and inference is able to identify

Structure	**Performance (F1)**	
	Illinois	**Joint Learning**
Subject-verb	39.64	**52.25**
Article-NPhead	30.65	**35.90**

Table 9.9: **Evaluation of the joint learning performance on the subset of the data containing interacting errors.** All results are on the CoNLL-2013 test data using the original annotations. *Illinois* denotes the result obtained by the top CoNLL-2013 shared task system. All improvements are statistically significant over the Illinois system (McNemar's test, $p < 0.01$).

correctly, along with the original predictions made by the Illinois system.

Joint Inference Versus Joint Learning

We wish to stress that the joint approaches do not simply perform better but also make coherent decisions by disallowing illegitimate outputs. The joint inference approach does this by enforcing linguistic constraints on the output. The joint learning model, while not explicitly encoding these constraints, learns them from the distribution of the training data.

Joint inference is a less expensive model, since it uses the scores produced by the individual classifiers and thus does not require additional training. Joint learning, on the other hand, is superior to joint inference, since it is better at modeling interactions where multiple errors occur simultaneously – it eliminates the noisy context present when learning the independent classifiers. Consider the first example from Table 9.8, where both the noun and the agreement classifiers receive noisy input: the verb *help* and the noun *technologies* act as part of input features for the noun and agreement classifiers, respectively. The noisy features prevent both modules from identifying the two errors.

Finally, an important distinction of the joint learning method is that it considers all possible output sequences *in training*, and thus it is able to better identify errors that require multiple changes, such as the last example in Table 9.8, where the Illinois system proposes no changes.

9.8 Summary

This chapter presents the first successful study that jointly corrects learner mistakes. Two pairs of interacting phenomena are addressed: it is shown that it is possible to reliably identify their components, thereby facilitating the joint approach.

We describe two joint methods: a *joint inference* approach implemented via ILP and a *joint learning* model. The joint inference enforces constraints using the scores produced by the independently-trained models. The joint learning model learns the interacting phenomena as structures. The joint methods produce a significant improvement over the state-of-the-art Illinois system that combines independently-trained models. Importantly, the joint models produce linguistically-legitimate output.

Chapter 10

Conclusion

Despite growing interest in automatic text correction, research in this area is in its preliminary stages. This dissertation identifies and investigates several fundamental questions that establish promising directions for future research in this field.

In Chapter 5, several state-of-the-art machine learning algorithms were compared to determine which models are superior for the task and under what conditions.

Chapter 6 addressed the question of adaptation, that is developing models that can utilize knowledge about error regularities without expensive annotation. The key idea of adaptation is that estimating error patterns does not require a lot of annotation and can be done by extracting error statistics from a small set of writing samples. Taking into account specific error patterns such as the native language of a foreign-language learner demonstrably improves error correction. Figure 6.1 clearly demonstrated this empirically.

Chapter 7 discussed the challenges of addressing mistakes on open-class words and developed specific strategies for correcting grammar and usage mistakes on two groups of open-class words: nouns and verbs.

In Chapter 8, we described an end-to-end system that we built using the techniques proposed in the previous chapters. This system addressed article, preposition, noun number, and most commonly-occurring types of grammatical verb errors, and ranked first in the CoNLL-2013 shared task on grammatical error correction. Through analysis of the system and comparing it with the implementations by other teams, we identified four design dimensions that are closely aligned with the methods proposed in this thesis and showed their importance for building a state-of-the-art error correction system.

Finally, Chapter 9 proposed a global approach that goes beyond word level error correction and considers linguistic phenomena at the sentence level via joint inference and joint learning. The proposed methods not only improve the state-of-the-art system in Chapter 8 but also produce more coherent output that obeys linguistic constraints.

The Use of Linguistic Knowledge One feature that sets this work apart from other approaches to the problem of ESL error correction is the use of linguistic knowledge and the demonstration that it is crucial in the development of an error correction system. It is quite common in many NLP tasks to focus on the machine learning side of the problem, but our results demonstrate that the use of linguistic knowledge is essential.

It is well known that writers, both native and non-native, exhibit specific error patterns. The adaptation idea utilizes the fact that learner mistakes are not random. In this thesis, error regularities have been discussed at multiple levels – across multiple groups of learners, first-language dependent, or even individual– and exposing models to error patterns through adaptation has been shown to be extremely effective.

