

ABSTRACT

GOTH III, JULIUS. Intrasentential Grammatical Correction with Weighted Finite State Transducers. (Under the direction of James C. Lester.)

Natural language processing (NLP) offers significant potential for significantly enriching the communicative capabilities for a broad range of learning technologies. For example, both adaptive writing support environments and computer assisted learning environments could benefit from robust NLP. However, because texts created by novice writers pose significant challenges for core NLP systems such as syntactic and semantic parsers, robust grammatical pre-processing systems must be introduced upstream in the NLP pipeline. These challenges are exacerbated by the fact that current methods designed to detect and correct ungrammatical text focus on identifying and repairing specific types of errors, or rely heavily on contextual clues that may be unreliable in highly disfluent text.

To address these problems, we propose a noisy channel model implemented with weighted Finite State Transducers (wFSTs), where weights represent the probabilistic likelihood of transitioning between states, or in this case, words in a sentence. To construct our language model, we use a corpus of children's stories from Project Gutenberg. For the noise model, a corpus consisting of passages composed by middle school students obtained from corpus acquisition experiments is utilized. The EM algorithm identifies optimal *a priori* probabilities of encountering an erroneous form of a word. Preliminary results are encouraging and suggest that wFSTs offer significant promise for detecting and correcting texts exhibiting significant disfluency.

Intrasentential Grammatical Correction with Weighted Finite State Transducers

by
Julius Goth III

A dissertation submitted to the Graduate Faculty of
North Carolina State University
in partial fulfillment of the
requirements for the Degree of
Doctor of Philosophy

Computer Science

Raleigh, North Carolina

2013

APPROVED BY:

Christopher Healey

Kristy Elizabeth Boyer

Marc Russo

James C. Lester
Chair of Advisory Committee

DEDICATION

This work is dedicated to my wife and best friend, Joellen Frances Goth, whose love and encouragement made it possible, and to my mother Anna Goth and Carlos Mestres, who always provided me with unconditional support from a great distance.

BIOGRAPHY

Julius Goth III was born in New Brunswick, NJ on August 27, 1973. He grew up in Eastampton, NJ, a small town in the suburban regions of Philadelphia, where he attended local elementary, middle, and high school. He obtained a Bachelor of Science degree in Computer Science from the New Jersey Institute of Technology in 1998. He worked as an Information Technology consultant within a wide spectrum of industry sectors for nearly a decade, while attending school part-time to complete his Master of Science degree in Computer Science. During the period of his Ph.D. studies, he worked as an intern at IBM. Julius Goth married Joellen Frances Goth in 2011.

ACKNOWLEDGMENTS

First and most, I would like to thank my advisor James Lester, who has provided me continuous support and encouragements during my Ph.D. study. He has been a great advisor throughout the years of collaboration and also is a lifetime role model. I am deeply grateful to the members of my committee, Kristy Boyer, Christopher Healey, and Marc Russo, for their insightful comments and suggestions. I greatly benefited from working with talented colleagues in the IntelliMedia research group. Alok Baikadi implemented the 3D effects in the Unity game engine to generate narrative visualizations in the Narrative Theatre. I have enjoyed helpful discussions with Joe Grafsgaard, Eunyong Ha, Seung Lee, Bradford Mott, Chris Mitchell, Rob Phillips, Jonathan Rowe, Jennifer Sabourin, Andy Smith, and Michael D. Wallis. Last but not the least, I would like to thank my best friend and wife, Joellen Goth, and my mother and stepfather, for their patience, encouragements, and loving care during my pursuit of the Ph.D. study. It was impossible for me to complete this journey without their unconditional support. I am also very grateful to my mother, Anna Goth, for her emotional support. The research described in this dissertation was supported by the National Science Foundation grant IIS-0757535. Any opinions, findings, conclusions, or recommendations expressed in this dissertation are those of the author and do not necessarily reflect the views of the National Science Foundation.

TABLE OF CONTENTS

LIST OF TABLES	viii
LIST OF FIGURES	ix
CHAPTER 1 Introduction	1
1.1 Motivation.....	2
1.2 Challenges.....	5
1.3 Research Questions and Hypotheses	10
1.4 Approach.....	11
1.5 Contributions	13
1.6 Organization	14
CHAPTER 2 Related Work and Background	16
2.1 Cognitive Models of Writing.....	16
2.2 Analytics for Noisy Text	19
2.3 Grammatical Disfluency Influences	21
2.4 Types.....	22
2.5 Preposition Choice Errors.....	30
2.6 Article Choice Errors	30
2.7 Language Learning	31
CHAPTER 3 Lexicographic Graph-Based Grammar Correction.....	34
3.1 Task Definition	34
3.2 Over-Generate and Select Methods	35
3.3 Hidden Markov Models.....	36

3.4	Noisy Channel Model Theory	36
3.5	Automated Grammar Correction using wFSTs	38
CHAPTER 4 wFST Automatic Grammar Correction Framework.....		44
4.1	System Architecture.....	44
4.2	Language Model	45
4.3	Noise Model.....	48
4.4	Noise Model Induction	50
4.5	Language-Noise Model	53
4.6	Error Types	53
4.7	Maximum Likelihood Error-Free Text.....	57
4.8	Summary	59
CHAPTER 5 Noise Model Corpus Collection.....		60
5.1	Narrative Theatre	60
5.2	Study Design and Procedure.....	62
5.3	Corpus Observations.....	63
CHAPTER 6 Evaluation		68
6.1	Methodology.....	68
6.2	Data Set.....	72
6.3	Results.....	73
6.4	Discussion.....	77
6.5	Summary of Chapter.....	80
CHAPTER 7 Conclusion.....		82
7.1	Hypotheses Revisited.....	83

7.2	Summary.....	86
7.3	Limitations.....	87
7.4	Future Work.....	89
7.5	Concluding Remarks	90
	Appendices.....	102
	Appendix A: FST Semirings.....	103

LIST OF TABLES

Table 1. The verb eat and its inflections	27
Table 2. Part of speech confused word forms	30
Table 3. FST definitions	40
Table 4. Special arc types	47
Table 5. Narrative Theatre corpus acquisition statistics	64
Table 6. Grammar error categories	67
Table 7. Questionnaire results.....	73
Table 8. Results of all sentences	75
Table 9. Results of sentences with below average semantic quality	76
Table 10. Runtime speed.....	76
Table 11. Example sentences shown to crowdsourcing annotators	79

LIST OF FIGURES

Figure 1: Parse trees of intended sentence vs. observed sentence	4
Figure 2: Benign example of grammatical error	5
Figure 3. Hayes & Flower Cognitive Model of Writing.....	17
Figure 4. Distribution of Tweet Lengths.....	21
Figure 5. Confusion sets	24
Figure 6. Confusion maps	25
Figure 7. Noisy channel model architecture	37
Figure 8. Noisy channel model for machine translation	38
Figure 9. Noisy channel model adapted to automated grammar correction	39
Figure 10. Weighted Finite State Transducer	41
Figure 11. wFST system architecture	45
Figure 12. Bigram language model depicted as FST	46
Figure 13. Noise model.....	50
Figure 14. Determination of α and β	51
Figure 15. Training phase	52
Figure 16. Language-noise generative model.....	53
Figure 17. Observed sentence and sentence-language-noise FST	58
Figure 18. Example sentence MLE determination	58
Figure 19. NARRATIVE THEATRE interface	61
Figure 20. Grammatical error occurrence frequency	66
Figure 21. Recall, precision and F-measure.....	69
Figure 22. SRL performance scoring example	71
Figure 23. Train/Test cross-validation methodology.....	73
Figure 24. Highly ambiguous input	78

CHAPTER 1

Introduction

Natural Language Processing (NLP) systems are responsible for the correct interpretation and comprehension of written or spoken input, and transforming this input into meaningful semantic representations. Because natural language is often highly ambiguous and imprecise, this is a challenging task. Natural language can include indirect expressions, a complex discourse hierarchy, as well as idiomatic references. However, even when these challenges are addressed in the context of interpreting well-written text, what happens when NLP systems do not have the luxury of interpreting well-formed text? How would systems designed interpret ill-structured text fare in cases where the wording and phrasing are distorted? NLP components, such as syntactic parsers, are very effective at analyzing the grammatical structure of textual input. However, these tools utilize learned models that are trained with corpora such as newswire information (i.e., *Wall Street Journal*) and various sources from literature such as the Brown Corpus (Klein and Manning 2002). Unfortunately, in many cases of written text, especially those created by novice writers, this ill-formed text will likely pose significant challenges for core components of the NLP pipeline such as syntactic parsers (Collins 1999) and semantic role labelers (Gildea and Jurafsky 2002). One alternative is to train new grammatical and semantic analysis tools that directly address the

disfluencies inherent in the problem domain. However, this approach is problematic because of the sheer amount of time associated with hand annotating training sentences, a necessary task in the development of syntactic and semantic analysis models. Another alternative is to develop an automated preprocessing system that provides grammatical correction functionalities, which offer the potential for improving the overall grammatical quality of text yet retaining the writer's intended meaning.

Automatic grammar correction techniques have seen improvement in recent years. However, many approaches are limited to examining a subset of grammatical disfluency categories, such as real-word spelling errors, verb tense misuse, or improper preposition usage. Some approaches focus on contextual features in localized context relative to an observed word replacement candidate. Utilizing localized context conveys a plausible procedure for scenarios where contextual cues are largely intact. However, in highly noisy text these contextual clues may also have been ill-formed (Pedler 2007). As a result, many conventional techniques fail due to incorrect or missing contextual features. As a result, it is necessary to explore possible solutions possessing the ability to exploring many combinations of alternatives simultaneously.

1.1 Motivation

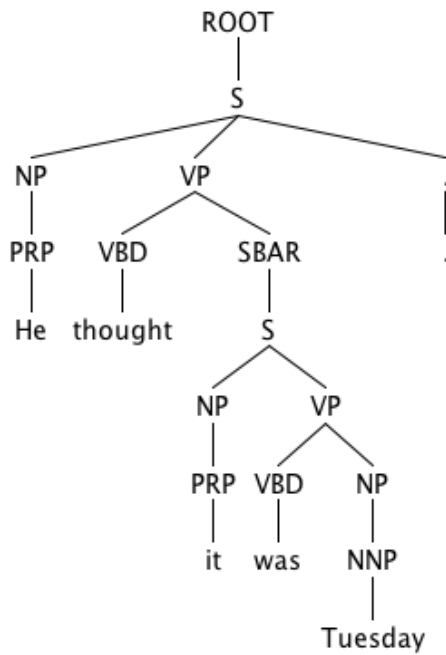
Due to the constraints of syntactic and semantic parsers requiring clean, grammatical text to process textual input, ill-formed grammar presents a significant challenge for natural language understanding (Eastman and McLean 1981). Consequently, the task of designing systems utilizing natural language processing for texts created by novice writers is

challenging. The lack of correct grammar in a text passage can confuse NLP components designed to syntactically and semantically interpret the texts. Additionally, since its pipeline architecture is structured in such a way as to have each component rely on output from its precedent component, a failure in an early stage can result in a cascading and catastrophic failure in the NLP engine's ability to interpret the intended meaning. Unfortunately, this area of research is by in large ignored by the NLP community (Voorhees et al. 2005).

As an example, consider the intended sentence: *He thought it was yesterday*. The syntactic parse tree for this sentence contains a noun phrase consisting of a determiner, a verb phrase containing a past tense verb along, and a sentential clause (Figure 1). A semantic analysis of this sentence yields a predicate consisting of the lexeme *thought*, supported by two arguments: *He* as subject (A_0), and *it was yesterday* as direct object (A_1). However, altering the lexeme *thought* to *though* invalidates the action of acquiring of the predicate *thought*, resulting in nullifying not only a predicate, but also the supporting roles A_0 and A_1 .

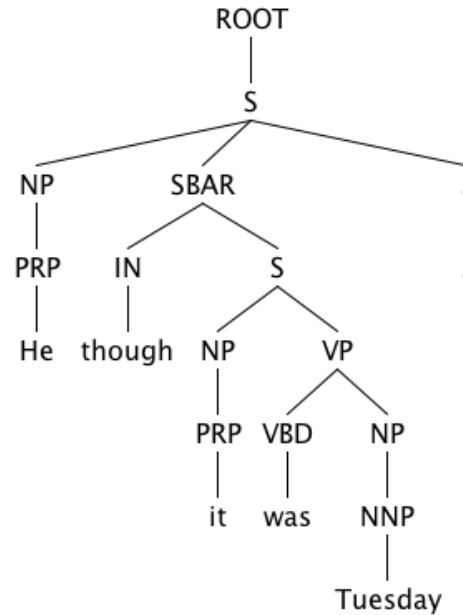
To address these challenges, automated grammar correction systems play a central role in the remediation of these errors and by consequence, enhancing pedagogical systems utilizing syntactic and smenatic analysis components. To that end, a key challenge posed by these methods is accumulating relevant domain-specific corpora. Additionally, many conventional approaches rely on linguistic contextual features to formulate corrections for target words or phrases. If the contextual clues consist of erroneous input, the methodology suffers. The system either has to ignore the input, or if no such fallback mechanism exists, to directly incorporate the errors into the model and potentially influencing the outcome.

He *thought* it was Tuesday



[He_{arg0}] thought [it was Tuesday_{arg1}]

He *though* it was Tuesday



N/A

Figure 1: Parse trees of intended sentence vs. observed sentence

While this type of grammatical error yields particularly devastating results in extracting semantic representation within an NLP pipeline process, other errors are rather benign by comparison. For instance, consider the sentence *He set foot in the island* (Figure

2). In this case, the sentences have equivalent syntactic features and only differ on the lexicon representing the destination preposition.

He set foot *on* the island.

He set foot *in* the island.

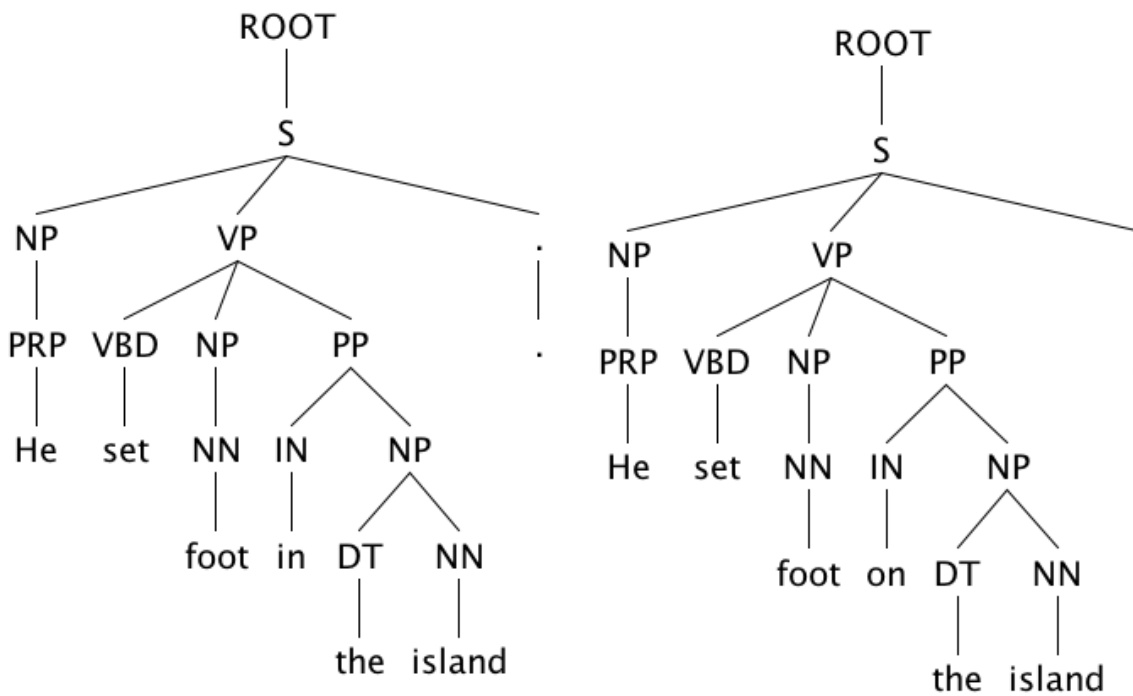


Figure 2: Benign example of grammatical error

1.2 Challenges

This work proposes a novel approach to automated grammar correction. Given a single observed sentence at a time, the proposed grammatical corrector aims to generate the most likely alternative that retains the original sentence based on an overall scan of the sentence

and probabilistic inference from a noisy channel model. The proposed task poses a number of challenges, which are described in the following sections.

1.2.1 Grammar Error Detection

One of the principal challenges in dealing with the automatic identification of grammatical errors is the wide array of grammatical error types that exist in languages. Real-word spelling errors, verb tense misuse, preposition misuse, and article errors.

The task of grammatical error detection is extremely challenging, in part due to the somewhat subjective nature of grammatical errors. Observationally, for humans, some grammatical errors are more easily detected than others. On the other hand, some are less glaring than others. For instance, it has been argued that composing a sentence ending with a preposition is considered bad form. However, scholars have debunked this theory at various levels. Some indicate that in such cases, trailing prepositions that cannot be removed to retain grammaticality are considered acceptable (Aitchison 2001).

While some grammatical errors can be objectively identified, others can be significantly nuanced and even create confusion among some readers as to determining the optimal correction suggestion. Additionally, each word in a sentence is a potential candidate for correction. In some cases, grammatical error types often overlap in early learners' writing (Weber 1970).

Some error types require further examination of high-level discourse features, specifically those lying outside of localized context. For example, a document centered on a particular topic such as dairy farming has a propensity for using the lexeme *dairy*. If a

spelling variation such as *diary* (inverting i and a) were included, the likelihood of a spelling error occurring would be much higher. Based on this notion, Hirst and Budanitsky (2005) introduce an approach to explore passages for discontinuities in lexical chains.

The task of identifying legitimate grammar errors is complicated by several factors. First, some spell correctors unconditionally interpret proper nouns to be legitimate spellings. This can present problems when words are accidentally or intentionally typed as uppercase. For example, consider the sentence *The Cag was purring*, where the word *cat* was accidentally spelled *Cag*. While the word *Cag* is clearly a non-word in the context of non-proper nouns, spell correctors have little in the way of methodology to suggest corrections due to it being identified as a proper noun. Perhaps *Cag* is an uncommon name for an individual or otherwise a newly defined term in modern culture? This task has been partially addressed in the medical domain using named entity recognition techniques (Ruch 2003). Another way grammar correction can be complicated is through the usage of intentionally misspelled idiomatic references and unseen foreign lexemes or phrases, such as *git 'er done* or *merci beaucoup*. This type of problem is not typically addressed in expository or non-fictional essay writing. However, in narrative centered writing tasks, these “erroneous” lexemes often exist (Abkarian et al. 1990). To date, there has not been a focused effort to address this problem.

