MONOLINGUAL SENTENCE REWRITING AS MACHINE TRANSLATION: GENERATION AND EVALUATION

by

Courtney Napoles

A dissertation submitted to The Johns Hopkins University in conformity with the requirements for the degree of Doctor of Philosophy.

Baltimore, Maryland

June, 2018

Abstract

In this thesis, we investigate approaches to paraphrasing entire sentences within the constraints of a given task, which we call *monolingual sentence rewriting*. We introduce a unified framework for monolingual sentence rewriting, and apply it to three representative tasks: sentence compression, text simplification, and grammatical error correction. We also perform a detailed analysis of the evaluation methodologies for each task, identify bias in common evaluation techniques, and propose more reliable practices.

Monolingual rewriting can be thought of as translating between two types of English (such as from *complex* to *simple*), and therefore our approach is inspired by statistical machine translation. In machine translation, a large quantity of parallel data is necessary to model the transformations from input to output text. Parallel bilingual data naturally occurs between common language pairs (such as English and French), but for monolingual sentence rewriting, there is little existing parallel data and annotation is costly. We modify the statistical machine translation pipeline to harness monolingual resources and insights into task constraints in order to drastically diminish the amount of annotated data necessary to train a robust system. Our method generates more meaning-preserving and grammatical sentences than earlier approaches and requires less task-specific data.

Once candidate sentences are generated, it is crucial to have reliable evaluation methods. Sentential paraphrases must fulfill a variety of requirements: preserve the meaning of the original sentence, be grammatical, and meet any stylistic or task-specific constraints. We analyze common evaluation practices and propose better methods that more accurately measure the quality of output. Often overlooked, robust automatic evaluation methodology is necessary for improving systems, and this work presents new metrics and outlines important considerations for reliably measuring the quality of the generated text.

Committee:

Chris Callison-Burch
Associate Professor
Department of Computer and Information Science
University of Pennsylvania

Philipp Koehn
Professor
Department of Computer Science
Johns Hopkins University

Benjamin Van Durme
Assistant Professor
Department of Computer Science
Johns Hopkins University

Acknowledgments

It takes a village to raise a child, and the same can be said of a thesis. I owe my success in graduate school to the numerous people who have supported and guided me through this process. I am first grateful to my three primary mentors throughout the course of my PhD, Chris Callison-Burch, Ben Van Durme, and Joel Tetreault. They pushed me beyond my limits, helped me grow as a researcher, and equipped me with the foundation to continue improving in the years to come. My advisors, Chris and Ben, provided mentorship, advocacy, support, and patience. I am thankful to Chris for recognizing my potential and taking a chance on me, and to Ben for his constant directness and insight. Joel's advice imparted me with clarity and direction, and his continuous mentorship, guidance, and friendship have been invaluable.

I thank Philipp Koehn for serving on my committee and providing insightful feedback and suggestions. I am fortunate to have worked with numerous other mentors over the years in the classroom and the lab, including Mark Dredze, Jason Eisner, David Yarowsky, Nitin Madnani, Aoife Cahill, Martin Chodorow, and Aasish Pappu. I am grateful to the other excellent researchers I have had the fortune to collaborate with, particularly Keisuke

Sakaguchi, Matt Post, and Wei Xu.

I owe much to being a member of the Johns Hopkins community, which cultivated a supportive, inquisitive environment. I am grateful for the camaraderie, support, and challenge from my lab mates and colleagues at CLSP, in particular from the fellow members of TeCHo and BLAB. I extend a special thanks to Ruth Scally, Desirée Cleeves, Cathy Thorton, and Debbie Reynolds for their support and warmth throughout the years. One of the greatest gifts has been the friends I have found in the NLP community, especially the lifelong friends made through Hopkins: Anni, Juri, Svitlana, Ann, and Katie.

I also wish to acknowledge those who helped set me down this path and without whom I would not be where I am today. Christiane Fellbaum and Phil Johnson-Laird inspired me with a passion for research and introduced me to the intersection of computation and language. Matthew Lore never ceased to challenge me and push me beyond my limits and comfort zone: to him in particular I am thankful because I owe so much of my success to him. He instilled in me a deep respect for the work of editors and writers, whose work can never be made obsolete by technology. I was fortunate to meet several people at Columbia who advised and guided at the beginning of this journey: David Elson, Kathy McKeown, and Drago Radev.

I am thankful to Joanna, who has been a constant support by my side through the good and the bad. To my Baltimore book club—Becky, Christina, Emily, Jason, Nina, and Tracy—thank you for your friendship and conversation over the years. I couldn't have done it without your help building my confidence and providing much needed balance.