In Chapter 7, we showed that verb errors are linguistically complex, and building a model that addresses verb properties in a linguistically-motivated approach is beneficial. These findings are further supported by the performance of the verb modules in the complete system presented in Chapter 8.

Moreover, the Illinois system at CoNLL makes use of linguistic knowledge in several of its components. In addition to the linguistically-based approach to verbs, morphological and syntactic knowledge is used both in features and trigger identification for open-class words. Analysis of the Illinois system in Chapter 8 reveals the importance of using linguistic knowledge to build a successful system.

10.1 Future Work

In this section, we consider directions for future research that stem from the work in this thesis. Our findings raise a number of questions for further study. We consider future research from several points of view. First, in Section 10.1.1, we discuss future work that is immediately related to the problem of correcting ESL errors. Next, in Section 10.1.2, we discuss the application of the proposed approaches to mistakes made by other types of writers. Finally, Section 10.1.3 discusses future research that is further afield from the focus of this dissertation but is an interesting and compelling idea for future work.

10.1.1 Research Questions Relating to the Framework Studied in This Dissertation

Expanding the Set of Errors This thesis addressed the problem of ESL error correction by focusing on several prominent categories of grammar and usage errors. It is important to note that although not all of the error types have been addressed directly in this dissertation, we considered several fundamental questions pertaining to automatic error correction in general. One obvious and fruitful direction for future work is to expand further the set of errors covered. Here, we wish to point out two types of errors that are set apart from other mistakes made by learners: lexical choice errors and mistakes in word order.

Word choice errors constitute one category that is out of the scope of this work. As discussed in Section 4.2.4, word choice errors constitute 28% and 20% of all mistakes in the Illinois and FCE data sets, respectively, and form the largest error category in both of the corpora. Building an automatic system that could identify and correct word choice mistakes requires special considerations, such as determining the set of candidate corrections. However, we believe that the ideas proposed in this thesis will definitely benefit such a system. For example, taking into account the influence of the first language for lexical mistakes is possible and should be extremely useful.

Another special category comprises errors in word order. These errors account for about 2% of all mistakes in the Illinois and FCE data sets, but they are worth exploring and may present interesting challenges. A system that could identify these mistakes might benefit from the knowledge of the grammar of the first language of the writer, as well. Building such a system is an exciting direction for future research.

Interacting mistakes and global approaches to error correction Chapter 9 discussed mistakes that occur in interacting linguistic structures and showed that joint approaches both make coherent predictions and correct errors in contexts that require entertaining multiple changes simultaneously. There are several research directions in this area. The first direction concerns extending the proposed joint approaches to other interacting structures. In Chapter 9, we considered two types of interaction. Identifying other linguistic phenomena that would benefit from joint processing is an obvious extension of the joint approach to error correction.

Furthermore, the joint *learning* approach proposed in this thesis directly addresses the problem of multiple errors in the surrounding context of a target word as it is able to entertain hypotheses that require multiple simultaneous changes. Mistakes that are present in the context, even when they do not interact at the level of linguistic structures, nevertheless negatively affect the feature representation for a target word; addressing the problem of the presence of multiple mistakes is another application of the joint learning approach. Such an effort could substantially improve error correction on mistakes that do not interact grammatically.

10.1.2 Correcting Mistakes Made by Native Speakers and Learners of Languages Other Than English

One interesting direction for future work is applying the proposed techniques to errors made by native speakers and learners of languages other than English. The approaches proposed in this thesis are not language- or error-specific and therefore it would be easy to apply them to errors made by native speakers and to mistakes made by learners of languages other than English.

Error patterns are exhibited by both native and non-native writers alike, and adaptation, which in this thesis was proven extremely useful for correcting all types of ESL mistakes – both on closed- and open-class words – should also improve models that target native speakers and learners of languages other than English. Moreover, we believe that morphologically-rich languages could especially benefit from the idea of adaptation: analysis of the adaptation approaches in Chapter 6 suggests that the larger the confusion sets, the more the model benefits from knowledge of typical mistakes. It would thus be interesting to investigate adaptation methods in the context of correcting errors in morphologically-complex languages. For example, these methods could be applied to noun declination errors in languages like Russian or verb conjugation errors in Hebrew.