1.2.2 Correction Candidate Generation

The second phase of automatic grammar detection and correction consists of processing the grammar error candidates obtained from the detection phase by generating and ranking candidates. Grammar correction requires extensive knowledge of various grammatical error categories being investigated, in addition to understanding the behavior of the target user audience. Possessing foreknowledge of correction alternatives is paramount, as these alternatives cannot be expressed if they do not exist in any relational mapping. This prerequisite partially overlaps with possessing an understanding of the user audience, as their behaviors and patterns in composing text present clues as to how these mappings are constructed. However, it is also important to model various subgroups of students, such as second language learners (referred to as *L2* learners) versus native learners who have differing cognitive processes for writing and consequently present differing frequencies and types of errors (Low and Siegel 2005).

One of the other major challenges relating to grammatical correction depends on how these correction candidates are presented. For interactive systems, a selection list of various alternatives may be provided. This list may be accompanied by explanations on why those particular alternatives have been offered to the user. However, in grammar detection and correction systems employing an automatic correction trigger mechanism, the highest ranked correction hypothesis is usually selected. This may be desirable in situations where feedback is not necessary or desired, but may present problems for learning if the correction hypothesis is incorrect for that particular context.

1.2.3 Feedback Generation

Systems relying on informative pedagogical feedback to convey results from error diagnosis require a methodology to organize this feedback appropriately. Some grammar errors feature conditions in which multiple error conditions exist. As of late, work has emerged suggesting that grammatical correction suggestions are detrimental to L2 learners (Truscott 2006). However, it must be noted that the nature of the pedagogical systems' feedback plays an important role in how effective these suggestions are to the student. Oversuggesting corrections can result in frustration and boredom for students, and balancing the active or passive nature of the feedback play important roles in maintaining interest in learning.

1.2.4 Highly Disfluent Text

Grammar correction poses significant computational challenges, particularly for the highly disfluent texts composed by novice writers. Text exhibiting a high concentration of grammatical errors presents additional challenges versus text with fewer errors due to the limitation of localized context, or word windows, used by some algorithms. Foundational work has sought to address this problem through relaxation techniques (Kwasny and Sondheimer 1981) and current work addresses grammar detection in language learning (Leacock et al. 2010). However, these approaches are primarily geared toward providing rich pedagogical feedback in an interactive environment. Work associated specifically targeting the remediation process to improve the quality of text to be processed by NLP methods is limited.

Significant challenges are posed by ill-formed natural language text. A failure in the early stages of analysis has the potential of disrupting the information extraction process (Fitzgerald and Jelinek 2008). For example, an incorrect morphological analysis of a passage may result in cascading errors. As a consequence of this, extracting syntactic features may result in parse trees bearing little resemblance to a parse tree representing the intended meaning. Semantic analysis often utilizes the syntactic features resulting from parsing techniques as a means to identify predicate and semantic roles

1.3 Research Questions and Hypotheses

With a focus on maximizing accuracy for automatic grammar correction tools using a noisy channel model, the following hypotheses are proposed:

Research Question 1: How can human annotation reliably establish a gold standard to compare system performance?

Hypothesis 1: Human judgement via a crowdsourcing task provides a sufficiently viable benchmark for our analysis.

Research Question 2: How can it be determined that weighted Finite State Transducers possess the capability to detect and correct grammar errors with high accuracy?

Hypothesis 2: The implemented noisy channel implementation provides significantly improved accuracy in relation to the grammar detection and correction facilities of two state-of-the-art systems. One of these systems is the grammar detection facility of

Google Docs, a web-based word processor system. The other is the online version of the Winnow context-sensitive spelling corrector.

Research Question 3: How can it be determined that weighted Finite State Transducers possess the capability to detect and correct grammar errors with high efficiency?

Hypothesis 3: The implemented system provides the automated grammar correction task in a bounded time limit comparable to both Google Docs and UIUC's Winnow based context-sensitive spelling corrector.

1.4 Approach

The task of automatic grammar correction can be cast as a machine translation problem. In machine translation, an intermediate process is responsible for decoding text from a source language to a destination language. Likewise, ungrammatical text can be thought of as the source language in the same paradigm, and a remediated form of the text with the same meaning as the destination language. To this end, the dissertation approaches this problem using a graph-based noisy channel model. Graph models are an efficient means of marrying graph theory and probability theory. Several types of models are implemented within this family and are used extensively in research, including Hidden Markov Models, Conditional Random Fields, Bayesian Networks, Neural Networks, among others (Alpaydin 2004). One type of graphical model that has seen focus in recent years is weighted Finite State Transducers (wFSTs). This model has been especially useful in modeling noisy information

accurately, and fits nicely in the role for detecting and correcting grammatical errors. The approach in this dissertation utilizes weighted Finite State Transducers exclusively to support the adaptation of probabilistic weights for multiple grammatical error types and provides the capability to explore a vast frontier of candidate inputs from an observed input. Our research methodology utilizes the following phases:

- **Corpus Acquisition:** To provide a basis for the computational model to generate a noise model and establish probability estimates, two corpus gathering experiments are conducted.
- **Language Model acquisition:** Using a subset of the corpus made available from the Gutenberg Project (Hart 2000), an FST derived from a smoothed n-gram statistical language model is learned (Chelba and Jelinek 2000).
- **Noise model construction:** Based on a set of interesting and relevant grammatical disfluency types, a set consisting of a combination of manually handcrafted and automatically extracted grammatical errors is produced, drawn from studies involving a prototype writing support environment, the NARRATIVE THEATRE.
- **Model building:** Using the language and noise model as inputs, a composite model is computed in preparation for observed input sentences.
- **Weight learning:** The probabilistic weights for an FST can either be uniform, hand-engineered, heuristically computed or learned through supervised/unsupervised methods. The Expectation Maximization algorithm is used to train the weights.

Model Evaluation: Using text manually corrected from a crowdsourcing task as a gold standard, comparison algorithms based on precision and recall metrics of semantic role labeling output. The values generated from our wFST model is compared alongside the Google Docs spell check/grammar correction tool as well as UIUC's state-of-the-art grammar correction model.

1.5 Contributions

This dissertation makes the following contributions to the field of NLP.

- **Whole sentence correction of noisy text.** The wFST approach for remediating noisy text models probabilistic relationships among alternative spellings of observed text, and does so on a whole sentence level. This approach can be desired over examining lexemes or phrases in isolation to determine optimal corrections (Park and Levy 2011).
- **Expansive grammatical error checking capabilities.** The ability of automated grammatical error detectors to identify a wide variety of grammatical error categories is essential in achieving favorable results. The noisy channel model used in this implementation allows flexibility for expansion across several error types as opposed to focusing on one or two different types of grammatical errors. This work in particular expands on real-word spelling error detection and correction by incorporating QWERTY related edit distance in addition to errors rooted in phonology.

- **Unsupervised training of models on ungrammatical text.** The advantages of a computational model trained on text not requiring annotation are significant. The cost of annotation for data can be quantified in terms of time. There exists a time delay to train users to efficiently identify grammatical errors and to properly annotate them via a well-established protocol. The majority of approaches for inducing models relating to automated grammar correction rely on annotated data.
- **Focus on narrative writing.** Approaches incorporating automated grammatical detection and correction typically focus on ESL learners, and in particular focus on expository writing. Narrative writing is distinctively different than expository writing, in particular with the literary structure that is absent in expository text, such as plot and voice. Consequently, the elements in their cognitive processes differ (Spiro and Taylor 1980).

1.6 Organization

The document is structured as follows. Chapter 2 explores related work in the context of cognitive models of writing explore various categories of grammatical disfluency and the context of related work, as well as explore a mainstream application where grammatical correction is often applied, Intelligent Computer Assisted Language Learning Systems. Afterwards, Chapter 3 aims to explore graph-based approaches using various computational models to automatically detect and correct errors in ungrammatical text and offers a background in weighted Finite State Transduction to approach the problem of grammatical

disfluency. Chapter 4 presents in detail the two-stage training as well as runtime phases of the system. Then, Chapter 5 introduces the corpus acquisition experiment using the Narrative Theatre writing support environment. Chapter 6 then presents the evaluation of the runtime phase on crowdsourced annotated experimental data using a cross-validation technique along with a discussion of the results, and Chapter 7 concludes with final remarks and future work.

CHAPTER 2

Related Work and Background

The work presented in this dissertation draws on several lines of research in linguistics that have been developed independently of each other, including cognitive models of writing and noisy text analytics. The following sections provide overviews of each of these areas. A prerequisite for designing models to automatically detect and correct grammatical errors is a deep understanding of different categories of grammatical errors. In this section, we review the influences and types of errors.

2.1 Cognitive Models of Writing

Much of the groundwork associated with understanding the mental processes involved in writing was introduced by Hayes and Flower (Hayes and Flower 1980). Hayes & Flower's model partitioned the overall task of writing into three basic processes: planning, which consists of idea generation, goal setting, and organization of structure; translation of planning to text; and reviewing (Figure 3). The processes are provided two types of information: a task environment context, consisting of the assignment and the produced text thus far, and knowledge stored in long-term memory, such as a model of the reading audience, writing plan, topic background knowledge, grammar production rules, and background knowledge of text standards.

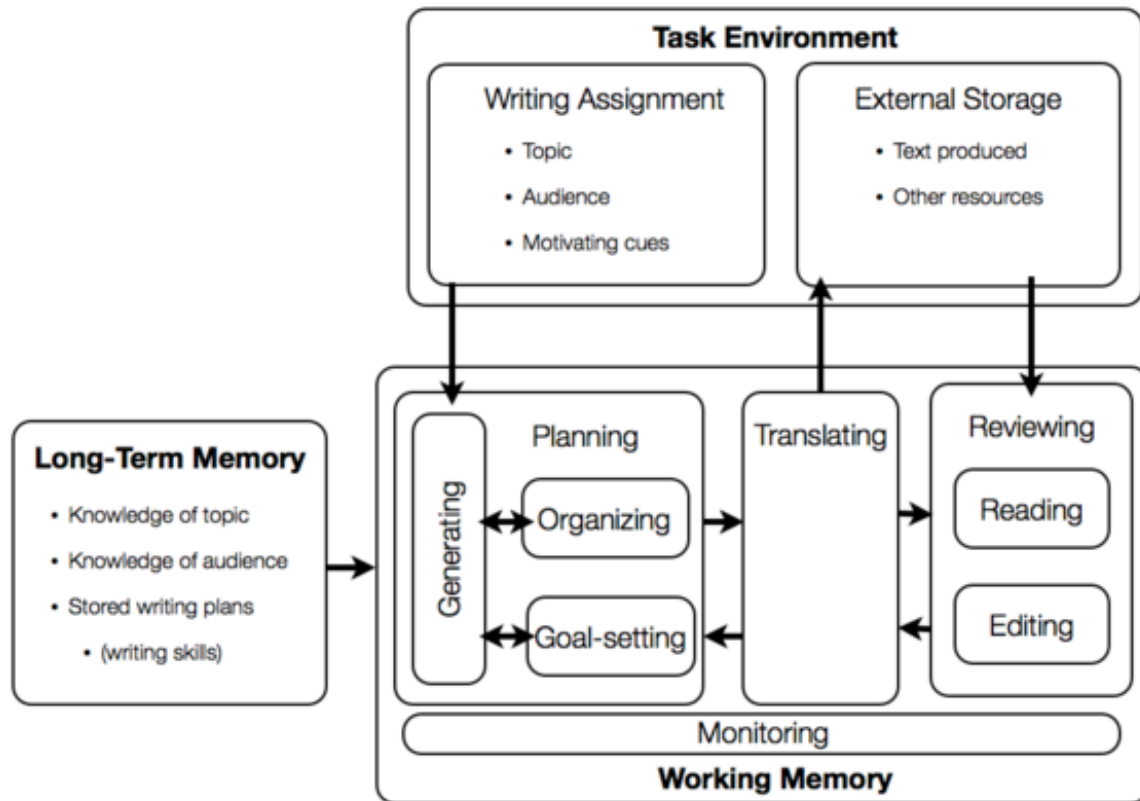


Figure 3. Hayes & Flower Cognitive Model of Writing

Perhaps one of the more important extensions of this research was the characterization of the differences between expert and novice writers (Hayes 1996). In this work, it was determined that expert writers tended to have a more highly refined and elaborate goal network. Also, expert writers tended to be more eager to deeply revise their passages above that of a word or sentence level, concentrating on meaningful discourse changes that altered the encompassing meaning of the writing. An important factor in dealing with novice writers is the concept of “cognitive load.” This phenomenon typically results

from the writer's relatively limited working memory capacity being exceeded by the complex set of processes required in composing a writing passage. For example, it was determined that spoken responses elicit less cognitive load among grade school children (Bourdin and Fayol 1994) but not for adults. This hypothesis was reinforced in (Bourdin et al. 1996) using narrative recall. In a third study (Bourdin and Fayol 2002), it was found that even expert writers, or adults in this case, exhibited a difference in performance of spoken versus written responses when the sentence production task become significantly more complex. Thus, managing cognitive load becomes a significant issue, especially for novice writers, and well-designed writing support tools should be able to strike a more delicate balance between the two factors. One common strategy for dealing with cognitive load is outlining ideas prior to the writing task. Research on outlining writing exercises as done in (Kellogg 1988) and (Kellogg 1990) suggests that cognitive load may be reduced, as writers are able to dedicate more of their mental faculties to articulating their ideas effectively in text. However, (Baaijen et al. 2008) found that for dyslexic writers, planning did not influence the quality of the produced text, which actually counters the previous research. In an updated version of the Hayes & Flower foundational writing model, (Hayes 1996) hypothesized that revision is influenced by a writer's working and long-term memory. Perhaps one of the most striking aspects of the newer model is the explicit incorporation of working memory, and expanded into multiple components as opposed to a monolithic resource. This work also reinforced the notion that expert writers tend to develop more elaborate plans, as well as refining these plans through the duration of their writing. It also supports the notion that revisions among expert writers tended to be deeper than structural changes from a sentential level.

Hayes and Flower (1980) hypothesized that novice writers too focused on grammar and spelling encounter difficulties in story planning. This is probably due to the fact that short-term memory recall may suffer as a result of devoting too many resources on spelling correction during the transcription process. Evidence of the negative effects of poor spelling skills was realized by Graham (1990, 2002) based on an evaluation with students who were subjected to supplemental spelling instruction versus a control group that underwent mathematics instruction. It was concluded that, at least in the short term, writing fluency was significantly improved among the spelling instruction group.

In short, it is clear that reducing cognitive load is a key factor in promoting novice writers' ability to plan better, resulting in richer, more compelling prose. To that end, mitigating students' need to fix shallow sentential errors in order to concentrate on bigger picture tasks, such as planning, would stand to benefit novice writers significantly.

2.2 Analytics for Noisy Text

Information extraction from structured text is a mature field. Techniques to syntactically parse and semantically interpret input from well-written text abound in research literature and in commercial applications. Unfortunately, in some cases, noise is ubiquitous in various categories of text. Most notably, in such areas as communication channels (chat, SMS, e-mails, blogs, instant messaging, etc), automatic speech recognition (ASR) output, optical character recognition (OCR), among others (Subramaniam et al. 2009). The generalized term for processing this type of text is sometimes referred to as noisy text analytics.

One area that holds particular promise in analyzing noisy text is social networking. In particular, sentiment analysis of microblogging tools such as Twitter feeds (Barbosa and Feng 2010; B. Han and Baldwin 2011). The distribution of feed entries, or “tweets,” on Twitter does not follow a normal distribution. In actuality, there is a propensity of tweets to consume most or all of the available character length¹ (Figure 4).

Techniques to address the issue of noisy text also examine the problem from an alternate angle. Rather than remediating text to preserve intent prior to forwarding content to syntactic and semantic analysis components, the downstream components could relax conditions for well-formed text. Such techniques are illustrated in noisy data adapted Part of Speech taggers (Gadde et al. 2011).

¹ <http://www.robweir.com/blog/2011/01/twitter-2010-by-the-numbers.html>

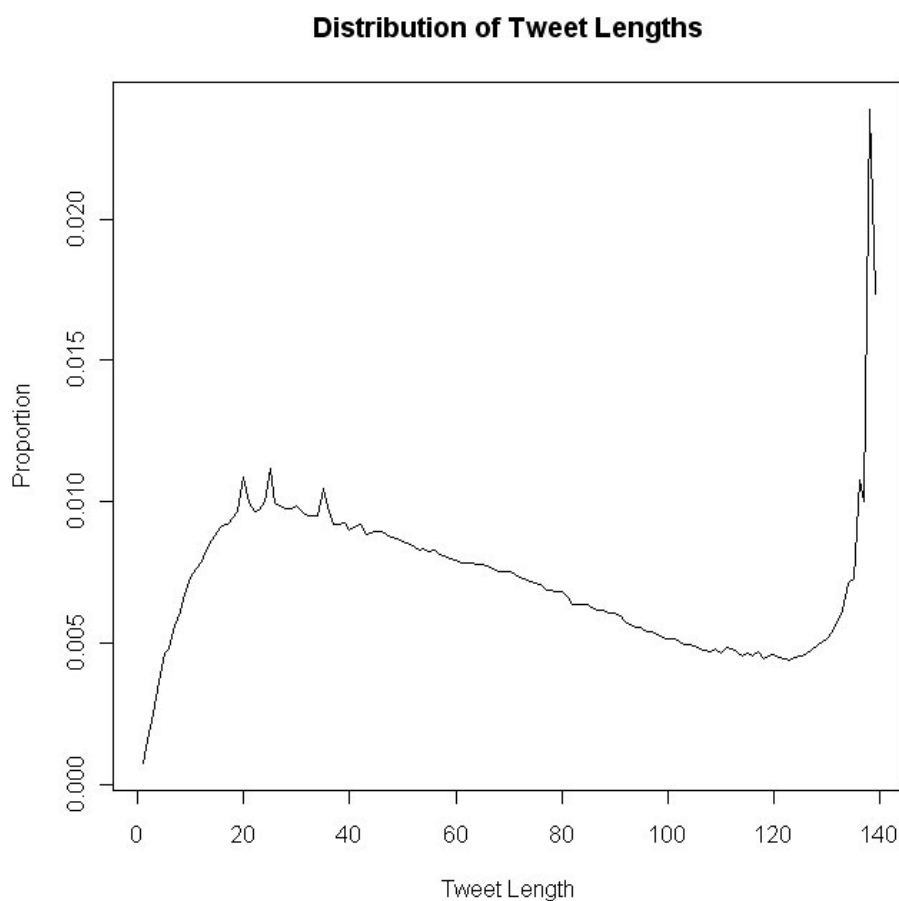


Figure 4. Distribution of Tweet Lengths

2.3 Grammatical Disfluency Influences

The causes of grammatical disfluencies among novice writers are very complex and not easily identified, but some answers may lie in the development of children's cognitive writing ability (Scardamalia and Bereiter 1987). Additionally, they may be an artifact of long-term usage of SMS devices (Coe and Oakhill 2011; Ryker et al. 2011). In other cases, an overreliance on conventional spell correctors can hinder writing ability and can potentially

be a catalyst for introducing much more challenging grammatical errors. (Deorowicz and Ciura 2005).

2.4 Types

Numerous grammatical error types may exist in written text and are well documented in writing self-improvement literature or online resources. Several key types are described in this section.