Finally, I am grateful to my family. To the Cohens, whose love and support gave me strength over the years. To Dan, who encouraged and believed in me when I had the crazy idea to change careers and move to Baltimore, and to Shelly and Ira for offering invaluable advice and grounding me. To Charlie for blessing me as a mother, providing immeasurable perspective and inspiration, and reminding me of the wonder of the world around us—you give meaning to everything I do. To Ashley, for being there and getting me (even when you thought I built robots)—I love you and value you beyond your imagination. Most importantly, I am thankful to my parents for always supporting me and believing that I can do more than I think I am capable of.

Dedicated to Charlie, who inspires me in everything that I do. Nothing is more rewarding or makes me prouder than being your mother.

In memory of my grandfathers, who nurtured my curiosity and whom I miss everyday. I would not be who I am without them.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Paraphrasing—communicating the same meaning with different surface forms—is one of the core characteristics of natural language and represents one of the greatest challenges faced by automatic language processing techniques. This work examines a collection of tasks in natural language processing (NLP) that fall under the umbrella of sentential paraphrasing or *text-to-text generation* (which we will refer to as T2T in this thesis). To illustrate what it means to paraphrase a sentence, consider the following sentence from Wikipedia (2010b):

> *An umbrella term is a word that provides a superset or grouping of related concepts, also called a hypernym.*

This sentence can be rewritten in several ways while preserving the meaning. It can be shortened:

> *An umbrella term, or hypernym, is a word that provides a superset of related concepts.*

or it can be lengthened:

> *An umbrella term, which is also called a hypernym, is a word with a broad meaning denoting a superset of related, more specific concepts.*

It can also be simplified:

> *An umbrella term is a word that describes a group of related ideas. It is also called a hypernym.*

Notice that there are several ways to rewrite this sentence so that the overall meaning remains the same. These examples include one or more of the following operations: deletion, insertion, and substitution. These operations can apply on the token or phrasal level. A paraphrase is a *phrasal* substitution, even if there may be a combination of all three operations on individual tokens within the paraphrase. Figure 1.1 shows paraphrases of *hit the sack* and how they can be inserted into a sentence as a phrasal substitution or token-level operations. Instead of examining individual, token-level operations, we focus on broader phrase-level rewrites.

In this work, we examine paraphrasing entire sentences given task-specific constraints, such as *length(output)* $<$ *length(input)* or *readability(output)* $>$ *readability(input)*. We demonstrate how to perform whole-sentence transformations with robust systems that require minimal annotated data by leveraging advances in statistical machine translation and general-purpose corpora. We also establish reliable evaluation methodologies that are appropriate for holistic, sentence-level transformations. The goal of this thesis is to examine three primary aspects of T2T, specifically

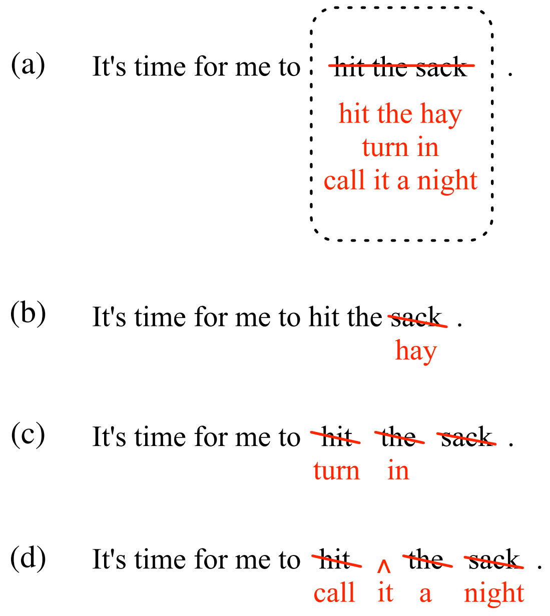

(a) It's time for me to ~~hit the sack~~ .
hit the hay
turn in
call it a night
(b) It's time for me to hit the ~~sack~~ .
hay
(c) It's time for me to ~~hit~~ ~~the~~ ~~sack~~ .
turn in
(d) It's time for me to ~~hit~~ ^ ~~the~~ ~~sack~~ .
call it a night

Figure 1.1: A sentence with paraphrase candidates for the phrase *hit the sack*, represented with a single phrase-level substitution (a) or token-level insertion, deletion, and substitution operations (b–d).