Joint approaches may be particularly useful for languages other than English, as morphologically-complex languages exhibit more interacting structures, and addressing these phenomena jointly should be beneficial. For example, constraints could encode the agreement between a noun and its modifiers – determiners or adjectives, or the dependence of case markings on the relationship between the words that form a grammatical structure.

10.1.3 Second Language Teaching

Although the focus of this dissertation is automatic error correction, our findings have direct and very interesting application in second language acquisition and teaching. Analysis of the annotated learner corpora provides quantitative evidence of the specific error patterns and the types of errors that are most commonly exhibited by ESL learners. As was shown, not all of these mistakes are equally likely to occur either within the group of learners of the same linguistic background or across different first languages. Furthermore, variation in error rates across different types of errors suggests that some language phenomena are more difficult to master for some ESL learners than for others. These findings can aid second language teaching, as knowledge about error patterns and the relation of these errors to the native language of the learner can be applied to target the mistakes made by non-native speakers as a way of quickly achieving the greatest improvement by those speakers. For example, instruction could be individualized based on the types of linguistic phenomena that are most difficult due to first-language influence or for learners in general. In fact, the idea of individualizing second language instruction based on the writers' native language is considered in Dalgish (1985), where error patterns for speakers of multiple first-language backgrounds are studied in the context of preposition and verb agreement mistakes. Determining whether such an effort could substantially improve second language acquisition remains to be seen. Moreover, the underlying idea behind the artificial errors approach could be directly used in this context to develop teaching materials that focus on the parts of grammar that are most challenging for the learner. Finally, an error correction system can also be integrated into second language classroom to facilitate the learning process by providing feedback to language learners; it can be further adapted to the areas that are particularly challenging to the learner.

Lastly, in Chapter 8, it was also shown that errors exhibit various difficulty levels for automated tools. Determining whether the same properties are more challenging both for humans and machines is an interesting direction for future work at the intersection of natural language processing, linguistics, and psychology.

It should stressed that, as discussed in Section 10.1.2, the research in this thesis is not restricted to ESL error correction. Therefore, these ideas can be also used to improve language teaching where the second language is a language other than English.

References

Banko, M. and Brill, E. (2001). Scaling to very very large corpora for natural language disambiguation. In *Proceedings of 39th Annual Meeting of the Association for Computational Linguistics*, pages 26–33, Toulouse, France.

Bergsma, S., Lin, D., and Goebel, R. (2009). Web-scale n-gram models for lexical disambiguation. In *21st International Joint Conference on Artificial Intelligence*, pages 1507–1512.

Bishop, C. (1995). *Neural Networks for Pattern Recognition, chapter 6.4: Modelling conditional distributions.* Oxford University Press.

Biskup, D. (1992). L1 influence on learners renderings of english collocations: A polish / german empirical study. *In J. Pierre, L. Arraud and H. Bejoint (eds.)*, 3:145–160.

Bitchener, J., Young, S., and Cameron, D. (2005). The effect of different types of corrective feedback on ESL student writing. *Journal of Second Language Writing.*

Boser, B. E., Guyon, I. M., and Vapnik, V. N. (1992). A training algorithm for optimal margin classifiers. In *Proc. 5th Annu. Workshop on Comput. Learning Theory.*

Brants, T. and Franz, A. (2006). *Web 1T 5-gram Version 1.* Linguistic Data Consortium, Philadelphia, PA.

Brockett, C., William, D. B., and Gamon, M. (2006). Correcting ESL errors using phrasal SMT techniques. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 249–256, Sydney, Australia. Association for Computational Linguistics.

Carlson, A. and Fette, I. (2007). Memory-based context-sensitive spelling correction at web scale. In *Proceedings of the IEEE International Conference on Machine Learning and Applications (ICMLA).*

Carlson, A. J., Rosen, J., and Roth, D. (2001). Scaling up context sensitive text correction. In *IAAI.*

Chang, M., Srikumar, V., Goldwasser, D., and Roth, D. (2010). Structured output learning with indirect supervision. In *Proc. of the International Conference on Machine Learning (ICML).*

Chen, S. and Goodman, J. (1996). An empirical study of smoothing techniques for language modeling. In *Proceedings of ACL 1996.*

Chodorow, M., Tetreault, J., and Han, N.-R. (2007). Detection of grammatical errors involving prepositions. In *Proceedings of the Fourth ACL-SIGSEM Workshop on Prepositions*, pages 25–30, Prague, Czech Republic. Association for Computational Linguistics.