2.4.1 Real-Word Spelling Errors

A real-word spelling error occurs when a correctly spelled word is substituted for the word that was intended. Real-word spelling errors account for as much as one quarter to one third to of all spelling errors (Mitton 1996). While this percentage does not seem particularly high, the percentage can vary greatly and is undoubtedly higher among children learners.

Many word processing applications include a form of spell correction. These applications typically work by examining lexemes in isolation, and errors are identified if a word does not appear in a pre-defined dictionary. The error may be automatically corrected, but in most cases the correction algorithm offers the user with a list of best matching alternatives that can be selected. Although this approach is adequate for detecting the majority of misspellings, many other types of possible error combinations remain where the word in question is a legitimate word but completely inappropriate for the context. As such, there exists no trivial triggering mechanism that identifies any particular word as a potential candidate. As such, all words must be considered suspect.

Though the root causes on how writers introduce real-word spelling errors has been studied (Vitevitch 1997), in many cases it is not entirely predictable. In some cases, a user will create a typographical error by substituting a key nearby the intended key, for example typing the word *in* versus *on*. In other cases, students (especially early learners) may use phonological clues to determine words (Scardamalia and Bereiter 1987). Some examples of these types of errors are *where/ware/were* and *their/there*.

Recent versions of Microsoft Word have embedded a context-sensitive spell checker and work has been done to examine its effectiveness (Hirst 2009). While the algorithmic details of the spell checker are opaque due to their proprietary nature, the spell checker is successful in identifying various rudimentary errors, and in empirical studies the application does suggest suitable corrections to the user. A rule-based contextual spell checker and grammar correction tool using hand-engineered regular expression notation has also been introduced as a Wordpress add-on producing favorable results on many commonly mistyped words (Mudge 2010).

Other types of words are sometimes recognized as incorrect, depending on the usage in the sentence. For example, the word *form* used as *from*:

*I came **form** the store.*

One explanation for the spell checker's ability to identify erroneous words is due to the variation in the anticipated part of speech. In the above example, a past tense verb is immediately followed by a present tense word (came *form*). Statistically, this juxtaposition is

much more unlikely than a past tense verb immediately followed by a preposition (came *from*). Other real-word spelling error correction implementations will often make use of the target word's part of speech as a feature to detect potential candidates (Golding and Schabes 1996).

An unsupervised learning method utilizing lexical chains based on information retrieved from Wordnet (Miller 1995) relies on discourse related information and is in the minority of approaches that leverages intersentential features (Hirst and Budanitsky 2005). Bayesian statistical model has also been implemented (Golding 1995). Another approach describes a weight-update machine learning algorithm using a variant of the Winnow algorithm (Golding and Roth 1999).

Many real-word spell correction algorithms use a predefined list of words often referred to as a *confusion set*. These confusion sets provide a facility to pre-determine the most commonly misspelled words (Figure 5).

$$\begin{aligned}
 C_1: & \{c_{11}, \dots, c_{1n}\} \\
 C_2: & \{c_{21}, \dots, c_{2n}\} \\
 & \dots \\
 C_m: & \{c_{m1}, \dots, c_{mn}\} \\
 \forall i, j, k, l, i \neq k, j \neq l: & c_{ij} \cap c_{kl} = \emptyset
 \end{aligned}$$

Figure 5. Confusion sets

Examples of common sets include $\{to/too/two\}$, $\{their/there/they're\}$, and $\{affect/effect\}$. Much of the research on the confusion set approach has suffered from two

distinct weaknesses (Pedler 2007). First, many of these methods deal primarily with a small number of confusion sets, often 20 or fewer, limiting the potential of real-world applicability. Second, confusion sets suffer from a level of ambiguity. For example, the word *too* could be spelled as *top*, based on a typographical error of entering the *p* key rather than the *o* key. However, *top* could also be spelled as *tip*, a result of another common typographical error (*o* → *i*). Likewise, *tip* could be spelled as *rip* due to the proximity of the letters *r* and *t* to each other on a standard QWERTY keyboard. Entries in confusion sets are meant to be unique across all confusion sets, and this clearly poses a problem. Some approaches have chosen to dispense with the concept of confusion sets entirely, such as one that employs a pure language model (Mays et al. 1991), where the confusion sets were conceptualized as *confusion maps* (Figure 6). Confusion maps are data structure containing a target word acting as a source mapping to a confusion set.

$$S_1 \rightarrow \{c_{11}, c_{12}, \dots, c_{1n}\}$$

$$S_2 \rightarrow \{c_{21}, c_{22}, \dots, c_{2n}\}$$

...

$$S_m \rightarrow \{c_{m1}, c_{m2}, \dots, c_{mn}\}$$

Figure 6. Confusion maps

Possible alternatives automatically derived as any word that differed from the target word, in this case by only a single-letter edit. Although this method is initially thought as a significant transformation from handcrafted confusion sets and an improvement over the

Wordnet-based discourse method, it has been criticized as being prone to overcorrection, resulting in a loss of precision (Wilcox-O’Hearn et al. 2008). This method of precomputing alternative spellings is also used in other instances (Fossati and Di Eugenio 2008), which began investigating real-word spelling errors using a hybrid Hidden Markov Model and trigram language model approach. Language model approaches have also been used in very large scale applications integrating a string matching algorithm (Islam and Inkpen 2009).

2.4.2 Punctuation Elision

Punctuation elision has received much less attention in research circles. One possible reason for this might be because the problem does not present itself as wide-scale as other grammatical errors, particularly for ESL students. Unlike primary and middle school native novice writing counterparts, ESL students tend to have a much better understanding relating to sentential boundaries (Wade-Woolley and Siegel 1997). However, for native speaking novice writers, punctuation recovery is certainly relevant in certain groups. Missing punctuation may present issues with the syntactic analysis of input text. One example of an area requiring punctuation is Automatic Speech Recognition (ASR) output, typically devoid of punctuation. Consequently, it presents itself an ideal candidate for the sentence segmentation task (Mrozinski et al. 2006). Machine translation methods have also met with some success in addressing punctuation elision (Matusov et al. 2006).

2.4.3 Wordform Errors

A wordform error indicates that the writer has chosen the correct basic word, but the form of the word does not suit its position in the sentence. To better understand how word form errors occur in writing, it is helpful to review how verbs contribute to word forms.

In order to describe the nuances of an action, a verb may be associated with various concepts such as tense, aspect, voice, mood, person and number. Some languages, such as Chinese, do not inflect words but the concept is expressed by modifications of words in the context. In English, a verb can be inflected in five forms (Table 1). Incorrect usage of a verb inflection can be a symptom of two primary underlying categories of errors. These categories are generally speaking syntactic and semantic in nature. Each of these types is mentioned here.

Table 1. The verb eat and its inflections

Type	Example
Base (inf.)	(to) eat
Third person singular	Eats
Past	ate
-ing participle	Eating
-ed participle	Eaten

Work relating to the correction of verb forms has increased in recent years. One approach used parse tree comparisons coupled with language models (Lee and Seneff 2008).

Another approach tailored to ESL students used Conditional Random Fields with features from globalized disource information and compared these results against Maximum Entropy and SVM classifiers with only localized context and achieved moderate improvement (Tajiri et al. 2012).

Semantic Errors

The first type of wordform error relates to the overall semantic quality of the verb. Errors in this category reflect the inappropriate candidate of tense, aspect, voice, or mood. For example, the sentence: He runs there since yesterday. Either *has been running* or *had been running* are valid candidates for correction. Without examining information outside the sentence boundaries that would uncover more background on the overall temporal framework of the passage, disambiguating the possible choices is impossible. Another example is, *She is running in the marathon*. While the sentence is written correctly, assume that the intended sentence was meant to imply *She ran in the marathon*. Semantic analysis requires much deeper real-world knowledge of the surrounding context. These goals are not in the scope of automatic grammar correction approaches that rely exclusively on intrasentential features.

Syntactic errors

A second category of wordform error involves the misuse of verb forms. One of these is the incorrect subject verb agreement, where the verb is improperly inflected with respect to the subject of the phrase. An example of this is the sentence, "*They is going outside to play*",

corrected by either modifying the subject *They* to a third person singular *He/she* or altering the verb to its plural version *are*. In addition, writers may fail to provide proper agreement between auxiliary verbs and the main verb. An example of this is, *He has been run for years*, where the present perfect progressive tense, *running*, was intended, but the verb *run* was left in base form. Finally, the syntactic wordform error subcategory of improper complementation results from a verb complementing another verb or preposition. One such example is “The cat really wants play now.” In such an instance, the verb *play* would have to be modified to its infinitive form to complement the verb *wants*.

Gerunds/Part-of-Speech

Apart from semantic and syntactic verb misuse errors, other more subtle categories of wordform misuse may occur in novice writers’ text. One such subcategory is the overuse of gerunds. An example of this could be as follows: “*He was responsible for increasing incentives.*” While syntactically sound, the gerund *increasing* should be substituted with such phrases as *an increase in* or *the development of*. Other parts of speech confusions could also pose an issue for novice writers. A subset of such part of speech confused word forms are depicted in Table 2.

Table 2. Part of speech confused word forms

Type	Examples
Noun/Adjective	The veterinarian department added two new courses.
Adverbs/Adjectives	Please explain why vaccinations result in dramatically drops in disease?
Verbs/Nouns	Failing to pass a budget will cause the economy to stagnant.

2.5 Preposition Choice Errors

Errors relating to incorrect usage of prepositions are a common type of error, especially among ESL students. Approaches to address this challenge include using Maximum Entropy classifiers combined with rule-based filters for ESL students (Chodorow et al. 2007).

2.6 Article Choice Errors

Similar to preposition errors, errors relating to the incorrect usage of articles a/an/the are often seen as an extraneous form of error. Work done by De Felice and Pulman (De Felice and Pulman 2008) addressed both preposition choice errors as well as using classification-based methods (Knight and Chander 1994) and phrasal statistical machine translation techniques (Brockett et al. 2006).

2.7 Language Learning

Recent years have been met with seemingly unbounded demand to address the gulf between an increasing population of L2 learners and limited human resources available to effectively instruct them. To address this shortage, the development of Computer Assisted Language Learner (CALL) systems has proliferated. Some of these systems are rather rudimentary in nature, providing relatively static and universal grade learning using multiple choice and/or cloze question type exercises. Such an approach to designing L2 learning software has several rather glaring deficiencies. First, the method of a one-size-fits-all approach for such systems will often be met with of such systems, causing disengagement brought on by boredom and frustration. As a result, learning gains from such systems pale in comparison to much more customized forms of tutoring. An extension to CALL systems is Intelligent Computer Assisted Language Learner (ICALL) systems.

While the dividing line between CALL and ICALL systems is somewhat subjective, it is argued that a core competency of ICALL systems is providing the learner with a framework for providing open-ended input in addition to scaffolding the learning process with meaningful feedback of their input. Meaningful feedback could be conveyed by responses that create learning opportunities for students. For example, errors that are detailed linguistically enable students to remedy the error (Pusack 1983). However, the methodology for generating meaningful feedback is somewhat nuanced. Feedback could be generated as a result of manually mapping errors to feedback text. However, this approach poses a variety of problems, one of which is the scalability, especially when highly complex mappings are established in the system's linguistic knowledgebase. One alternative is to automatically

generate feedback based on sophisticated answer processing mechanisms, such as robust AI-based dialogue exchanging with the learner to provide grounding in the knowledge acquisition process.

Based on previous studies, the feedback mechanism for an ICALL system must also adhere to three key characteristics (Amaral and Meurers 2006, 2007). First, feedback needs to be highly accurate to the student. The feedback must be aggregated/filtered to consist of one key error message rather than multiple error messages, even if multiple errors exist in the students' input, the point being that the act of reporting multiple errors will eventually be ignored by the student. A filtering mechanism must be applied to identify the most relevant candidate errors to report. In order to satisfy this requirement, the grammar correction mechanism in an ICALL system must be able to incorporate student context, such as information gathered from a student model that maps known deficiencies from the student. Finally, the error message must be succinct and brief for it to be useful to the student.

While existing applications such as Microsoft Word include a grammar checker, some are not tailored toward learning grammatical correctness. While they sometimes provide recommendations for correction, they lack an integrated feature providing thorough explanations on recommended corrections. Without providing some sort of remediative explanation on the purpose of the fix, it is possible that novice writers may develop an overreliance on grammar correction over the long term.

Significant momentum has been made to inject increasingly robust artificial intelligence models into ICALL systems (Schulze 2010). Such systems include Robo-Sensei (Nagata 2003, 2009), a Japanese language tutor employing robust morphological and syntactic

parsing components to analyze sentences for grammatical errors coupled with a rule-based feedback generator to report on such types of errors as predicate form, missing/unexpected words, modifier and word ordering errors. Dickinson and Herring (2008) developed an ICALL system for Russian language learners that utilizes finite state automata for morphology-based error detection. In addition to standard writing exercise based language learning systems, ICALL systems have also incorporated game-based learning environments as an alternative, and is argued to provide a more immersive variant of language pedagogy for L2 learners (Johnson and Beal 2005). Systems utilizing student models to customize learners' ability to acquire language skills are presented in TAGARELA, an ICALL system supporting the instruction of Portuguese (Ott and Ziai 2010).

CHAPTER 3

Lexicographic Graph-Based Grammar Correction

An approach to automatic grammar detection and correction is through the global examination of candidate answers in the observed sentence via a graph representation. The candidates are analyzed via traversal of the entire frontier of subgraphs and producing the optimal hypothesis via probabilistic methods. In this chapter, an overview of graph-based approaches is provided, including defining the task and describing previous work using this methodology.

3.1 Task Definition

The goal of a graph-based approach in the context of automatic grammatical remediation of disfluent text is aimed at expanding the space of possible alternatives from an observed unit of discourse, typically mapped to sentence boundaries. This method of grammatical correction can be considered a recasting of the machine translation task (Bertoldi et al. 2010), in which a sentence containing broken text is translated to its correct form. However, one of the main caveats of examining grammar correction from a graph-based approach is the combinatorial explosion of correction candidates. For example, a sentence such as:

It was at mist two dollars.

Can be interpreted as having the following possible candidates:

It was at most two dollars

It was at mist two dollars

It was at must two dollars

It was at most to dollars

...

A brute-force approach to expand and explore the global set of alternatives is computationally intractable for any appreciable set of alternatives and/or length of sentence. Thus, a methodology must be devised to manage such a large set of possible alternatives to prune candidates not expected to attain a high likelihood probability in order to permit the efficient yet effective detection and correction of grammatical errors. Some of these heuristics are illustrated in the next few sections.

3.2 Over-Generate and Select Methods

Overgeneration and ranking approaches, also referred to as generate and select have become increasingly popular in recent years, as it allows a highly expressive method for Overgeneration/ranking has been particularly useful at addressing various NLP tasks such as natural language speech generation (Belz 2008; Langkilde-Geary 2002; Varges 2006), also

including question generation tasks (Heilman and N. A. Smith 2009). In automated grammar correction, Lee and Seneff (2006) utilizes an overgenerate/ranking approach based on a word lattice data structure, producing copious combinations of output, followed by a syntactic parsing rank scorer that reduces the large set of candidates to a maximum likelihood candidate.

3.3 Hidden Markov Models

HMMs are statistical Markov models that assume a Markov process with the additional characteristic of recording unobserved, or hidden, states. They form the trunk of another family of Bayesian networks referred to as dynamic Bayesian networks. Using the Viterbi algorithm (Forney 1973), observed inputs can be mapped back to the most likely sequence of state transitions.

HMMs are particularly useful in addressing tasks in NLP such as speech recognition, part-of-speech tagging, text summarization, handwriting recognition and information extraction. However, automatic grammar correction using HMMs have not been particularly popular as a computational model, though some effort has been made to leverage them in recent years. In the domain of real-word spelling error detection and correction, a mixed trigram approach incorporating HMMs with states representing Part-of-Speech and dictionary entry features (Fossati and Di Eugenio 2008).

3.4 Noisy Channel Model Theory

Noisy channel models (Shannon et al. 1949) are decoder-type frameworks that are tasked with finding an intended piece of information given a stream of scrambled input (Figure 7).

This defining feature makes them especially useful in problems such as machine translation, speech recognition, question answering, and spell checkers. The core element in a noisy channel model is the induction of a decision function. The induction of such a function can be approached from many different perspectives. Some methods include maximum likelihood rule, maximum *a posteriori* rule, and minimum distance rule (Jurafsky and J. H. Martin 2008; Manning and Schütze 1999).

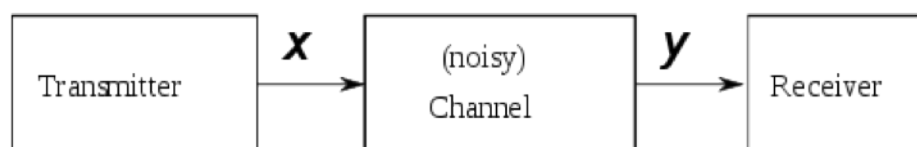


Figure 7. Noisy channel model architecture

Construction of noisy channel models is dependent on a variety of factors. First, probabilities are required to map substitution, insertion and deletion of base information, such as phonemes, letters, words, or other features. This information is typically crafted as a matrix that considers the conditional probability of the acts of substitution, insertion, or deletion. One of the key challenges is how these mappings are discovered. As often is the case, mappings are automatically computed using distance-based measures, such as Damerau-Levenshtein distance for spell checkers (Brill and Moore 2000) or HMM-distance based measures.

Noisy channel theory is used extensively in tasks involving machine translation. Conventionally, a corpus of parallel text is maintained mapping phrasal elements from L1 to

L2. A probabilistic decoding function is learned from this annotated data (Figure 8), and the transmitter and receiver are mapped to L1 and L2, respectively.

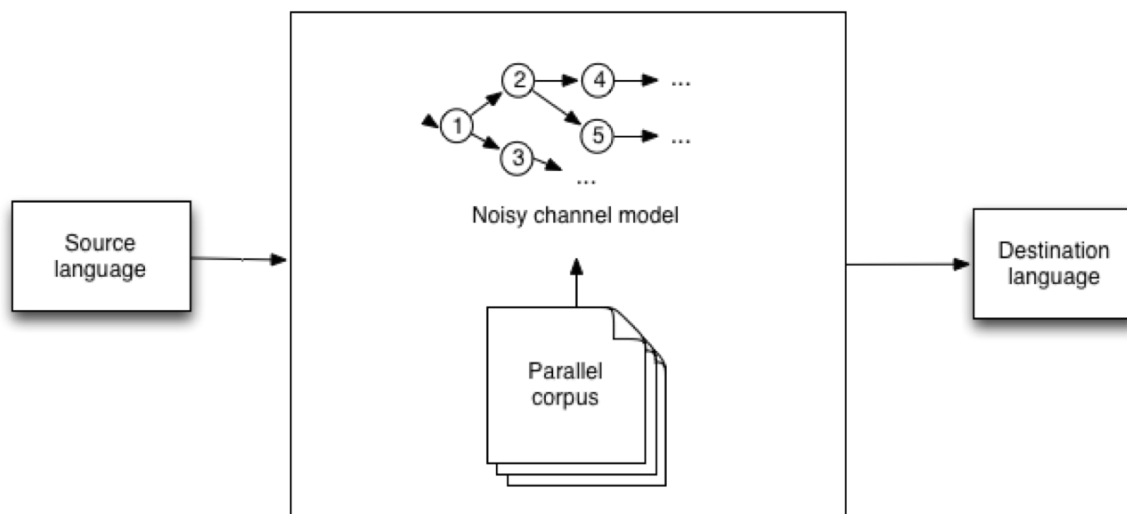


Figure 8. Noisy channel model for machine translation

3.5 Automated Grammar Correction using wFSTs

The task of whole sentence grammar correction can be approached efficiently using a noisy channel model and crafting the problem as a machine translation task (Bertoldi et al. 2010). To that end, it can be further postulated that lattices and directed graphs lend themselves handily as data structures for exploring the vast frontier of candidate substitutions in a given sentence. In short, noisy channel methods can be modified to be amenable to the task of grammar correction (Figure 9), where the source and destination language components are

replaced with the observed (presumably erroneous) text and the destination language is replaced with the original (error-free) text.