Data. Models for text-to-text generation rely on *parallel corpora*, consisting of an original text aligned to a that has been transformed into a text that meets the task-specific constraints. The original and transformed texts are parallel since the sentences are aligned to each other. Parallel corpora often do not reflect the requirements of the task (e.g., text labeled *simple* is in fact equally or more difficult to read than *complex* text). Additionally, there is frequently only one gold-standard version of each sentence, which is problematic because there is seldom only one correct way to change a sentence, and creating corpora for these tasks is expensive and time consuming.

Evaluation methodology. NLP tasks often rely on *automatic evaluation*, which as-

> signs a score to a model's output using some automatically calculated statistics. We rely on these metrics to assess how good a model is, since evaluating output manually with human raters is time-consuming and expensive. Automatic evaluation is biased in many ways but, across tasks, metrics favor systems that make minimal changes to the input even when the objectives of the task are not met. Because human evaluation is expensive and infrequently performed, unbiased evaluation methodologies are crucial for accurately comparing systems and advancing the field.
>
> **Methods.** We view each of these problems as a semi-supervised *paraphrasing* task and develop a unified framework that removes the dependency on large amounts of parallel task-specific training data and harnesses advances in statistical machine translation (SMT) to outperform earlier models.

A clear analogy to monolingual T2T is machine translation (MT), which is a bilingual task of translating text from a source language into a target language, under the constraints that the generated text must be *fluent* (grammatical and well-formed) and *adequate* (meaning preserving). Our T2T framework is inspired and adapted from MT and therefore use the vocabulary of MT: the input sentence is the *source*; gold-standard, human-rewritten sentences are the *references* or *target*; and system output is called the *candidate*. The unified T2T framework is shown in Figure 1.2, and Figure 1.3 provides a detailed look into the T2T component.

Automatic systems for all of these tasks share the following requirements:

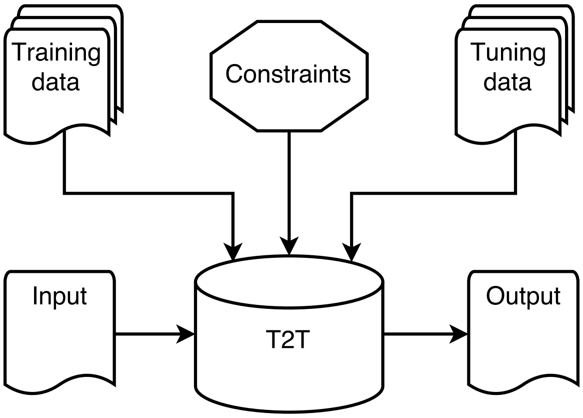


Figure 1.2: A unified framework for T2T.

- **Objectives.** A set of constraints that the output must meet. All tasks share the constraints of *fluency* and *adequacy* and have additional task-specific constraints.

- **Parallel data.** Parallel text where one side reflects standard input text and the other side represents that text transformed in accordance with the objectives of the task.

- **Evaluation technique.** Methodology for evaluating system output using human judges and/or automatic metrics.

This thesis will focus on three T2T tasks of increasing complexity: sentence compression, text simplification, and grammatical error correction. Sentence compression (SC) is the most straight-forward of these tasks because there is a single objective—output text must be shorter than the input—which is simple to quantify by comparing the length of the source and candidate. In its most simple form, SC is a task that only uses a deletion operation. In this thesis, we will discuss how substitution (paraphrasing) can be used to improve the quality of sentence compressions.