Clarke, J. and Lapata, M. (2007). Modelling compression with discourse constraints. In *Proceedings of the 2007 Joint Conference of EMNLP-CoNLL.*

Dahlmeier, D. and Ng, H. (2012). A beam-search decoder for grammatical error correction. In *EMNLP-CoNLL*, Jeju Island, Korea. Association for Computational Linguistics.

Dahlmeier, D., Ng, H., and Wu, S. (2013). Building a large annotated corpus of learner english: The nus corpus of learner english. In *Proc. of the NAACL HLT 2013 Eighth Workshop on Innovative Use of NLP for Building Educational Applications*, Atlanta, Georgia. Association for Computational Linguistics.

Dahlmeier, D. and Ng, H. T. (2011). Grammatical error correction with alternating structure optimization. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 915–923, Portland, Oregon, USA. Association for Computational Linguistics.

Dale, R., Anisimoff, I., and Narroway, G. (2012). A report on the preposition and determiner error correction shared task. In *Proc. of the NAACL HLT 2012 Seventh Workshop on Innovative Use of NLP for Building Educational Applications*, Montreal, Canada. Association for Computational Linguistics.

Dale, R. and Kilgarriff, A. (2011). Helping Our Own: The HOO 2011 pilot shared task. In *Proceedings of the 13th European Workshop on Natural Language Generation.*

Dalgish, G. (1985). Computer-assisted ESL research. *CALICO Journal*, 2(2).

Felice, R. D. and Pulman, S. (2007). Automatically acquiring models of preposition use. In *Proceedings of the Fourth ACL-SIGSEM Workshop on Prepositions*, pages 45–50, Prague, Czech Republic.

Felice, R. D. and Pulman, S. (2008). A classifier-based approach to preposition and determiner error correction in L2 English. In *Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008)*, pages 169–176, Manchester, UK.

Fellbaum, C. (1998). *WordNet: An Electronic Lexical Database.* Bradford Books.

Freund, Y. and Schapire, R. E. (1996). Experiments with a new boosting algorithm. In *Proc. 13th International Conference on Machine Learning.*

Freund, Y. and Schapire, R. E. (1999). Large margin classification using the perceptron algorithm. *Machine Learning.*

Gale, W. A., Church, K. W., and Yarowsky, D. (1993). A method for disambiguating word senses in a large corpus. *Computers and the Humanities*, 26:415–439.

Gamon, M. (2010). Using mostly native data to correct errors in learners' writing. In *NAACL*, pages 163–171, Los Angeles, California.

Gamon, M. (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*, pages 180–189, Portland, Oregon. Association for Computational Linguistics.

Gamon, M., Gao, J., Brockett, C., Klementiev, A., Dolan, W., Belenko, D., and Vanderwende, L. (2008). Using contextual speller techniques and language modeling for ESL error correction. In *Proceedings of IJCNLP*.

Gamon, M., Leacock, C., Brockett, C., Dolan, W. B., Gao, J., Belenko, D., and Klementiev, A. (2009). Using statistical techniques and web search to correct ESL errors. *CALICO Journal, Special Issue on Automatic Analysis of Learner Language*, 26(3):491–511.

Gass, S. and Selinker, L. (1992). *Language transfer in language learning.* John Benjamins.

Golding, A. R. (1995). A bayesian hybrid method for context-sensitive spelling correction. In *Proceedings of the Third Workshop on Very Large Corpora.*

Golding, A. R. and Roth, D. (1996). Applying Winnow to context-sensitive spelling correction. In *ICML.*

Golding, A. R. and Roth, D. (1999). A Winnow based approach to context-sensitive spelling correction. *Machine Learning.*

Golding, A. R. and Schabes, Y. (1996). Combining trigram-based and feature-based methods for context-sensitive spelling correction. In *Proceedings of the 34th Annual Meeting of the Association for Computational Linguistics.*

Granger, S., Dagneaux, E., and Meunier, F. (2002). *International Corpus of Learner English.* Presses universitaires de Louvain.