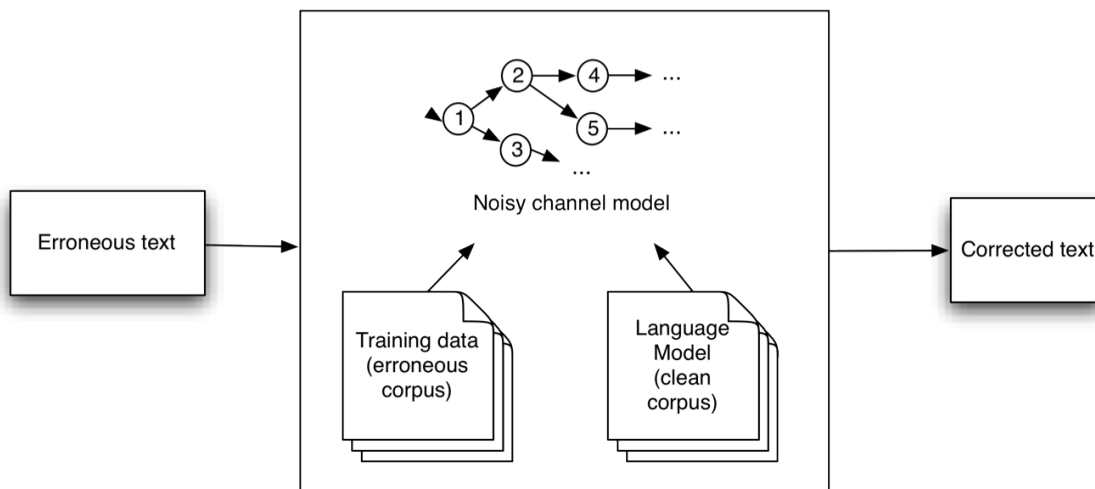


Figure 9. Noisy channel model adapted to automated grammar correction

To that end, Finite State Transducers (FSTs) are a type of finite state automata that contains the tuple $(Q, \Sigma, \Gamma, I, \Phi, \delta, \alpha)$, defined in Table 3. FSTs have key applications in speech recognition, optical character recognition, pattern matching, string processing, and information extraction.

Table 3. FST definitions

Tuple value	Definition
Q	States
Σ	Input alphabet
Γ	Output alphabet
I	Initial states
Φ	Final states
δ	Transition relation
α	Weight (cost)

FSTs are similar to finite state machines (FSM) in the sense that they, like FSMs, begin at an initial state, then transition states based on the acceptance of input. Also, like FSMs, an input exists in the FSM's grammar if the input results in a successful traversal to the final state. One of the key distinctions between FSMs and FSTs is the additional property of an output tape. This property is useful for storing properties such as lexical transformations occurring between state transitions. Additionally, FST arcs have the ability to store weight values (hence the term "weighted" FSTs), where each transition is labeled with a value representing the cost incurred during traversal. This weighting metric provides significant advantages. First, it allows for representing FSTs as probabilistic models (e.g., n-gram model or pronunciation model). This property provides an opportunity for FSTs to be utilized in machine learning applications, and for operations to be performed on them.

However, in order for operations to be well-defined, the weights must form a semiring (Golan 1999). FSTs often use the tropical semiring in practice, as they are most suited for determining the shortest path, as the tropical semiring examines the minimum distance across paths emanating from a source state.

An example of a simple wFST is shown in Figure 10. This FST unconditionally begins at state 1, and based on observing the available paths, we can determine that a partial list of accepted strings is $\{aba, bba, bcba, bccba\}$. The probability of accepting string aba and producing string zzy is represented as the conditional probability $P('zzy'|'aba')$, and is quantified with the value .00042, based on traversing the states $1 \rightarrow 3 \rightarrow 4 \rightarrow 5$, where the conditional probability consists of the joint probability of values $(.2)(.07)(.1)(.2)$, the first three values represent traversing to the final state. The final value represents the unconditional value of terminating at state 5.

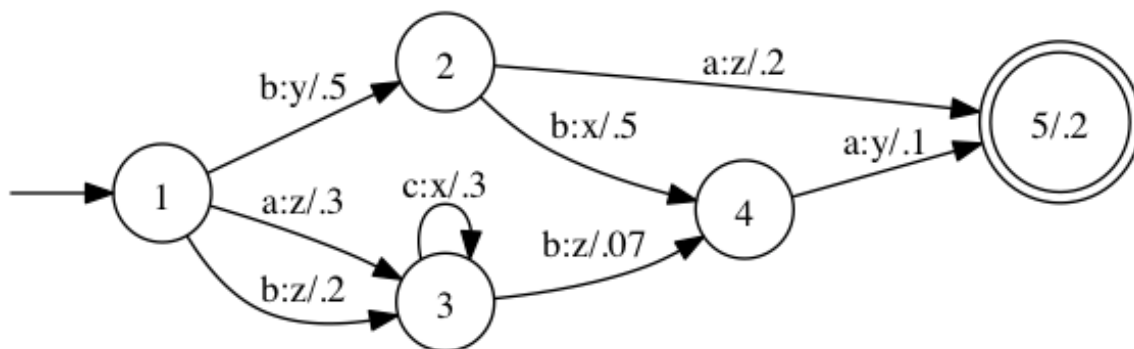


Figure 10. Weighted Finite State Transducer

Weighted Finite State Transducers (wFSTs) possess the capability of managing a vast frontier of alternate variations of a sentence's lexicographic structure, based on their graphical structure. Additionally, wFSTs can be combined. They have efficient algorithms for determining shortest distance to any state in the graph, calculus to cascade models onto other wFSTs, and flexibility in modes of calculus through the ability to support alternative weight types. As noted above, the majority of mainstream approaches using FSTs utilize what is known as the tropical semiring. However, other semirings exist, such as log, Boolean, Gallic, among others, can be used to model transition probabilities.

Finite state transducers have applications in several problem domains. In the domain of Automated Speech Recognition (ASR), wFSTs can associate input and output labels with phonemes along its paths (Mohri 2002). OCR post-processing noise reduction techniques combine character level wFSTs to automatically correct hypothesized errors (Llobet et al. 2010). Several finite state transducer implementations currently exist, such as Juicer targeted toward ASR implementations (D. Moore et al. 2006). However, one of the most widely used implementations is OpenFST (Allauzen et al. 2007), an open source weighted finite state transducer framework written in the C++ programming language. OpenFST was developed as a joint contribution between Google research and NYU's Courant Institute. It is also utilized in the implementation for this dissertation.

The implementation in this dissertation draws parallels from work inspired by Park and Levy's automated grammatical corrector (2011), in addition to spell correction frameworks (Pirinen and Lindén 2010; Pirinen and Silfverberg 2012). Park and Levy's work utilized a training corpus mined from the Web that contained TOEFL essays composed by Korean ESL

students. While the approach is useful for L2 learners, the dynamics of grammatical disfluencies among middle school students' writing differs from those of ESL students. Consequently, this approach was adapted for this user population, and has been extended in several key areas. First, the existing work in this area tunes the weights of the FST noise model that relate to spell corrections, but are only dependent on the length of the target word. For each word length n , each alternative spelling for an observed word has a uniform probability weight. This forces the algorithm to rely entirely on the disposition of the language model corpus in order to disambiguate alternative outcomes. Also, their work focused on spelling alternatives having an edit distance of 1 in Damareau-Levenshtein distance, but did not rely at all on other factors such as phonetic features nor an edit distance function that utilized the QWERTY keyboard layout as input. It is presumed that based on research done with middle school writers (Scardamalia and Bereiter 1987) that such features could improve a system implementing these metrics. Additionally, the existing work uses a bigram language model, which the work has admitted to being limited in its ability to provide sufficient contextual feature in several cases. Next, while other grammatical error types such as word form errors and misplaced articles are also examined, this does not encompass all of the possible errors middle school novice writers make. Lastly, this work proposes to investigate additional grammatical error types, since the previously mentioned approach looks at only three different error types.

CHAPTER 4

wFST Automatic Grammar Correction Framework

4.1 System Architecture

Automatic grammar correction using finite state transduction operates under the umbrella of a two-phase training procedure (Figure 11). First, grammatical text is pre-processed using morphology tools, then projected onto an n-gram model using language modeling software and smoothing techniques. Second, ungrammatical text is processed and combined with the language model in a training phase to converge weights of observed words to be substituted as incorrect entries. This chapter describes the process in detail.

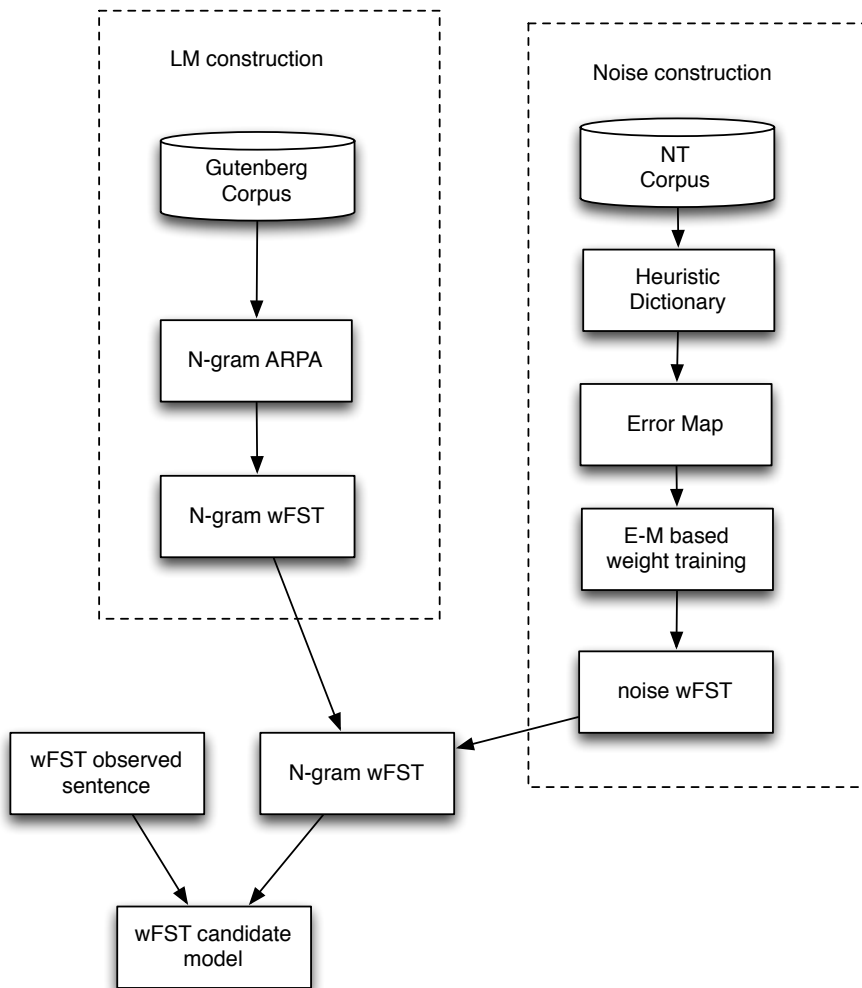


Figure 11. wFST system architecture

4.2 Language Model

Language models are used in many natural language processing applications such as speech recognition, machine translation, part-of-speech tagging, parsing, and information retrieval.

Language models help to capture the properties of a language, and serve as a mechanism for word prediction. This is accomplished by means of the probability

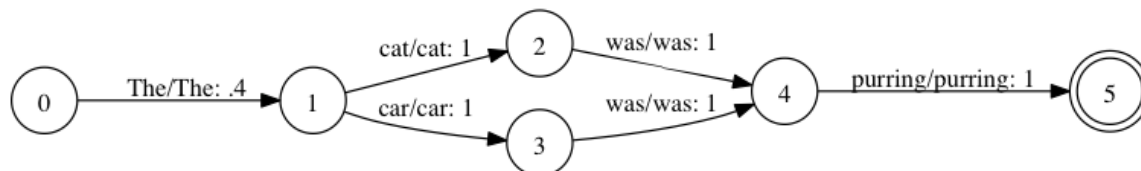


Figure 12. Bigram language model depicted as FST

distribution of encountering an unobserved word, based on the words immediately preceding it. Various tools have been developed to construct language models from a training corpus (Clarkson and Rosenfeld 1997). Language models are also referred to as n -gram models, where n represents the length of continuous grams, or words, the probabilistic model approximates. Typical n values for n -gram models range from 2 to 5, where larger values of n allow for more contextual information, but the amount of resources required for storing models is exponential to the value of n . Estimating the probability of all possible sequences occurring in unobserved data is unlikely even in very large corpora, since not all sequences will be encountered in cases where phrases or sentences are arbitrarily long. Thus, some sequences are not observed during the training of the language model. This is often referred to as data sparseness and may result in overfitting. Consequently, these models are often approximated using smoothing techniques such as Katz's backoff. (Katz 1987).

Language models are often produced in ARPA format, listing each n-gram as well as the probability of encountering such an n-gram. Likewise, language models can also be conceptualized as FSTs (Figure 12). The way this is accomplished is by representing the FST in such a way that each valid path represents a sentence in the training corpus. However, since an exact sentence in the training corpus is unlikely to be encountered in an unseen test instance, a fallback mechanism is usually implemented. This is accomplished through the use of special arc types (Table 4).

Table 4. Special arc types

Arc type	Consumes symbol?	Match “otherwise” condition?
ϕ	No	Yes
ϵ	No	No
σ	Yes	No
ρ	Yes	Yes

Some current FST n-gram generators exclusively utilize epsilon arcs to create a non-deterministic model. However, the tool selected for the construction of the language model in this research uses ϕ arcs. This special arc serves two purposes: first, it is used to designate back-off probability values for the purposes of smoothing on non-existing n-grams. Secondly, using ϕ arcs gives the FST the characteristic of determinism by reducing the need

for non-deterministic ϵ expansion. This feature reduces time and computational effort for exploring states during runtime. On the other hand, one of the weaknesses of ϕ arcs is the heavy restriction it places on certain operations, as only composition is permitted. Operations such as epsilon removal, determination, etc. are not compatible with FSTs containing ϕ arcs.

The statistical language model in this implementation is induced using the SRILM toolkit (Stolcke 2002; Stolcke et al. 2011). A trigram model is learned to balance the efficiency of shorter n-grams with the need for a reasonable window of localized context. Constructing a language model requires the selection of a smoothing algorithm when faced with sparse data (Chen and Goodman 1996). This becomes increasingly important relative to the n selected for constructing a language model. Modified Kneser-Ney smoothing was used for sparse n-gram context. Kneser-Ney provides advantages for cases where unigram probabilities should be discounted absent of preceding context. For example, *Francisco* as a unigram would be severely discounted absent of the lexeme *San* preceding it.

4.3 Noise Model

The purpose of a noise model is to quantify, in some manner, the likelihood of errors occurring in the writing process. There are several major challenges that arise in constructing a noise model. First, we must investigate the categories of errors that may occur in this particular problem domain. Second, based on these categories, a heuristic is required that captures the alternative tokens given an observed token. For example, it is well understood that real-word spelling errors may occur in writing. For example, given the word *lie*, it is possible this word could be misspelled as *lay*, *lye*, *lid*, *li*, *balh*.

In the implementation used for this dissertation, noise models are represented as an FST with one state (Figure 13). This single state serves as both the initial and final state, and all arcs emanating from the initial state loop back to it. The input token parameter for an arc represents an intended token, whereas the output token is one of two categories of values: the erroneous counterpart, or the same token, representing a non-error condition. A correct input token can be mapped to many different erroneous output tokens. Weights are *a priori* probabilities of the error occurring in text. For example, the act of misspelling the word *cat* as *car* may be expressed as a probability of .1, whereas correctly entering the word is represented as the inverse probability .9.

Unfortunately, there is a considerable obstacle concerning the weights in a noisy model: How are the weights derived? Although one may utilize approaches such as probabilistic evidence gathered from data that has been manually annotated with grammatical corrections, an alternative approach that is especially useful when provided with a sample corpus in the problem domain is inducing them from a corpus of erroneous text. To that end, training algorithms for FST weights has begun to be addressed very recently. Work has been done to unify the training process and directly estimate weights using unsupervised methods such as Expectation Maximization (Dempster et al. 1977; Eisner 2001; Wu 1983). This theoretical foundation also has been implemented using the expectation semiring (Dreyer et al. 2008; Jason Eisner 2002). Our model will incorporate a similar model, using an initial guess either informed by a heuristic or as a uniform probability across all possible alternatives. The unsupervised approach is especially advantageous for any sizable corpus as it eliminates the time and manpower required to train humans to correct and annotate errors.

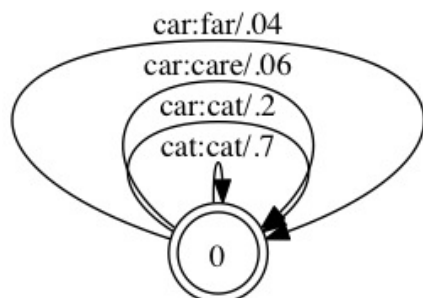


Figure 13. Noise model

4.4 Noise Model Induction

The FST noise model is induced from training data obtained from a corpus gathering study involving middle school novice writers described later. One of the key features of text from this source is that it contains information relating to grammatical errors made by middle school writers. To date, no large corpus exists publicly that closely parallels this particular domain. To learn the weights on each arc that maps to the probability of typing in that particular error, our implementation uses the EM algorithm as depicted by Eisner (2002). The algorithm is applied to our noise in a relatively straightforward manner. To begin, our model is initialized with uniform probabilities based on the number of calculated alternatives an observed word can take. This sets up the initial E step. Following this, we must determine all of the possible alternatives an observed sentence may take and calculate their probabilities relative to the sum probability of all paths. This is first accomplished through the construction of an *LM-noise* composition model, which is the result of performing the composition operation on the LM and noise model. Then, the LM-noise model is composed

with the observed sentence. The observed sentence is represented as a linear FST with states connected to each other with a probability of 1. This essentially generates all of the candidate versions of a sentence for the observed sentence. For this LM-noise-sentence FST, the expected probability of each path is efficiently calculated by pre-computing the shortest distance of each state to the final state in the log semiring. If the FST were thought of as an HMM, this effectively calculates α and β values at each state. Summing the arc between state i and j , α of state i , and β of state j , we obtain our expected value for a particular alternative. This procedure of obtaining the LM-noise-sentence FST is repeated for each sentence. At the conclusion of iterating all training sentences, the aggregate value of each alternative suggestion is divided by the total value for all alternatives for a particular observed token. This computation concludes the E step. The M step simply consists of the action of transferring these values into the next iteration, as these values represent the maximum likelihood estimate for each alternative (Figure 15). This process is repeated until the maximum change δ in any alternative is less than a convergence threshold ϵ (Wu 1983).