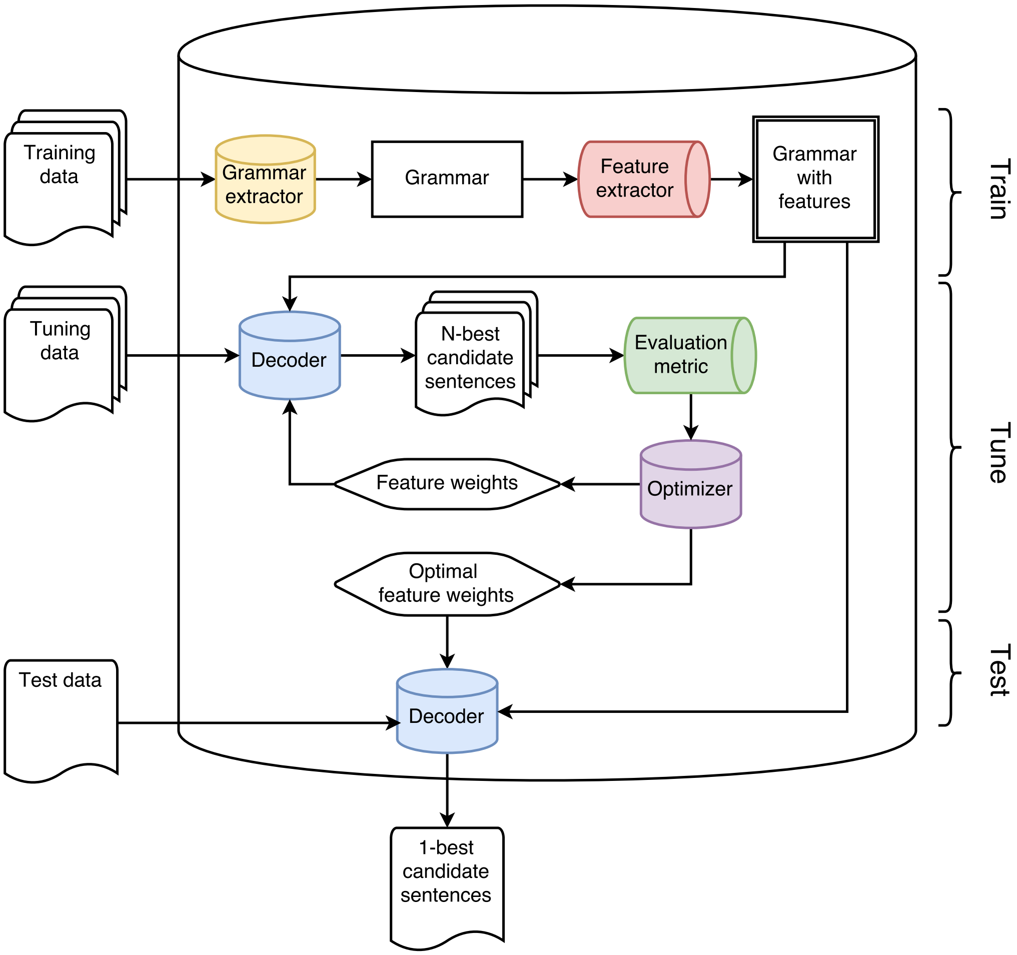


Figure 1.3: Detailed schematic of an SMT-based T2T system. Constraints are encoded in the evaluation metric.

Text simplification (TS) and grammatical error correction (GEC) are significantly more complex, not only because they involve all three operations, but because *readability* and *grammaticality* are underspecified terms that cannot be easily quantified. The objective of TS is to increase the readability of an input text, however it is difficult to measure this change automatically. There are cognitively motivated metrics for measuring the complexity of text based on the average word and sentence lengths, but these metrics do not represent fluency or adequacy and are therefore insufficient for TS evaluation. They also

(a)	Colorless green ideas sleep furiously.
(b)	*Furiously sleep ideas green colorless.

Table 1.1: Two sentences that are nonsensical, but only (b) is ungrammatical (Chomsky, 1957, p. 15).

fail to evaluate syntactic complexity or long-distance dependencies.

GEC is the only of these tasks in which the input is not well-formed, which may obfuscate the intended meaning and therefore require inference to correct.[1] The objective of GEC may appear obvious at first glance, however the definition of *grammaticality* is subject to interpretation, in the following ways. It can mean a sentence that is syntactically correct but not necessarily semantically correct (Figure 1.1); notions of grammaticality can vary based on dialect (Zanuttini and Horn, 2014); and *acceptability* may be sufficient, even while not all acceptable sentences are grammatical (Otero, 1972). In addition to this lack of consensus, grammaticality in any form is difficult to quantify automatically.

We will address these issues in subsequent chapters of this thesis.

1.1 Organization

For each task included in this thesis we explore *evaluation* methodologies, *data* for validating and testing, and *generation* with the unified framework. We will first provide a literature review of the existing work related to the tasks described in this thesis (Chapter 2). The rest of the thesis is structured as follows:

[1] For the work described herein, we assume that the input text for SC and TS is clean and grammatical.

Part I: Sentence Compression

Evaluation: Chapter 3 identifies bias in existing practices in evaluating sentence compressions and makes recommendations for more balanced, informative assessment.

Data: In Section 3.4 we propose a new source of training data for SC and identify that systems are unfairly compared when the degree of compression is not considered, and Section 3.5.1 describes the creation of a new, multiple-reference test set that contains gold-standard compressions of different lengths.