Gui, S. and Yang, H. (2003). *Zhongguo Xuexizhe Yingyu Yuliaohu. (Chinese Learner English Corpus).* Shanghai Waiyu Jiaoyu Chubanshe. (In Chinese).

Han, N., Chodorow, M., and Leacock, C. (2006). Detecting errors in English article usage by non-native speakers. *Journal of Natural Language Engineering*, 12(2):115–129.

Han, N., Tetreault, J., Lee, S., and Ha, J. (2010). Using an error-annotated learner corpus to develop and ESL/EFL error correction system. In *LREC*, Malta.

Hanley, J. and McNeil, B. (1983). A method of comparing the areas under receiver operating characteristic curves derived from the same cases. *Radiology*, 148(3):839–843.

Hirst, G. and Budanitsky, A. (2005). Correcting real-word spelling errors by restoring lexical cohesion. *Journal of Natural Language Engineering*, 11(1):87–111.

Ionin, T. and Montrul, S. (2009). *Article use and generic reference: parallels between L1- and L2-acquisition.* John Benjamins.

Ionin, T., Zubizarreta, M., and Bautista, S. (2008). Sources of linguistic knowledge in the second language acquisition of english articles. *Lingua*, 118:554–576.

Izumi, E., Uchimoto, K., and Isahara, H. (2004). The overview of the SST speech corpus of Japanese learner English and evaluation through the experiment on automatic detection of learners' errors. In *LREC.*

Izumi, E., Uchimoto, K., Saiga, T., Supnithi, T., and Isahara, H. (2003). Automatic error detection in the Japanese learners' English spoken data. In *The Companion Volume to the Proceedings of 41st Annual Meeting of the Association for Computational Linguistics*, pages 145–148, Sapporo, Japan.

Kao, T.-H., Chang, Y.-W., w. Chiu, H., Yen, T.-H., Boisson, J., c. Wu, J., and Chang, J. (2013). Conll-2013 shared task: Grammatical error correction nthu system description. In *Proceedings of the Seventeenth Conference on Computational Natural Language Learning: Shared Task*, pages 20–25, Sofia, Bulgaria. Association for Computational Linguistics.

Lafferty, J., McCallum, A., and Pereira, F. (2001). Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *ICML*.

Leacock, C., Chodorow, M., Gamon, M., and Tetreault, J. (2010). *Automated Grammatical Error Detection for Language Learners*. Morgan and Claypool Publishers.

Lee, J. (2011). Verb tense generation. *Social and Behavioral Sciences*, 27:122–130.

Lee, J. and Seneff, S. (2008a). An analysis of grammatical errors in non-native speech in English. In *Proceedings of the 2008 Spoken Language Technology Workshop*.

Lee, J. and Seneff, S. (2008b). Correcting misuse of verb forms. In *ACL*, pages 174–182, Columbus, Ohio. Association for Computational Linguistics.

Liu, V. and Curran, J. R. (2006). Web text corpus for natural language processing. In *Proceedings of EACL*.

Madnani, N., Chodorow, M., Tetreault, J., and Rozovskaya, A. (2011). They can help: Using crowdsourcing to improve the evaluation of grammatical error detection systems. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 508–513, Portland, Oregon, USA. Association for Computational Linguistics.

Marneffe, M., MacCartney, B., and Manning, C. (2006). Generating typed dependency parses from phrase structure parses. In *LREC*.

Martins, A., N. Smith, N., Figueiredo, M., and Aguiar, P. (2011). Dual decomposition with many overlapping components. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 238–249, Edinburgh, Scotland, UK. Association for Computational Linguistics.

Mays, E., Damereau, F. J., and Mercer, R. L. (1991). Context based spelling correction. *Information Processing and Management*, 25(5):517–522.

Montrul, S. (2000). Transitivity alternation in L2 acquisition: Toward a modular view of transfer. *Studies in Second Language Acquisition*, 22:229–273.

Montrul, S. and Slabakova, R. (2002). Acquiring morphosyntactic and semantic properties of preterite and imperfect tenses in L2 spanish. *Perez-Lroux A-T and Liceras J (eds)*.

Nagata, R., Kawai, A., Morihiro, K., and Isu, N. (2006). A feedback-augmented method for detecting errors in the writing of learners of English. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 241–248, Sydney, Australia. Association for Computational Linguistics.