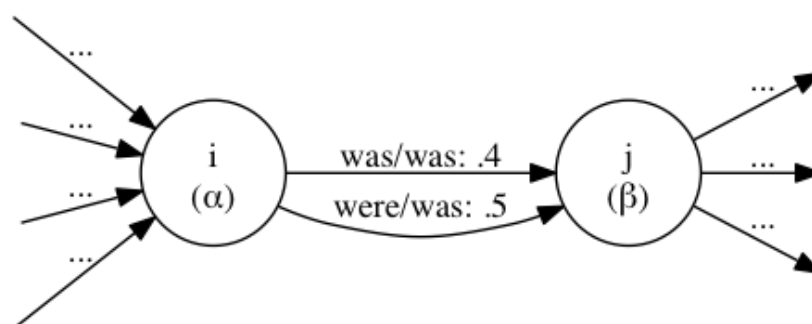


Figure 14. Determination of α and β

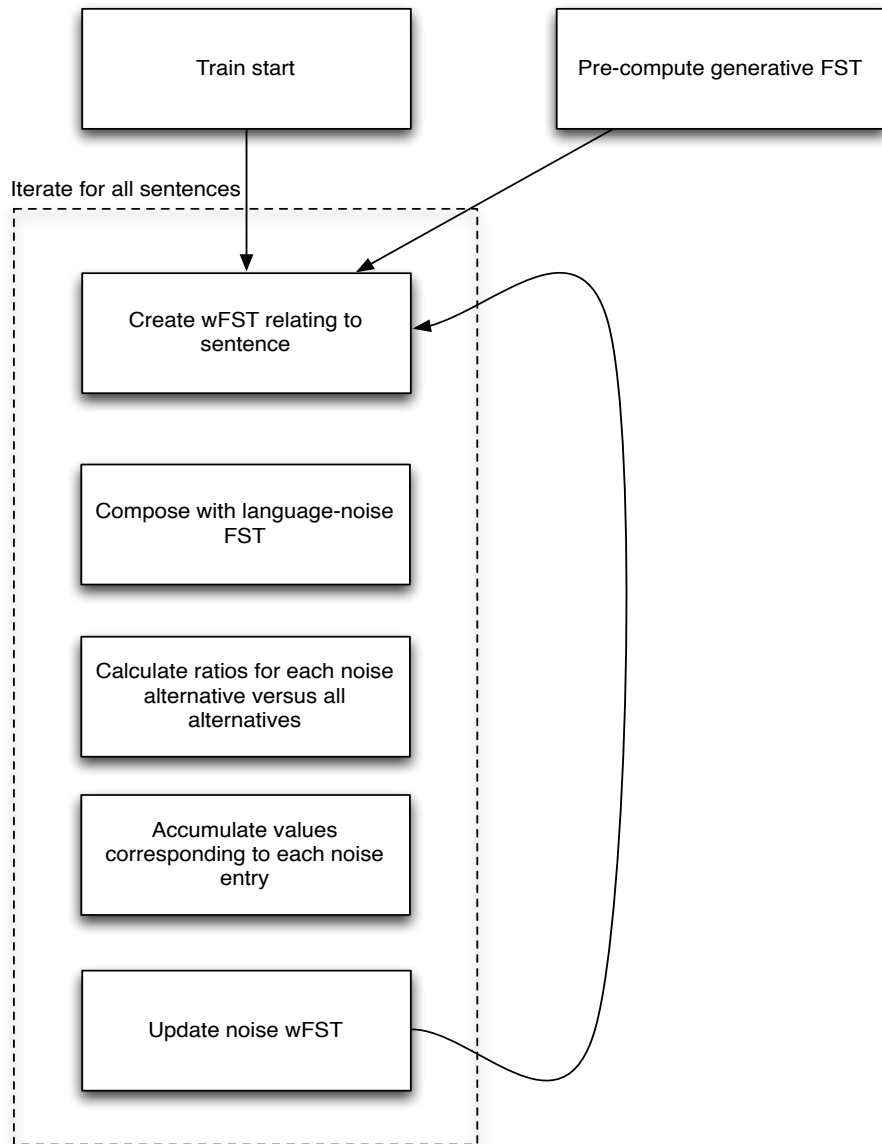


Figure 15. Training phase

4.5 Language-Noise Model

To compute all possible traversals for a token stream, a hybrid FST is created. This is accomplished by composing the language model onto the noise model. This is done twice, once in the training and once in the runtime phase. In training, each iteration of the E-M algorithm incorporates the composition of the LM and noise model as a set-up phase to iterate through training instances. At the conclusion of the iteration, the noise weights are updated with the new values and the process is repeated. This is done until the weights converge to within some ϵ value $< .001$. The Language-noise model is once again composed with the noise model at runtime, but with the finalized weights learned from the training step. This result of this composition is also referred to as the *language-noise generative* model (Figure 16). This generative model will be composed with a linear deterministic fst representing the training or test sentence.

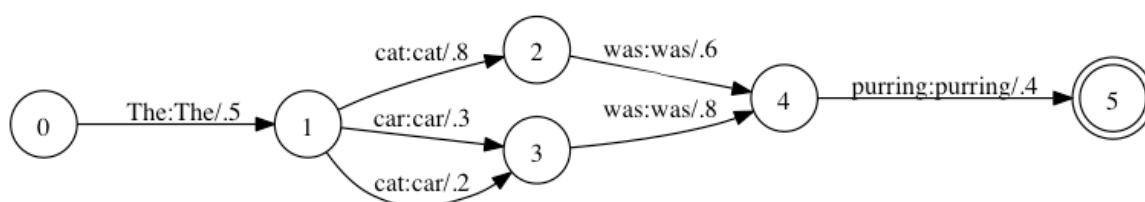


Figure 16. Language-noise generative model

4.6 Error Types

The proposed noisy channel model is made up of a collection of several smaller noise models. Each of these smaller noise models focuses on one particular grammatical error

category. This section enumerates the proposed error types and the heuristics used to generate alternatives.

4.6.1 Real-Word Spelling Errors

The real-word spelling error noise model implemented in this work accounts for spelling errors made by novice writers where the incorrect word is also a word in the dictionary. For spelling errors, the noise model implemented in this dissertation considers spelling errors of three categories, as illustrated in Table 3.

Table 3. Real-word spelling error types

Real-word spelling error type	Example
Edit distance (QWERTY)	car → {cat, vat}, buy → nut
Homophone	their → {there, they're}, rain → reign

First, a conventional edit distance metric is considered. This metric is referred to as the Damareau-Levenshtein measure, where the distance between two strings is the minimum number of single character insertions, substitutions and deletions and neighboring inversions. Damareau-Levenshtein has been utilized in various spell correction work in the past, including noisy channel models (Brill and R. C. Moore 2000) as well as intelligent web query searching (Cucerzan and Brill 2004). To maximize efficiency by constraining the number of candidates, the noise model generator will consider all words with a Damareau-Levenshtein edit distance of 1.

Next, a modified version of the edit distance metric is selected based on the edit distance of a character substitution, where the letter of a target word has an error in which one or more letters differ from the target word as mapped on the QWERTY keyboard. An example of this is the word *cat*. If this word was intended to be typed by the writer (i.e., fits in the context of the sentence) but was misspelled as *car*, it can be suggested that *car* may more likely be mapped to *cat* as opposed to other words with a Damareau-Levenshtein edit distance of 1, such as *vat*, *cay*, *hat*, etc.

Finally, the spelling error noise model contains homophones for each word. This is implemented by analyzing the edit distance based on a word's phonemes and including in a set all alternatives with an edit distance of 0. For example, the word *where* is made up of the phonemes: W EH1 R, and the word *ware* is made up of the same phonemes.

A popular method for providing phonetic coding is based on the Metaphone family of algorithms, such as Metaphone (Philips 1990), Double Metaphone (Philips 2000), and Metaphone 3 (Pande 2011). Metaphone and Double Metaphone have an approximately 89% accuracy rating whereas Metaphone 3 has a 98% accuracy rating. However, Metaphone 3 is not freely available. Consequently, the knowledge base of all relevant spellings and pronunciations are imported from a phoneme dictionary contained in the Sphinx-4 framework (Walker et al. 2004). This work considers words with substitutions that are at most one edit distance with relationship to the QWERTY keyboard. This measurement has also been seen use in search engine queries (Martins and Silva 2004).

4.6.2 Article Errors

The articles choice error noise model simulates the incorrect usage of the 3 articles in the English language (*a, an, the*). Examples of such errors include: *He ate a apple*, where the article *a* would be replaced with *an*. For the constructed noise model, there are 6 error weights ($3 * 2$) for the model.

4.6.3 Preposition Errors

The preposition choice error noise model simulates the incorrect composition of prepositions by a novice writer. For example, stating that a character in the story is *in* the barn would be correct, but in most convention cases, stating that they are *on* the barn would be incorrect. The goal is to incorporate the 12 most commonly misused prepositions by ESL writers (Gamon et al. 2009). Then, one parameter is specified for each for each preposition pair, similar to how this is done for the article choice error noise model. This gives 132 ($12 * 11$) parameters.

4.6.4 Pronominalization

The pronominalization error noise model will simulate the incorrect usage of pronouns. For example, sentences such as, *The cat and dog took his time*, would be corrected to, *The cat and dog took their time*. Since addressing pronominalization errors may require broader discourse related features than an n-gram model may provide, it is not anticipated that incorporation of this error model will dramatically increase performance.

4.6.5 Wordform Choice Error

The wordform choice error noise model captures the likelihood of using an incorrect word form of a word. This type of error can be caused either by choosing the incorrect tense of a verb (e.g., *have* → *had*), or by subject-verb disagreement (e.g., *he go* → *he went*). The 100 most commonly used verbs are used for this model, and the number of parameters for this error model is constrained to a maximum of 10 inflections per verb. This results in learning between 2 and 10 weights per verb to learn.

4.7 Maximum Likelihood Error-Free Text

To calculate the maximum likelihood sentence, a posterior probability calculation must be done based on the posterior probability calculation in Equation 1.

$$P(S_{original} | S_{observed}) = \frac{P(S_{observed} | S_{original}) P(S_{original})}{P(S_{observed})}$$

$$p(S_{original} | S_{observed}) = \operatorname{argmax}_y P(S_{observed} | S_y) P(S_y)$$

Equation 1. Error-free sentence determination

$S_{original}$ refers to the alternative sentence being examined, and $S_{observed}$ refers to the observed sentence. The conditional probability of the observed sentence given an alternative, $P(S_{observed} | S_{original})$ is calculated based on the LM-noise generative composed with the observed sentence, whereas the apriori probability of the alternative under investigation, $P(S_{original})$ is calculated by the composition of the LM and observed sentence model. This is

first accomplished by composing the language and noise models together, to create a language-noise model. This language-noise model contains the probabilities of alternative sentence path traversals, and can be used against an input sentence and a most likely correction candidate to be extracted. To do this, an input sentence must also be represented as an FST. This can be done by constructing a linear FST, where each arc contains a word in the sentence and has a transition probability of 1 (Figure 17).

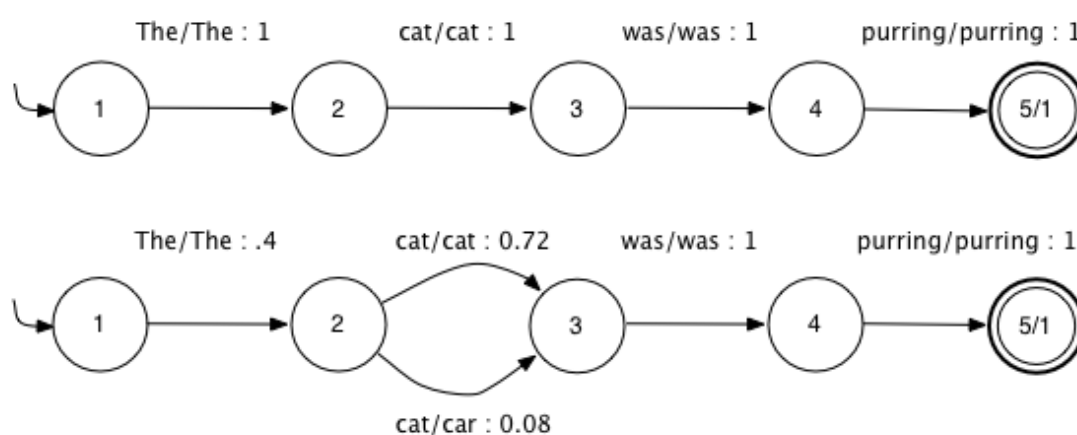


Figure 17. Observed sentence and sentence-language-noise FST



Figure 18. Example sentence MLE determination

Each sentence to be corrected is composed against the language-noise model. Performing this action results in an FST that shows all possible corrections for the original sentence. The remaining task is to traverse all of the paths and calculate the path with the minimum probability cost. Taking a more comprehensive example using the sentence *The fix run past him*, the shortest distance is determined using the tropical semiring. In this context, the tropical semiring is the joint probability calculated on edges from the starting state to the finish state, following the dotted lines the corrected sentence is written as *The fox ran past him*. (Figure 18).

4.8 Summary

This chapter described a weighted finite state transduction framework for automatic grammar correction. The goal of this approach is to express correction alternatives in a graph-based form, using probabilistic floating point values to represent transitions from source to destination states. The wFST framework consists of two primary models, a language model representing grammatical text, and a noise model representing the erroneous text. Using n-gram statistical algorithms, a trigram model is generated from the grammatical text. For the erroneous text, an unsupervised EM approach is employed to learn transition probabilities, representing the likelihood of an observed lexeme being mistakenly typed as an alternate lexeme. Example error types addressed by this system were introduced, and the runtime approach using the Naïve Bayes representation of the Maximum Likelihood Estimate was introduced in order to determine the best alternative grammatical sentence.

CHAPTER 5

Noise Model Corpus Collection

To investigate the effectiveness of noisy channel models in novice writers' text, a related corpus is required to train the language and noise model. While corpora exist relating to English as Second Language (ESL) writing, there has been substantially less effort in accumulating childrens' writing in comparison, though some examples of existing corpora are available (Pedler 2001). This section details the system used for the corpus acquisition process and outlines the study design and procedure.

5.1 Narrative Theatre

For the corpus acquisition studies, the NARRATIVE THEATRE (Baikadi et al. 2011) was utilized. The NARRATIVE THEATRE is a narrative-centered writing support environment developed for middle school students for the domain of sixth-grade language arts. It is built as a Flash™ web application. NARRATIVE THEATRE features a rich writing interface in the genre of fables where students can plan and write their stories. The NARRATIVE THEATRE is designed to enhance creativity for 6th grade language arts writers and a framework to provide adaptive writing support. The writing interface is a visual interface designed to lead the writers through the planning, writing, and revision process (Figure 19). First, the writers are asked to select story elements, such as setting and characters, to use in their story. After this

selection phase, writers are prompted to plan their stories by making an outline based on the standard three-act Aristotelian plot structure. After planning their fable according to this outline, they are then able to write their fable. During the writing process, all of their previous decisions are viewable from the writing window.

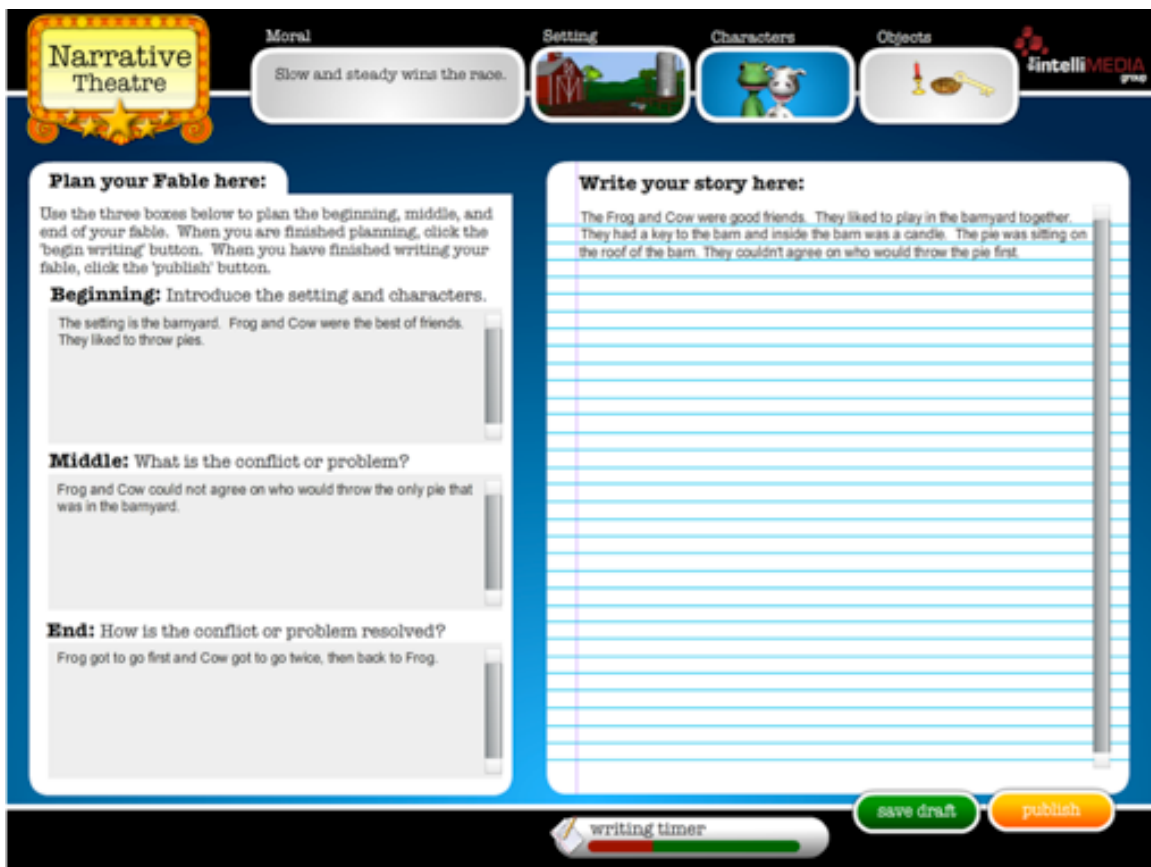


Figure 19. NARRATIVE THEATRE interface

5.2 Study Design and Procedure

In this section, the participants of the study as well as the design of the corpus acquisition experiment and procedure are outlined.