Nagata, R., Whittaker, E., and Sheinman, V. (2011). Creating a manually error-tagged and shallow-parsed learner corpus. In *ACL*, pages 1210–1219, Portland, Oregon, USA. Association for Computational Linguistics.

Nesselhauf, N. (2003). The use of collocations by advanced learners of english and some implications for teaching. *Applied Linguistics*, 24:223–242.

Ng, H. T., Wu, S. M., Wu, Y., Hadiwinoto, C., and Tetreault, J. (2013). The conll-2013 shared task on grammatical error correction. In *Proc. of the Seventeenth Conference on Computational Natural Language Learning.* Association for Computational Linguistics.

Odlin, T. (1989). *Language transfer: Cross-linguistic influence in language learning.* Cambridge University Press.

Oh, E. and Zubizaretta, M. (2003). Does morphology affect transfer? the acquisition of english double objects by korean native speakers. In *Proceedings of the 28th Annual Boston University Conference on Language Development*, pages 402–413. Burgos A, Micciulla L, and Smith C (eds).

Park, A. and Levy, R. (2011). Automated whole sentence grammar correction using a noisy channel model. In *ACL*, Portland, Oregon, USA. Association for Computational Linguistics.

Pradhan, S., Moschitti, A., Xue, N., Uryupina, O., and Zhang, Y. (2012). CoNLL-2012 shared task: Modeling multilingual unrestricted coreference in ontonotes. In *Joint Conference on EMNLP and CoNLL - Shared Task*, pages 1–40, Jeju Island, Korea. Association for Computational Linguistics.

Punyakanok, V. and Roth, D. (2001). The use of classifiers in sequential inference. In *NIPS*.

Punyakanok, V. and Roth, D. (2005). Inference with classifiers: The phrase identification problem. *Computational Linguistics.* In submission.

Punyakanok, V., Roth, D., and Yih, W. (2008). The importance of syntactic parsing and inference in semantic role labeling. *Computational Linguistics*, 34(2).

Quirk, R., Greenbaum, S., Leech, G., and Svartvik, J. (1985). *A Comprehensive Grammar of the English Language.* Longman, New York.

Radford, A. (1988). *Transformational Grammar.* Cambridge University Press.

Reichart, R. and Rappoport, A. (2010). Tense sense disambiguation: A new syntactic polysemy task. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 325–334, Cambridge, MA. Association for Computational Linguistics.

Riedel, S. and Clarke, J. (2006). Incremental integer linear programming for non-projective dependency parsing. In *EMNLP*.

Riedel, S. and McCallum, A. (2011). Fast and robust joint models for biomedical event extraction. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 1–12, Edinburgh, Scotland, UK. Association for Computational Linguistics.

Ringbom, H. (1978). The influence of the mother tongue on the translation of lexical items. *Interlanguage Studies Bulletin*, 3:80–100.

Rizzolo, N. and Roth, D. (2007). Modeling Discriminative Global Inference. In *Proceedings of the First International Conference on Semantic Computing (ICSC)*, pages 597–604, Irvine, California. IEEE.

Rod, E. (2008). *The Study of Second Language Acquisition.* Oxford University Press.

Rosenblatt, F. (1958). The perceptron: A probabilistic model for information storage and organization in the brain. *Psych. Rev.* (Reprinted in *Neurocomputing* (MIT Press, 1988).).

Roth, D. (1998). Learning to resolve natural language ambiguities: A unified approach. In *AAAI*.

Roth, D. (1999). Learning in natural language. In *IJCAI*.

Roth, D. and Yih, W. (2004a). A linear programming formulation for global inference in natural language tasks. In Ng, H. T. and Riloff, E., editors, *CoNLL*.

Roth, D. and Yih, W. (2004b). A linear programming formulation for global inference in natural language tasks. In *Proceedings of AI & Math*.

Roth, D. and Yih, W. (2007). Global inference for entity and relation identification via a linear programming formulation. In Getoor, L. and Taskar, B., editors, *Introduction to Statistical Relational Learning*.

Rozovskaya, A., Chang, K.-W., Sammons, M., and Roth, D. (2013). The University of Illinois system in the CoNLL-2013 shared task. In *CoNLL Shared Task*.