5.2.1 Participants

Students from a local middle school volunteered to participate in the corpus acquisition study. Students were selected from a local middle school, and were currently in the 6th grade taking a language arts class as part of their curriculum. Ages ranged from 10-13 years ($M=11.7$, $SD=.45$). 4 sessions per day were conducted, with each session yielding approximately 30-40 students, and each session lasting approximately 60 minutes. The students and were physically located in the same room, each provided with their own workstation for the entire session.

5.2.2 Procedure

The corpus collection activity spanned across two days for each student. Prior to the activity, students filled out a pre-survey during regular classroom time. On the first day, the students were seated at a computer and presented with a login screen to the Narrative Theatre. Upon logging in, students were asked to click a link that initiated the Narrative Theatre interface. This link initiated the NARRATIVE THEATRE interface. Upon opening the interface, students are asked to select a moral for their story. Students then select from this list of morals (e.g., *slow and steady wins the race*). Afterwards, students were provided with a choice of 4 different settings. Following this, students selected between 2 and 4 characters who participate in the story. Finally, they select between 1 and 5 objects that serve as props that

may be manipulated in the story. After making their selections, students are given 15 minutes to plan their fables, using a three-part outlining scheme encapsulating the beginning, middle and end of their stories. Following the completion of this planning phase or 15 minutes has elapsed, whichever comes first, students proceed to writing their fables. During the writing phase, students are allotted 25 minutes to compose their fable. The second day was spent entirely on having the students revised their existing passage. During both sessions, the students' actions were being recorded in an activity log. Activities such as keyboard presses, moral/setting/character/object selections, usage of the fable characteristic hints at the top of the screen, and timed snapshots were recorded. All personal data relating to the student were anonymized and stored in a MySQL DBMS along with the activity log table used for accumulating user actions.

5.3 Corpus Observations

The corpus acquisition activity yielded a sizable amount of data. In this section, some high-level corpus statistics are presented in addition to observations relating to the grammatical errors.

5.3.1 Corpus statistics

Statistics were gathered for the corpus, including subject information, quantity of lexical features and temporal information (Table 5). In total, 352 subjects participated, each contributing one passage. In total, the students wrote 4,281 sentences (M=12.16, SD=4.2) and 26,174 words (M=74.36, SD=6.8). The total elapsed time for planning was just over 15 minutes (M=15.23, SD=3.4) and writing time contributed just under 25 minutes (M=24.74, SD=7.2).

Table 5. Narrative Theatre corpus acquisition statistics

Type	Amount
Subjects	352
Passages	352
Sentences	4,281
Words	26,174
Time elapsed writing (min)	24.74
Time elapsed planning (min)	15.23

5.3.2 Corpus errors

The corpus gathering experiment resulted in stories with a wide spectrum of grammatical errors (Goth et al. 2010). These error types vary greatly and were rooted in student characteristics such as demographics and self-efficacy (Table 6). 2,000 sentences were randomly selected for evaluation from a crowdsourcing task described in the next chapter of this dissertation. From this collection of records, 5,812 grammatical errors ($M=2.9$, $SD=.72$) were encountered. The most common type of grammatical error made by participants was real-word spelling errors, where 3,118 were encountered. Next, 1,282 wordform errors were made, the majority by auxiliary verb misuse. 819 incorrect prepositions were input, 413 article errors were produced, and 180 pronominalization errors were introduced (Figure 20). Several errors were not caught by our heuristics. One of the most predominant types of errors were real-word spelling errors that fall outside the edit distance requirement. Examples of this were words such as *played* typed as *plaid*. Novice writers may phonetically process the word incorrectly and this is consistent with current literature (Scardamalia and Bereiter 1987).

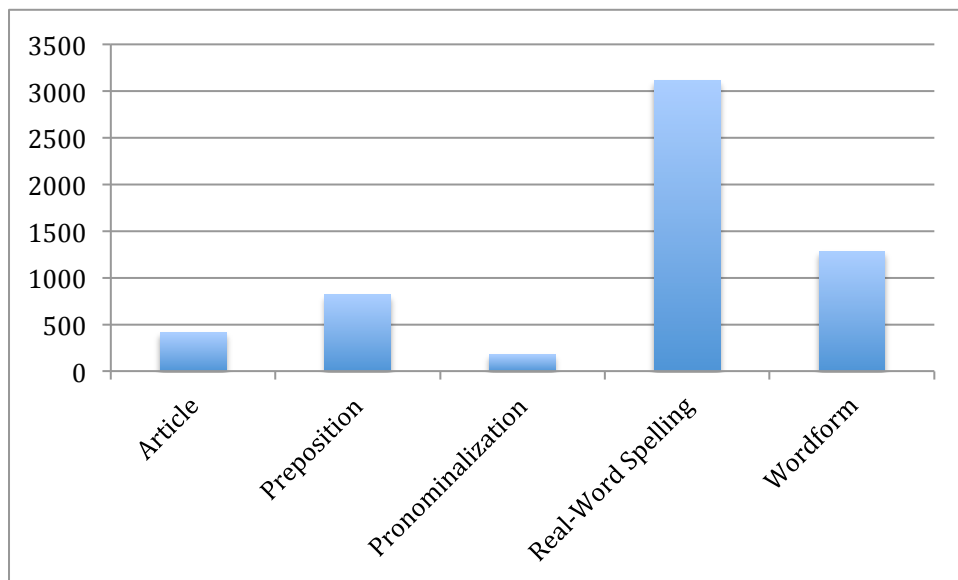


Figure 20. Grammatical error occurrence frequency

Table 6. Grammar error categories

Error	Example
Subject-verb agreement	...and the dog, frog, and rabbit they <i>was</i> sooooo happy for the lion...
Real-word spelling error	suddenly she <i>herd</i> foot steps
Quotation elision	She said [“]it was not shiny enough[. ”]
Word form error	after all the fun they <i>want</i> to go back home to the farm.
Article elision	Once there was [<i>a</i>] lion that always wanted to go out on a journey to find gold.
Question mark elision	the only way to remeber is to look in the mirror and see your real face". said te owl. can you reverse it[?]"
Multiple errors	they begin <i>there</i> walk in the pasture and <i>pick</i> flowers.

A noisy channel model is an ideal approach for several of these error types. Errors such as article elision/substitution and word form errors involve lexical substitution and is relatively trivial to conceptualize using a noisy channel model. However, quotation elision is quite a bit more challenging, since quotations represent lexical cues marking a text region as dialogue as opposed to an isolated lexical substitution, thus it is outside the scope of this work.

CHAPTER 6

Evaluation

In this dissertation, an implemented prototype for an automatic grammar detection and correction framework was introduced, which integrates a language model of clean grammatical text and a learned noise model for modeling different categories of grammatical errors. The runtime portion of the correction framework begins its process by receiving noisy or ungrammatical text. The sentences in a passage are iterated, and each sentence is composed against a generative model combining the language and learned noise model. This chapter evaluates the performance of the runtime component. The performance of the runtime component is evaluated with widely used automatic metrics: recall, precision, and F1 measure, which compare the answers predicted by a model to the gold-standard answers. The quality of sentential level grammatical correction was judged by human subjects.

6.1 Methodology

Two automated metrics that are widely used to evaluate the performance of natural language processing components are recall and precision. Recall is a metric that reflects the portion of labeled constituents that are correctly identified by a system with respect to the total number of labeled constituents in the gold standard answer. Labeled precision measures the portion of correct labels with respect to the total number of labels predicted by a system.

Recall and precision are usually considered together because, in an extreme case, one can be exaggerated at the cost of the other. For example, a very conservative system that makes predictions only when its confidence level is above 90% will achieve high precision but its recall will be likely very low. On the contrary, a very relaxed system that makes duplicate predictions with different labels for given candidate constituents could achieve high recall but the precision will be low. F1 measure, also known as balanced F-score, combines these two metrics. It is the harmonic mean of recall and precision (Figure 21).

$$R = \frac{tp}{tp + fn}$$

$$P = \frac{tp}{tp + fp}$$

$$F = \frac{2 \times P \times R}{P + R}$$

Figure 21. Recall, precision and F-measure

BLEU scores (Papineni et al. 2002) are often used to evaluate the effectiveness of machine translate tasks. NIST (Doddington 2002) modifies the BLEU score slightly to include information content in the calculation, to weigh less used terms greater in score generation. METEOR scores (Banerjee and Lavie 2005) presents an alternative version of these scores that addresses some of the limitations that may cause lower human agreement scores using BLEU or NIST. Methodologies also exist to compare and contrast these scores (Koehn 2004; Zhang et al. 2004).

The evolution of methodologies designed to evaluate automated grammatical correction systems is still in its early stages. Some approaches include human annotators making blind qualitative comparisons between an automatically corrected sentence and a human corrected version of the text (Lee and Seneff 2006). Additionally, a rather novel approach was taken using machine translation metrics such as BLEU, NIST, and METEOR (Park and Levy 2011). The usage of these scores has also been reported as correlating well with human judgment. However, while these scores assess the overall improvement in quality of transliterated text as a whole and can incorporate term frequency, synonymy and stemming to better approximate the score, they can fall short in assessing improved semantic quality of the transformation. Since the goal of our system is to improve the semantic quality of our output when analyzed with a semantic role labeller, a different approach is necessitated.

To help address this challenge, our evaluation technique utilizes the CoNLL 2004 shared task evaluation software to assess overall semantic quality (Carreras and Màrquez 2005). The original intent for the challenge was to quantitatively evaluate semantic role labelling software. For this analysis, we use it to evaluate how closely our post-processed text aligns semantically with human corrected text. The evaluation software measures precision, recall and F-measure. Evaluation of recall involves the successful identification of semantic roles, whereas precision measures the accuracy of the identified semantic roles (Figure 22). In the first errored instance, the semantic role A0 is successfully identified. However, the second semantic role label is incorrectly identified as a MNR (manner) rather than LOC (location). The second erroneous example, due to the incorrectly spelled intended determiner,

underestimates the length of the A0 role but correctly identifies the LOC role. An incorrectly tensed verb in the third example is the catalyst for incorrectly labeling the A0 role as A1, and the incorrect spelling of the verb *ran* as *fan* prevents semantic role labeling from occurring in this particular sentence.

Gold standard semantic role labeling (2 of 2)	<table border="0"> <tr> <td>A0</td> <td>V</td> <td>AM-LOC</td> </tr> <tr> <td>The quick fox</td> <td>ran</td> <td>near the dog</td> </tr> </table>	A0	V	AM-LOC	The quick fox	ran	near the dog						
A0	V	AM-LOC											
The quick fox	ran	near the dog											
Incorrectly spelled word <i>fox</i> , mislabeled semantic role (1 of 2)	<table border="0"> <tr> <td>A0</td> <td>V</td> <td>AM-MNR</td> </tr> <tr> <td>The quick fix</td> <td>ran</td> <td>near the dog</td> </tr> </table>	A0	V	AM-MNR	The quick fix	ran	near the dog						
A0	V	AM-MNR											
The quick fix	ran	near the dog											
Incorrectly spelled word <i>The</i> , non-aligned semantic role (1 of 2)	<table border="0"> <tr> <td></td> <td>A0</td> <td>V</td> <td>AM-LOC</td> </tr> <tr> <td>Then</td> <td>quick fox</td> <td>ran</td> <td>near the dog</td> </tr> </table>		A0	V	AM-LOC	Then	quick fox	ran	near the dog				
	A0	V	AM-LOC										
Then	quick fox	ran	near the dog										
Incorrect verb tense of <i>ran</i> , Wrong semantic role found (1 of 2)	<table border="0"> <tr> <td>A1</td> <td>V</td> <td>AM-LOC</td> </tr> <tr> <td>The quick fox</td> <td>run</td> <td>near the dog</td> </tr> </table>	A1	V	AM-LOC	The quick fox	run	near the dog						
A1	V	AM-LOC											
The quick fox	run	near the dog											
Incorrectly spelled verb <i>ran</i> , no semantic roles found (0 of 2)	<table border="0"> <tr> <td></td> <td></td> <td>V</td> <td></td> </tr> <tr> <td></td> <td></td> <td>?</td> <td></td> </tr> <tr> <td>The quick fox</td> <td>fan</td> <td>near the dog</td> <td></td> </tr> </table>			V				?		The quick fox	fan	near the dog	
		V											
		?											
The quick fox	fan	near the dog											

Figure 22. SRL performance scoring example

To evaluate our data, we took resulting output from our FST system and compared the SRL frames to human annotated results from crowdsourcing (described below). We compared the SRL frames to those generated by output when manually pasted into both Google Docs and the UIUC context-sensitive spelling demonstration system,² Both of these systems not only offer alternative suggestions for some real-word spelling errors, but they also have the ability to offer correction candidates for various wordform errors.

² <http://cogcomp.cs.illinois.edu/demo/cssc/?id=11>

6.2 Data Set

For test data, we selected a total of 2,000 sentences from our corpus. The sentences were divided into 10 folds for purposes of bootstrapping a cross-validation test. The FST noise model was trained on 90% of the data and observed sentences for testing were iterated on the remaining 10%. To establish a gold standard sentence structure, a crowdsourcing task was initiated (Figure 23). A Human Intelligent Task (HIT) was constructed on Amazon Mechanical Turk (AMT). One of the key strengths of AMT is the ability for it to provide quality data in the form of annotations, question answering, etc. (Buhrmester et al. 2011). Human annotators were tasked with manually correcting a randomly chosen sentence in the dataset. Turkers were given 5 minutes to accomplish this task. In addition, Turkers were asked to respond to four survey type questions relating to the sentence, based on a 1-5 Likert scale. These questions gauged the user's qualitative analysis as to the semantic quality of the sentence prior to correction, a self-efficacy value relating to the level of improvement based on their correction, the post-correction semantic quality of the sentence (since corrections followed a strict protocol to only correct sentences based on the category types our system is designed to handle), and the overall difficulty of the task (Table 7).

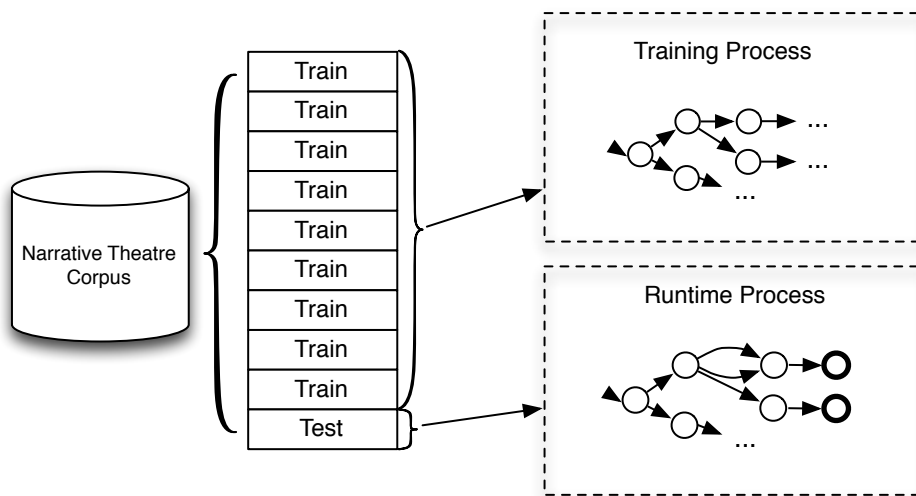


Figure 23. Train/Test cross-validation methodology

Table 7. Questionnaire results

Question Type	Mean/StDev	Weighted Kappa
1) Semantic Quality prior to correction	2.67 (1.44)	.62
2) Improvement realized with corrections	2.56 (1.51)	.71
3) Semantic Quality after correction	2.11 (1.21)	.58
4) Difficulty of task	2.03 (1.20)	.53

6.3 Results

In this section, results for the semantic role processing task as well as the runtime speed will be presented.

6.3.1 Cross-annotator performance

To evaluate overall annotator quality, their consistency for gauging the task as well as self-assessed performance in correcting sentences was measured using a weighted Kappan inter-rater agreement measure (Cohen 1968). This metric allows for measuring agreement where codes are ordered. In this scenario, the ordinal nature of the Likert-scale set of responses has this property. Based on the results (Table 7), weighted Kappa scores across annotators were considered acceptable for all four questions, ranging from .53 to .71. These results provided evidence that Turkers moderately agreed on pre-correction quality, improvement, post-correction quality and difficult of task.

6.3.2 SRL-based performance

To evaluate SRL based performance, this implementation involved comparing post-SRL output using the SENNA framework (Collobert et al. 2011), a highly efficient state-of-the-art neural network induced semantic role labeler. For comparison, we use McNemar's Chi-square test for paired samples as described in (Japkowicz and Shah 2011). Our McNemar test revealed that our FST system had statistically significant precision and recall when compared with the UIUC system, with $p < .05$. While the FST system outperforms Google Docs, the difference in performance is weakly statistically significant with $p < .1$ (Table 8).

Consequently, we examined our corpus from another angle. Our next experiment was to determine how well our approach performed against some of the more grammatically disfluent text. Instead of examining the data as a whole, we analyzed the data that had below

average semantic quality. This was accomplished by filtering our evaluation dataset to only include results from sentences that were rated 1 or 2 on the 1-5 scale for original semantic quality, which is Question 1 in the questionnaire.

Based on the results from this analysis, we hypothesize that our FST framework would be able to perform more efficiently than the two competing grammatical systems we are comparing it to based on two factors. One possibility is that the in-domain nature of the corpus inducing our language and noise models was a factor.

Table 8. Results of all sentences

System	Precision	Recall	F-measure
FST	83.6	48.49	61.38
Google Docs	73.5	50.15	59.61
UIUC	71.0	42.72	53.34

Following the adjustment to account for semantic quality, the results of this analysis when compared against the Google Docs built-in grammar corrector in addition to the online version of the University of Illinois context-sensitive system, which also detects other types of grammatical errors such as subject/verb agreement.

Our McNemar Chi-square for paired samples test revealed that our wFST system had statistically significant precision and recall when compared with both Google Docs and the UIUC system, with $p < .05$ (Table 9).

Table 9. Results of sentences with below average semantic quality

System	Precision	Recall	F-measure
FST	81.38	56.21	66.49
Google Docs	72.79	47.1	57.19
UIUC	54.87	36.0	43.48

6.3.3 Runtime Speed

Our next test was to evaluate and compare our system based on time performance. To perform this analysis, first a subset of 100 records was randomly selected from the full population of 2,000 records. For each sentence, the time to interpret the grammatical quality of the sentence and provide all alternative outcomes in seconds was observed and recorded (Table 10). This was relatively straightforward for the wFST system. However, for the two compared systems, an estimate was provided based on the cumulative duration of iterating the two-step cycle of the system detecting an error, then choosing the highest confidence suggestion. If additional suggestions were offered based on re-evaluating the sentence post-correction, the total time was added on.

Table 10. Runtime speed

System	Average time (in seconds)
FST	4.3 (SD = 1.7)
Google Docs	5.1 (SD = .8)
UIUC	8.7 (SD = 2.1)

6.4 Discussion

Our results showed that for certain types of error correction types, our system varied in performance. Real-word spelling error correction played a central part in improving the intended semantic role labelling structure of a sentence. Article corrections were made on several occasions, but did not contribute much to evaluated results. One reason for this was due to the SRL system being somewhat agnostic to article choice. Whether a sentence read *Rabbit threw a apple* or *Rabbit threw an apple* did not affect the extraction of the noun phrase encompassing the word *apple*. This is consistent with how many conventional SRL algorithms utilize syntactic features to identify semantic roles (Pradhan et al. 2004; Punyakanok et al. 2005; Xue and Palmer 2004). Additionally, the correct article choice often depended largely on discourse related information. Preposition choice corrections had a marginal outcome in contribution. However, the correct choice of article and preposition can play an important role in how a downstream component interprets the final knowledge representation.

From a runtime speed perspective, the primary factor that corresponded to slower computation time was the number of states and transitions produced from the noise-language generative FST model. The number of states is largely influenced by the topology of the language model. Projection into the noise-language FST model is relatively fast due to optimizations in OpenFST for larger models. However, traversal of the model results in a combinatorial expansion as each possible alternative is explored. Pruning provides some recourse for optimization, as edges are not explored if doing so exceeds the cached optimal probabilistic alternative. Still, certain types of input may degrade runtime performance if the

edge count is very high. Consider the erroneous sentence: *She went too ad won more thing*. Here, the tokens *too*, *ad*, and *won* are spelled incorrectly. These words are highly ambiguous in terms of possible alternatives that satisfy the requirements in our system. For example, the observed word *ad* could be alternatively spelled *sad*, *tad*, *rad*, *add*, *ads*, *aid*, etc. The other two words also yield a high number of alternate possible outcomes (Figure 24). On the other hand, some words will produce much fewer alternatives. Longer length words with less common phoneme and letter ordering, such as *influence*, *prepare*, *tomorrow* would have many fewer grammatical alternatives and be less of a contributor in longer runtime processing. As sentence size increases, the likelihood of encountering lexemes producing a higher combination of alternatives increases.

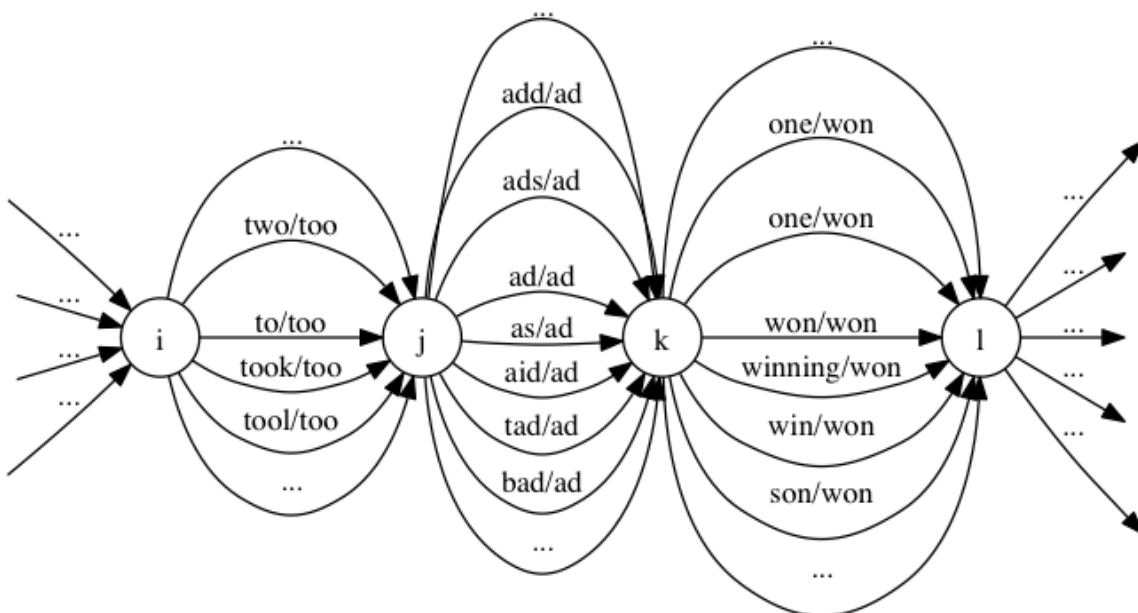


Figure 24. Highly ambiguous input

Where our system appears to be more successful in correcting grammatical errors in comparison to other systems in our evaluation were in cases where the observed sentence exhibited a high degree of grammatical disfluency. This seems reasonable, as many existing algorithms for grammatical correction tend to utilize contextual cues from words existing within a window. It stands to reason that if the words within this contextual window are also incorrect, the features used as input to these algorithms would be unreliable and may negatively affect performance of the grammatical detection and correction system.

Since our system performs well under conditions of highly disfluent text, its usage in areas such as hurriedly composed social network data (B. Han and Baldwin 2011), in-game chat systems (Huffaker et al. 2009), etc. are particularly interesting. In addition, transforming disfluent text to better organized and structure text while retaining the intended meaning allows for the use of existing syntactic and semantic parsers.

Table 11. Example sentences shown to crowdsourcing annotators

Input	the loin haired a guy to spun fire rings for the party
Corrected	the lion hired a guy to spin fire rings for the party
Input	sheep was sacred because she was so shy
Corrected	sheep was scared because she was so shy
Input	when they got ready to go home they got lost in the forest
Corrected	when they got ready to go home they got lost in the forest

6.5 Summary of Chapter

The performance of the automatic grammar detection and correction system using weighted Finite State Transduction was evaluated. The semantic quality of the corrected sentences were evaluated using three automated metrics, recall, precision, and F1, which compare the corrected sentences with the crowdsourced annotations as gold standard. Human judges were cross-evaluated to compare their relative understanding of the tasks and their shared ability to correct sentences in a similar fashion.

To evaluate the finite state transduction framework, two state-of-the-art online systems were used: the grammar correction feature built into Google Docs, a proprietary system featuring contextual spelling and grammar checking, and the online contextual spelling and grammar checker available from UIUC. The performance of the wFST was examined within two dimensions: semantic quality suitable for Semantic Role Labeling, and runtime speed. In the context of SRL quality, the wFST system was statistically significantly better than the UIUC system and weakly statistically significant than the Google Docs system. Limiting the scope of the analysis by examining only sentences that were more poorly structured based on annotator questionnaire responses, the wFST model was once again compared to the two competing models, in which the wFST model achieved statistically significant results compared to both UIUC and Google Docs. For runtime speed, the systems were compared based on the average time required to compute the best grammatical alternative offered. The wFST system performed statistically significantly better than the UIUC system and at a runtime speed comparable with the Google Docs system. In

short, the wFST model achieved state-of-the-art results for producing text more suitable for semantic processing.

CHAPTER 7

Conclusion

While most successful work in extracting semantics representation from natural language has focused on relatively clean and grammatical text, many domains exist where this high standard is not guaranteed. On the contrary, the relative quality in domains such as middle school writers, second language learners, and various forms of idiomatically inspired shorthand.

We present a useful method of identifying and correcting ungrammatical text made by novice writers. One primary benefit to this approach is the ability to use an unsupervised method to train FST noise weights. This allows for the collection of data without requiring the cost of time and money to train human annotators. This can be especially useful in cases where one desires to accumulate large quantities of text for training. This feature is particularly valuable given the copious amounts of unannotated text available in various domains containing ungrammatical text.

For evaluation, we have included a novel approach involving the comparison of semantic frame output from textual input. This evaluation approach has the added benefit for providing a preview into how effective our grammatical corrections will be realized in a system that relies on clean, grammatically sound text as input into an NLP pipeline. It may

also provide a framework for future evaluation efforts into observing not only the overall grammatical improvement, but to see in detail how relevant the improvements are to the particular application.

This framework has application in several domains outside of narrative writing. For example, social media text contains text that varies greatly in quality of grammar. Likewise, SMS messaging, in-game chat sessions, and other forms of electronic shorthand can vary greatly in quality of grammar.

Grammatical detection and correction can play a significant role in improving overall semantic quality of ungrammatical text. Our analysis showed that that framework could perform better than other state-of-the-art spelling and grammar correction tools. This was especially the case for text ranked by human annotators as below average in semantic quality.

7.1 Hypotheses Revisited

The research presented in this dissertation has produced evidence that supports the following hypotheses.

- Hypothesis 1: Human judgement via a crowdsourcing task provides a sufficiently viable benchmark for our analysis.
 - A human annotation exercise was constructed and distributed online. This crowdsourcing task required users to correct sentences and supplement their corrections with Likert scale ratings on both the pre-

and post-correction versions of the sentences. These scale ratings were cross-checked using a weighted Kappa metric.

- Hypothesis 2: The implemented noisy channel implementation provides significantly improved accuracy in relation to the grammar detection and correction facilities of two state-of-the-art systems. One of these systems is the grammar detection facility of Google Docs, a web-based word processor system. The other is the online version of the Winnow context-sensitive spelling corrector.
 - Evaluation of both Google Docs and UIUC's system was facilitated by manually pasting each test sentence into the framework, then running the Semantic Role Labeling framework against the output. Using the CoNLL-2005 shared task as a guideline, each system's SRL output was tested against the SRL output from the annotation produced as a result of the crowdsourcing task. For observed sentences that were rated below average in semantic quality, the wFST framework outperformed both evaluated systems.
- Hypothesis 3: The implemented system provides output for the automated grammar correction task in a bounded time limit comparable to both Google Docs and UIUC's Winnow based context-sensitive spelling corrector.
 - Evaluation of both Google Docs and UIUC's system based on runtime speed was facilitated by approximating the average runtime performance for correcting sentences, then comparing that time to the wFST framework.

For hypothesis 1, weighted Kappa scores showed moderately favorable results for rating quality prior to and following correction, as well as overall difficulty of the task. There may be several ways to improve this score. Compensation for Turkers could be increased as well as limiting participants to those who have higher approval ratings (Kittur et al. 2008).

Evaluating hypothesis 2 involved comparing precision and recall of semantic role labeling of the proposed wFST system against the grammatical correction process of the Google Docs application as well as the online demonstration version of UIUC's grammar correction framework. Results showed that while the wFST framework did produce a statistically significant improvement compared to the UIUC system, it did not show the same significant improvement compared to Google Docs. Limiting the pool of responses to only those where pre-correction quality was below average (1-2 on 1-5 Likert scale), re-running the evaluation showed a statistically significant improvement for the wFST system when compared against both UIUC and Google Docs systems. One possible conclusion to this result is that sentences with poor grammatical structure likely resulting in very poor semantic quality were more effectively evaluated by the wFST system versus Google Docs and UIUC, possibly due to the difficulty of Google Docs and UIUC in handling errors where localized features are unreliable.

Evaluating hypothesis 3 proved to be interesting from the perspective of evaluating runtime performance of both the UIUC and Google Docs system, as there was no machine recorded time available for either system and the overall time to execute had to be manually recorded and compared. Based on a subset of this data, the wFST system was statistically

significantly faster in computing the most favorable alternative. One cause of this result was the cascading nature of the other two systems. In cases where a grammatical error was discovered and alternatives were presented, selecting an alternative required the system to retry the sentence with the updated alternative. The accumulated time for sentences, especially sentences with multiple errors, contributed to an overall increased execution time.

7.2 Summary

The implemented prototype simultaneously identifies and corrects grammatical disfluencies using finite state transduction. The training phase consists of two stages. The first stage is language model acquisition, where grammatical text from an in-domain source, specifically the Gutenberg Project, are ingested, pre-processed, and broken down into n-gram likelihood probabilities, subsequently transitioned into an equivalent FST containing backoff probabilities. A noise model is constructed and hypothesized alternatives are generated based on various heuristics that differ based on grammatical category, then initialized with a uniform probability. The weights are then learned based on instances from an ungrammatical text source, the Narrative Theatre. The training is unsupervised and merely requires the presence of text that contributes to hypothesized error counts. Based on composition calculus available in the OpenFST library, the language and noise model are cascaded and the result is a language-noise generative model. Test instances are converted into linear FST representations and composed against the language-noise model to present alternative outcomes of the observed text. Using Bayesian inference, the optimal hypothesized sentence is calculated by determining the maximum likelihood estimate, or which product of posteriori

(language-noise-sentence) and apriori (language-sentence) yields the maximum probability. The proposed approach is the first attempt at addressing grammatical disfluencies geared toward middle school students' narrative writing.

7.3 Limitations

This dissertation research has proposed a novel approach to automated grammar correction, which examines sentences and hypothesized corrections from a whole sentence level. There are several limitations of the research.

First, the narrative text has a very complex rhetorical structure. Since writers introduce the concept of characters in the story, the dynamic of these characters interacting and/or speaking to themselves or to others presents challenges as the narrative voice often shifts repeatedly during the story. In many cases, stories do not follow a linear path from beginning to end, but can introduce novel variations of timeline shifting such as flashbacks, foreshadowing and zigzags (W. Mann 1987; W. C. Mann and Thompson 1988). The challenges relating to rhetorical structure are further compounded by syntactic and semantic challenges, especially in the context of novice writers (Hayward and Schneider 2000). When writing narrative centered text, novice writers often omit quotation delimiters identifying a change in narrative voice. These characteristics are amplified in the context of middle school student' writing. The highly diverse and imaginative stories offered by these students are dense with not only significant and highly varied shifts in plot progression, but also more nuanced elements of writing style that may include idiomatic expressions, slang that may or may not be intentionally misspelled, among others.

Second, the wFST noise model used in this implementation, while an effective approach for modeling grammatical errors, is uniform for all students. It is hypothesized that middle school students encounter different types of language obstacles when constructing passages. Some schools in the United States, and especially the school used to obtain the NARRATIVE THEATRE corpus, often have very diverse student populations. Such subpopulations include: ESL students, below grade level native learners, average performers, and gifted students, all whom differ greatly in the dynamics of their writing, and consequently the types of grammatical errors they exhibit. For instance, below grade native learners may tend to write passages with more phonetic related spelling errors. ESL students may tend to use incorrect prepositions or omit or write unnecessary articles relating to a particular noun. Gifted students may produce copious typographical errors as a consequence to a very fluid writing pace.

Second, the existing n-gram model used in this wFST implementation was limited to trigram probabilities. This length of context was chosen as a compromise among various factors such as resource limitations due to sparsity and size of resulting FST, vocabulary size/diversity, and computational performance at runtime. While an improvement to the bigram model used by Park & Levy (2011), the highly localized context still limits the scope of errors that can be accounted for. Incorporating more globalized context into the implementation, either directly through the wFST framework, or by a hybrid approach using global discourse cues, might improve the breadth of grammatical error types that are supported.

Next, punctuation management such as sentence segmentation and missing and spurious comma detection was not implemented in this work. Sentence segmentation does play an important role in some problem domains such as ASR, where the post-ASR output is considered one continuous sentence and in some cases must be segmented into sentences to be managed downstream. Unfortunately, these prosodic features are absent in written text. The noise model in this system could be made to incorporate this functionality with significant changes.

Some grammatical error types are highly dependent on obtaining more globalized discourse features. Examples of this include pronominalization, where an incorrectly used pronoun, such as *She said that he was tired*. While this sentence is certainly syntactically correct, the semantic accuracy is open to discussion. The occurrence of the word *he* could refer to a second person referred by the speaker. Alternatively, it could be an unintentional misspelling wherein no second person exists in the context.

7.4 Future Work

There are several promising lines of future work. First, we plan to use a more extensive language model to incorporate additional discourse information, migrating from a trigram model to a 4 or 5-gram model. This would have the effect of increasing scope for localized discourse. We also propose future work to train several noise models based on user information obtained from our pre-study survey, namely those questions referring to demographics, primary language spoken in household, and writing and spelling self-efficacy. Standardized reading/writing scores and teacher assessment of student performance were also

recorded and could serve as determiners for alternative noise models. The motivation behind this is to observe if personalizing grammar correction systems based on this background information may yield improved results. We would also like to expand our noise model to incorporate other grammatical disfluency categories, such as near-homonym misspellings introduced into the passages (e.g., grand/grant). This could be done with an adjacency matrix mapping phoneme distances based on how similar these phonemes sound. Finally, expanding the size of the corpus for the language model by including additional childrens' works may result in an improvement in the accuracy of an LM-noise composition model.

7.5 Concluding Remarks

Over the past decade, there has been significant progress in many areas of NLP, due in large part to data-driven approaches, such as statistical syntactic parsing and semantic role labeling. However, these models are only as effective as the text that is presented to it. Due to the onerous cost of manually annotating parse tree banks for training statistical syntactic parsers, an alternative approach and one presented in this dissertation is to modify the surface level representation of the input text to better match trained syntactic and semantic analysis frameworks while maintaining the original intended meaning. Based on a statistical noisy channel model technique, the research described in this dissertation has explored the utility of sentence-level grammar correction using weighted Finite State Transduction models. The research described in this dissertation represents a first step toward further exploration in grammar correction for highly noisy text, such as that composed by elementary and middle school writers. Additionally, this work is intended to motivate discussion in the design of

evaluation metrics for tasks relating to post-grammar correction in the context of NLP by presenting a heuristic that compares pre- and post-corrected text against a state-of-the-art semantic role labeler.

References

- Abkarian, G.G., Jones, A., and West, G. (1990). Enhancing Children's Communication Skills: Idioms 'Fill the Bill'. *Child Language Teaching and Therapy*, 6(3), 246–254.
- Aitchison, J. (2001). *Language Change: Progress or Decay?* Cambridge University Press, 2001.
- Allauzen, C., Riley, M., Schalkwyk, J., Skut, W., and Mohri, M. (2007). OpenFst: A General and Efficient Weighted Finite-state Transducer Library. In *Proceedings of the Ninth International Conference on Implementation and Application of Automata*. Springer Ontario, Canada, 11–23.
- Alpaydin, E. (2004). *Introduction to machine learning*. MIT press, 2004.
- Amaral, L. and Meurers, D. (2006). *Using Foreign Language Tutoring Systems for Grammatical Feedback*. 2006.
- Amaral, L. and Meurers, D. (2007). Conceptualizing student models for ICALL. *User Modeling 2007*, 4511, 340–344.
- Baaijen, V., Galbraith, D., Smith-Spark, J., and Torrance, M. (2008). The Effects of Dyslexia on the Writing Processes of Students in Higher Education. In *11th EARLI SIG Writing Conference*. Lund, Sweden.
- Banerjee, S. and Lavie, A. (2005). METEOR: An Automatic Metric for MT Evaluation with Improved Correlation with Human Judgments. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, 65–72.
- Barbosa, L. and Feng, J. (2010). Robust Sentiment Detection on Twitter from Biased and Noisy Data. In *Proceedings of the 23rd International Conference on Computational Linguistics: Posters*, 36–44.
- Belz, A. (2008). Automatic generation of weather forecast texts using comprehensive probabilistic generation-space models. *Natural Language Engineering*, 14(4), 431.
- Bertoldi, N., Cettolo, M., and Federico, M. (2010). Statistical Machine Translation of Texts with Misspelled Words. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*. Association for Computational Linguistics Los Angeles, CA, 412–419.