Rozovskaya, A. and Roth, D. (2010a). Annotating ESL errors: Challenges and rewards. In *Proceedings of the NAACL Workshop on Innovative Use of NLP for Building Educational Applications*.

Rozovskaya, A. and Roth, D. (2010b). Generating confusion sets for context-sensitive error correction. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Rozovskaya, A. and Roth, D. (2010c). Training paradigms for correcting errors in grammar and usage. In *NAACL*.

Rozovskaya, A. and Roth, D. (2011). Algorithm selection and model adaptation for esl correction tasks. In *ACL*.

Rozovskaya, A., Sammons, M., Gioja, J., and Roth, D. (2011). University of Illinois system in HOO text correction shared task.

Rozovskaya, A., Sammons, M., and Roth, D. (2012). The UI system in the hoo 2012 shared task on error correction.

Schütze, C. (1996). *The Empirical Base of Linguistics: Grammaticality Judgments and Linguistic Methodology.* University of Chicago Press.

Shannon, C. (1948). A mathematical theory of communications. *Bell Systems Technical Journal*, 27:623–656.

Sontag, D. and Jaakkola, T. (2007). New outer bounds on the marginal polytope. *Advances in Neural Information Processing Systems*, 20:1393–4000.

Stolcke, A. (2002). Srilm-an extensible language modeling toolkit. In *Proceedings International Conference on Spoken Language Processing*, pages 257–286.

Sutton, C. and McCallum, A. (2007). Piecewise pseudolikelihood for efficient training of conditional random fields. In Ghahramani, Z., editor, *ICML*.

Tajiri, T., Komachi, M., and Matsumoto, Y. (2012). Tense and aspect error correction for esl learners using global context. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 198–202, Jeju Island, Korea. Association for Computational Linguistics.

Tetreault, J. and Chodorow, M. (2008a). Native judgments of non-native usage: Experiments in preposition error detection. In *Coling 2008: Proceedings of the workshop on Human Judgements in Computational Linguistics*, pages 24–32, Manchester, UK. Coling 2008 Organizing Committee.

Tetreault, J. and Chodorow, M. (2008b). The ups and downs of preposition error detection in ESL writing. In *Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008)*, pages 865–872, Manchester, UK.

Tetreault, J., Foster, J., and Chodorow, M. (2010). Using parse features for preposition selection and error detection. In *ACL*.

Valiant, L. G. (1984). A theory of the learnable. *Communications of the ACM*.

Wu, Y. and Ng, H. (2013). Grammatical error correction using integer linear programming. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1456–1465, Sofia, Bulgaria. Association for Computational Linguistics.

Xiang, Y., Yuan, B., Zhang, Y., Wang, X., Zheng, W., and Wei, C. (2013). A hybrid model for grammatical error correction. In *Proceedings of the Seventeenth Conference on Computational Natural Language Learning: Shared Task*, pages 115–122, Sofia, Bulgaria. Association for Computational Linguistics.

Xing, J., Wang, L., Wong, D., Chao, L., and Zeng, X. (2013). UM-Checker: A hybrid system for english grammatical error correction. In *Proceedings of the Seventeenth Conference on Computational Natural Language Learning: Shared Task*, pages 34–42, Sofia, Bulgaria. Association for Computational Linguistics.

Yannakoudakis, H., Briscoe, T., and Medlock, B. (2011). A new dataset and method for automatically grading esol texts. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics*, pages 180–189, Portland, Oregon, USA. Association for Computational Linguistics.

Yarowsky, D. (1994). Decision lists for lexical ambiguity resolution: Application to accent restoration in Spanish and French. In *Proceedings of the 32nd Annual Meeting of the Association for Computational Linguistics*.

Yi, X., Gao, J., and Dolan, W. (2008). A web-based English proofing system for ESL users. In *Proceedings of IJCNLP*.

Yoshimoto, I., Kose, T., Mitsuzawa, K., Sakaguchi, K., Mizumoto, T., Hayashibe, Y., Komachi, M., and Matsumoto, Y. (2013). NAIST at 2013 CoNLL grammatical error correction shared task. In *Proceedings of the Seventeenth Conference on Computational Natural Language Learning: Shared Task*, pages 26–33, Sofia, Bulgaria. Association for Computational Linguistics.