Bourdin, B., Fayol, M., and Darciaux, S. (1996). The Comparison of Oral and Written Modes on Adults and Children's Narrative Recall. In *Theories, models and methodology in writing research*. Amsterdam University Press, Amsterdam, 1996, 159–169.

Bourdin, B. and Fayol, M. (1994). Is Written Language Production More Difficult than Oral Language Production? A Working Memory Approach. *International Journal of Psychology*, 29(5), 591–620.

Bourdin, B. and Fayol, M. (2002). Even in Adults, Written Production is Still More Costly than Oral Production. *International Journal of Psychology*, 37(4), 219–227.

Brill, E. and Moore, R. (2000). An Improved Error Model for Noisy Channel Spelling Correction. In *Proceedings of the 38th Annual Meeting on Association for Computational Linguistics*. Association for Computational Linguistics Hong Kong, 286–293.

Brockett, C., Dolan, W.B., and Gamon, M. (2006). Correcting ESL Errors using Phrasal SMT Techniques. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, 249–256.

Buhrmester, M., Kwang, T., and Gosling, S.D. (2011). Amazon's Mechanical Turk: a New Source of Inexpensive, Yet High-Quality, Data? *Perspectives on Psychological Science*, 6(1), 3–5.

Carreras, X. and Márquez, L. (2005). Introduction to the CoNLL-2005 Shared Task: Semantic Role Labeling. In *Proceedings of the Ninth Conference on Computational Natural Language Learning*, 152–164.

Chelba, C. and Jelinek, F. (2000). Structured language modeling. *Computer Speech & Language*, 14(4), 283–332.

Chen, S.F. and Goodman, J. (1996). An empirical study of smoothing techniques for language modeling. In *Proceedings of the 34th annual meeting on Association for Computational Linguistics*, 310–318.

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

Coe, J.E. and Oakhill, J.V. (2011). 'txtN is ez fu no h2 rd': the Relation between Reading Ability and Text-messaging Behaviour. *Journal of Computer Assisted Learning*, 27(1), 4–17.

- Cohen, J. (1968). Weighted kappa: Nominal scale agreement provision for scaled disagreement or partial credit. *Psychological bulletin*, 70(4), 213.
- Collins, M. (1999). Head-driven statistical models for natural language parsing. 1999.
- Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., and Kuksa, P. (2011). Natural Language Processing (Almost) from Scratch. *Journal of Machine Learning*, 12, 2493–2537.
- Cucerzan, S. and Brill, E. (2004). Spelling correction as an iterative process that exploits the collective knowledge of web users. In *Proceedings of EMNLP*. Barcelona, Spain, 293–300.
- Dempster, A.P., Laird, N.M., and Rubin, D.B. (1977). Maximum Likelihood from Incomplete Data via the EM Algorithm. *Journal of the Royal Statistical Society*, 39(1), 1–38.
- Deorowicz, S. and Ciura, M.G. (2005). Correcting Spelling Errors by Modelling their Causes. *International journal of applied mathematics and computer science*, 15(2), 275.
- Dickinson, M. and Herring, J. (2008). Developing online ICALL exercises for Russian. In *Proceedings of the Third Workshop on Innovative Use of NLP for Building Educational Applications*, 1–9.
- Doddington, G. (2002). Automatic evaluation of machine translation quality using n-gram co-occurrence statistics. In *Proceedings of the second international conference on Human Language Technology Research*, 138–145.
- Dreyer, M., Smith, J., and Eisner, J. (2008). Latent-Variable Modeling of String Transductions with Finite-State Methods. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics Honolulu, Hawaii, 1080–1089.
- Eastman, C.M. and McLean, D.S. (1981). On the Need for Parsing Ill-Formed Input. *Computational Linguistics*, 7(4), 257.
- Eisner, J. (2001). Expectation Semirings: Flexible EM for Finite-State Transducers. In *Proceedings of the ESSLLI Workshop on Finite-State Methods in Natural Language Processing (FSMNL)*.
- Eisner, J. (2002). Parameter Estimation for Probabilistic Finite-State Transducers. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL)*. Philadelphia, 1–8.

- De Felice, R. and Pulman, S.G. (2008). A classifier-Based Approach to Preposition and Determiner Error Correction in L2 English. In *Proceedings of the 22nd International Conference on Computational Linguistics-Volume 1*, 169–176.
- Fitzgerald, E. and Jelinek, F. (2008). Linguistic resources for reconstructing spontaneous speech text. In *Proceedings of the Language Resources and Evaluation Conference*.
- Forney, G.D. (1973). The viterbi algorithm. *Proceedings of the IEEE*, 61(3), 268–278.
- Fossati, D. and Di Eugenio, B. (2008). I saw TREE Trees in the Park: How to Correct Real-word Spelling Mistakes. In *Proceedings of the 6th International Conference on Language Resources and Evaluation*. Marrakech, Morocco, 896–901.
- Gadde, P., Subramaniam, L.V., and Faruque, T.A. (2011). Adapting a WSJ Trained Part-of-Speech Tagger to Noisy Text: Preliminary Results. In *Proceedings of the 2011 Joint Workshop on Multilingual OCR and Analytics for Noisy Unstructured Text Data*, 5.
- Gildea, D. and Jurafsky, D. (2002). Automatic Labeling of Semantic Roles. *Computational Linguistics*, 28(3), 245–288.
- Golan, J.S. (1999). *Semirings and their Applications*. Springer, 1999.
- Golding, A.R. and Roth, D. (1999). A Winnow-based Approach to Context-sensitive Spelling Correction. *Machine learning*, 34(1), 107–130.
- 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 on Association for Computational Linguistics*. Association for Computational Linguistics, 71–78.
- Golding, A.R. (1995). A Bayesian Hybrid Method for Context-sensitive Spelling Correction. In *Proceedings of the Third Workshop on Very Large Corpora*, 39–53.
- Goth, J., Baikadi, A., Ha, E., Rowe, J., Mott, B., and Lester, J. (2010). Exploring Individual Differences in Student Writing with a Narrative Composition Support Environment. In *Proceedings of the NAACL HLT 2010 Workshop on Computational Linguistics and Writing: Writing Processes and Authoring Aids*. Association for Computational Linguistics Los Angeles, CA, 56–64.
- Graham, S. (1990). The Role of Production Factors in Learning Disabled Students' Compositions. *Journal of Educational Psychology*, 82(4), 781–791.

- Graham, S. (2002). Contribution of Spelling Instruction to the Spelling, Writing, and Reading of Poor Spellers. *Journal of Educational Psychology*, 94(4), 669–686.
- Han, B. and Baldwin, T. (2011). Lexical normalisation of short text messages: Mkn sens a# twitter. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, 368–378.
- Hart, M. (2000). Project Gutenberg. 2000. <http://www.gutenberg.org>.
- Hayes, J. and Flower, L. (1980). Identifying the Organization of Writing Processes. *Cognitive processes in writing*, .
- Hayes, J. (1996). A Framework for Understanding Cognition and Affect in Writing. In C. Levy and S. Ransdell, eds., *The Science of Writing*. Lawrence Erlbaum Associates, Mahwah, NJ, 1996, 1–28.
- Hayward, D. and Schneider, P. (2000). Effectiveness of teaching story grammar knowledge to pre-school children with language impairment. An exploratory study. *Child Language Teaching and Therapy*, 16(3), 255–284.
- Heilman, M. and Smith, N.A. (2009). *Question generation via overgenerating transformations and ranking*. 2009.
- Hirst, G. and Budanitsky, A. (2005). Correcting Real-Word Spelling Errors by Restoring Lexical Cohesion. *Natural Language Engineering*, 11(01), 87–111.
- Hirst, G. (2009). An evaluation of the contextual spelling checker of Microsoft Office Word 2007. In *Recent Advances in Natural Language Processing (RANLP) 2009*. Borovets, Bulgaria.
- Huffaker, D., Wang, J., Treem, J., Ahmad, M.A., Fullerton, L., Williams, D., Poole, M.S., and Contractor, N. (2009). The Social Behaviors of Experts in Massive Multiplayer Online Role-playing Games. In *Computational Science and Engineering, 2009. CSE'09. International Conference on*, 326–331.
- Islam, A. and Inkpen, D. (2009). Real-Word Spelling Correction using Google Web IT 3-grams. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 3-Volume 3*. Edinburgh, United Kingdom, 1241–1249.
- Japkowicz, N. and Shah, M. (2011). *Evaluating Learning Algorithms: A Classification Perspective*. Cambridge University Press, 2011.

Johnson, W.L. and Beal, C.R. (2005). Iterative Evaluation of a Large-Scale, Intelligent Game for Language Learning. In *Proceedings of the Twelfth International Conference on Artificial Intelligence in Education*, 290–297.

Jurafsky, D. and Martin, J.H. (2008). *Speech and language processing*. 2008.

Kellogg, R.T. (1988). Attentional Overload and Writing Performance: Effects of Rough Draft and Outline Strategies. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 14(2), 355.

Kellogg, R.T. (1990). Effectiveness of prewriting strategies as a function of task demands. *The American Journal of Psychology*, 103(3), 327–342.

Kittur, A., Chi, E.H., and Suh, B. (2008). Crowdsourcing User Studies with Mechanical Turk. In *Proceedings of the twenty-sixth annual SIGCHI conference on Human factors in computing systems*, 453–456.

Klein, D. and Manning, C.D. (2002). Fast exact inference with a factored model for natural language parsing. *Advances in neural information processing systems*, 15(2002), 3–10.

Knight, K. and Chander, I. (1994). Automated postediting of documents. In *Proceedings of the National Conference on Artificial Intelligence*, 779.

Koehn, P. (2004). Statistical significance tests for machine translation evaluation. In *Proceedings of EMNLP*. Hong Kong, 388–395.

Kwasny, S.C. and Sondheimer, N.K. (1981). Relaxation Techniques for Parsing Grammatically Ill-formed Input in Natural Language Understanding Systems. *Comput. Linguist.*, 7(2), 99–108.

Langkilde-Geary, I. (2002). An empirical verification of coverage and correctness for a general-purpose sentence generator. In *Proceedings of the 12th International Natural Language Generation Workshop*, 17–24.

Leacock, C., Chodorow, M., Gamon, M., and Tetreault, J. (2010). Automated Grammatical Error Detection for Language Learners. *Synthesis lectures on human language technologies*, 3(1), 1–134.

Lee, J. and Seneff, S. (2006). Automatic Grammar Correction for Second-Language Learners. In *Proceedings of Interspeech*. Citeseer, 1978–1981.

Lee, J. and Seneff, S. (2008). Correcting Misuse of Verb Forms. *Proceedings of ACL-08: HLT*, 174–182.

- Llobet, R., Navarro-Cerdan, J.R., Perez-Cortes, J.C., and Arlandis, J. (2010). OCR Post-Processing using Weighted Finite-State Transducers. *a: a*, 6, 0–5.
- Low, P.B. and Siegel, L.S. (2005). A comparison of the cognitive processes underlying reading comprehension in native English and ESL speakers. *Written Language & Literacy*, 8(2), 131–155.
- Mann, W. (1987). *Rhetorical structure theory: A framework for the analysis of texts*. University of Southern California, Information Sciences Institute, 1987.
- Mann, W.C. and Thompson, S.A. (1988). Rhetorical structure theory: Toward a functional theory of text organization. *Text-Interdisciplinary Journal for the Study of Discourse*, 8(3), 243–281.
- Manning, C.D. and Schütze, H. (1999). *Foundations of statistical natural language processing*. MIT press, 1999.
- Matusov, E., Mauser, A., and Ney, H. (2006). Automatic sentence segmentation and punctuation prediction for spoken language translation. In *Proceedings of the International Workshop on Spoken Language Translation (IWSLT)*, 158–165.
- Mays, E., Damerau, F.J., and Mercer, R.L. (1991). Context Based Spelling Correction. *Information processing & management*, 27(5), 517–522.
- Miller, G.A. (1995). WordNet: a lexical database for English. *Communications of the ACM*, 38(11), 39–41.
- Mitton, R. (1996). *English Spelling and the Computer*. Longman Group, 1996.
- Mohri, M. (2002). Weighted Finite-state Transducers in Speech Recognition. *Computer Speech & Language*, 16(1), 69–88.
- Moore, D., Dines, J., Doss, M., Vepa, J., Cheng, O., and Hain, T. (2006). Juicer: A Weighted Finite-state Transducer Speech Decoder. *Machine Learning for Multimodal Interaction*, 4299(2006), 285–296.
- Mrozinski, J., Whittaker, E.W.D., Chatain, P., and Furui, S. (2006). Automatic Sentence Segmentation of Speech for Automatic Summarization. In *Acoustics, Speech and Signal Processing, 2006. ICASSP 2006 Proceedings. 2006 IEEE International Conference on*. Toulouse, France.

- Mudge, R. (2010). The Design of a Proofreading Software Service. In *Proceedings of the NAACL HLT 2010 Workshop on Computational Linguistics and Writing: Writing Processes and Authoring Aids*. Association for Computational Linguistics, 24–32.
- Nagata, N. (2003). Robo-Sensei: Personal Japanese Tutor. *Cheng & Tsui*, 2003.
- Nagata, N. (2009). Robo-Sensei's NLP-based error detection and feedback generation. *Calico Journal*, 26(3), 562–579.
- Ott, N. and Ziai, R. (2010). Evaluating dependency parsing performance on German learner language. In *Proceedings of the Ninth International Workshop on Treebanks and Linguistic Theories*, 175–186.
- Pande, B.P. (2011). Application of Natural Language Processing Tools in Stemming. *International Journal of Computer Applications*, 27(6), 14–19.
- Papineni, K., Roukos, S., Ward, T., and Zhu, W.J. (2002). BLEU: a Method for Automatic Evaluation of Machine Translation. In *Proceedings of the 40th annual meeting on association for computational linguistics*. Association for Computational Linguistics, 311–318.
- Park, Y.A. and Levy, R. (2011). Automated Whole Sentence Grammar Correction using a Noisy Channel Model. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics*. Portland, OR.
- Pedler, J. (2001). The Detection and Correction of Real-word Spelling Errors in Dyslexic Text. in *Proceedings of the 4th Annual CLUK Colloquium*, .
- Pedler, J. (2007). Computer Correction of Real-Word Spelling Errors in Dyslexic Text. *ReCALL*, 2007.
- Philips, L. (1990). Hanging on the Metaphone. *Computer Language*, 7(12 (December)).
- Philips, L. (2000). The Double Metaphone Search Algorithm. *C/C++ Users Journal*, 18(6), 38–43.
- Pirinen, T.A. and Lindén, K. (2010). Finite-State Spell-Checking with Weighted Language and Error Models. In *Proceedings of the Seventh SaLTMiL workshop on creation and use of basic lexical resources for less-resourced languages*. Valletta, Malta, 13–18.
- Pirinen, T.A. and Silfverberg, M. (2012). Improving Finite-State Spell-Checker Suggestions with Part-of-Speech N-Grams. In *International Conference on Intelligent Text Processing and Computational Linguistics CICLING 2012*.

- Pradhan, S., Ward, W., Hacioglu, K., Martin, J., and Jurafsky, D. (2004). Shallow semantic parsing using support vector machines. In *Proceedings of HLT/NAACL*, 233.
- Punyakanok, V., Roth, D., and Yih, W. (2005). The necessity of syntactic parsing for semantic role labeling. In *International Joint Conference on Artificial Intelligence*, 1117.
- Pusack, J.P. (1983). Answer-processing and error correction in foreign language CAI. *System*, 11(1), 53–64.
- Ryker, R.E., Viosca, C., Lawrence, S., and Kleen, B. (2011). Texting and the Efficacy of Mnemonics: Is Too Much Texting Detrimental? *Information Systems Education Journal*, 9(2), 27.
- Scardamalia, M. and Bereiter, C. (1987). *The Psychology of Written Composition*. Erlbaum Hillsdale, 1987.
- Shannon, C.E., Weaver, W., Blahut, R.E., and Hajek, B. (1949). *The mathematical theory of communication*. University of Illinois press Urbana, 1949.
- Spiro, R.J. and Taylor, B.M. (1980). *On Investigating Children's Transition from Narrative to Expository Discourse: The Multidimensional Nature of Psychological Text Classification. Technical Report No. 195*. ERIC, 1980.
- Stolcke, A., Zheng, J., Wang, W., and Abrash, V. (2011). SRILM at Sixteen: Update and Outlook. In *Proc. IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*.
- Stolcke, A. (2002). SRILM-an extensible language modeling toolkit. In *Seventh International Conference on Spoken Language Processing*.
- Subramaniam, L.V., Roy, S., Faruque, T.A., and Negi, S. (2009). A survey of types of text noise and techniques to handle noisy text. In *Proceedings of The Third Workshop on Analytics for Noisy Unstructured Text Data*, 115–122.
- Tajiri, T., Komachi, M., and Matsumoto, Y. (2012). Tense and Aspect Error Correction for ESL Learners Using Global Context. In *Proceedings of ACL*, 198–202.
- Truscott, J. (2006). The case against grammar correction in L2 writing classes. *Language learning*, 46(2), 327–369.
- Varges, S. (2006). Overgeneration and ranking for spoken dialogue systems. In *Proceedings of the Fourth International Natural Language Generation Conference*, 20–22.

Vitevitch, M.S. (1997). The neighborhood characteristics of malapropisms. *Language and Speech*, 40(3), 211–228.

Voorhees, E., Harman, D.K., and others. (2005). *TREC: Experiment and evaluation in information retrieval*. MIT press Cambridge^ eMA MA, 2005.

Wade-Woolley, L. and Siegel, L.S. (1997). The spelling performance of ESL and native speakers of English as a function of reading skill. *Reading and Writing*, 9(5), 387–406.

Weber, R.M. (1970). A Linguistic Analysis of First-Grade Reading Errors. *Reading Research Quarterly*, , 427–451.

Wilcox-O’Hearn, A., Hirst, G., and Budanitsky, A. (2008). Real-Word Spelling Correction with Trigrams: A Reconsideration of the Mays, Damerau, and Mercer Model. *Computational Linguistics and Intelligent Text Processing*, 4919(2008), 605–616.

Wu, C.F. (1983). On the Convergence Properties of the EM Algorithm. *The Annals of Statistics*, 11(1), 95–103.

Xue, N. and Palmer, M. (2004). Calibrating Features for Semantic Role Labeling. In *Proceedings of EMNLP*.

Zhang, Y., Vogel, S., and Waibel, A. (2004). Interpreting BLEU/NIST scores: How much improvement do we need to have a better system. In *Proceedings of LREC*. Lisbon, Portugal, 2051–2054.

Appendices

Appendix A: FST Semirings

Name	Set	\oplus (Plus)	\otimes (Times)	$\bar{0}$ (Zero)	$\bar{1}$ (One)
Boolean	$\{0, 1\}$	\vee	\wedge	0	1
Real	$[0, \infty]$	+	*	0	1
Log	$[-\infty, \infty]$	$-\log(e^{-x} + e^{-y})$	+	∞	0
Tropical	$[-\infty, \infty]$	min	+	∞	0