Generation: Chapter 4 demonstrates how paraphrasing is effective for compression. In a constrained task that compresses sentences using paraphrasing and not deletion, we determine that paraphrastic compressions are better than extractive compressions at the same compression rate. We also motivate the use of character-based compression rates (instead of token-based). Section 4.5 describes a method for SC inspired by statistical machine translation (SMT).[2]

Part II: Text Simplification

Data: In Chapter 5, we consider how to use an existing source of data, Wikipedia, for the task of automatic sentence simplification. Section 5.2 examines features of simplifications in a parallel corpus of English Wikipedia and Simple English Wikipedia documents and Section 6.1.4 identifies problems with the using an automatically

[2] From joint work with Juri Ganitkevitch (Ganitkevitch et al., 2011). My contributions to that work are discussed in Section 1.3.

aligned Wikipedia corpus for this task. In this corpus, simplified sentences are not always more simple—and sometimes are more complex—than the original sentence.

Evaluation: Section 6.1.1 investigates potential methods for automatically evaluating simplified text. We examine whether MT metrics can be applied to the task, and finding them lacking, propose a new metric for evaluating TS that incorporates a measure of readability with adequacy and fluency.

Generation: In Chapter 6, we identify how to apply a general-purpose paraphrase database for TS and develop features indicative of lexical and syntactic complexity. We combine these findings with the metric described in Chapter 5 into a model that outperforms earlier approaches.

Part III: Grammatical error correction

Data: Chapter 7 critiques the framing of GEC as a task that targets individual, isolated errors and argues that the goal should be jointly correcting whole sentences and have the goal of correcting for fluency instead of just grammaticality. To support this new definition, we develop a new, multiple-reference test corpus.

Evaluation: Section 7.4 presents the first meta-evaluation of GEC, collecting human judgments of GEC system outputs and calculating how reliable existing metrics are. We propose a new metric that outperforms earlier metrics and strongly correlates with human judgments.

Generation: Chapter 8 presents a method for GEC that represents an adaptation of SMT and uses a small number of artificially generated paraphrases to overcome sparsity in training data.

1.2 Contributions

The primary contributions of this thesis are as follows:

1. Empirical

We describe a unified framework for monolingual T2T tasks inspired by statistical machine translation and demonstrate how it can be applied to the complex tasks of simplification and grammatical error correction. One aspect of this framework is to design targeted feature functions for the paraphrase grammar, which enables the system to automatically choose paraphrases from a large source of general-purpose paraphrases that satisfy the objective of each task. We show that paraphrastic sentence compressions have higher fluency and adequacy than extractive compressions at the same compression rate (Chapter 4). We also demonstrate that better simplifications are generated using a large resource of general paraphrases instead of task-specific paraphrases. Finally, we show that artificially generated data significantly improves the performance of SMT-based GEC.

2. Evaluation

We examine existing methodology for evaluating the output of T2T systems and identify several instances in which the evaluation can be misleading by not reflecting human judgments or making mismatched comparisons.

3. Datasets and Resources

In support of this work, we have developed and released the following resources:

- A set of *language packs* for performing sentence compression and text simplification using the models described in this thesis (Napoles et al., 2016b).[3] A language pack contains pre-trained models and software needed to generate output, which can be treated as a black box or further customized. The language packs are described in Appendix B.
- Two annotated corpora for evaluating GEC: a set of four multiple references for the CoNLL-2014 Shared Task test set (Sakaguchi et al., 2016) and JFLEG (Napoles et al., 2017c), a new tuning and test set containing four multiple references for a subset of the GUG corpus (Heilman et al., 2014).[4] The motivation and process for collecting these corpora are described in Sections 7.1 and 7.3.

[3] https://cwiki.apache.org/confluence/display/JOSHUA

[4] https://github.com/keisks/jfleg

4. Software

We have released several software packages used for this research. Specifically:

- An integer linear programming (ILP) framework for sentence compression: A re-implementation of Clarke and Lapata (2008)'s ILP model for sentence compression along with an extension that allows for phrase substitution (Chapter 4). This uses an industrial ILP solver (IBM CPLEX), allowing large problem sizes to be solved in a relatively short time. `https://github.com/cnap/sentence-compression`
- Java classes for extracting readability-based features and a readability metric for text simplification (Napoles, 2012). `https://github.com/cnap/joshua`
- Three GEC evaluation metrics described in Chapter 7, one of which is a re-implementation of the grammaticality model of Heilman et al. (2014). `https://github.com/cnap/grammaticality-metrics`
- An online platform for scoring GEC output on CodaLab, with a public leaderboard and shared metrics and evaluation data. `https://competitions.codalab.org/competitions/15475`
- A pipeline for MT-based GEC, including feature extraction and creating artificial data, and an accompanying program that provides a detailed lexical and syntactic analysis of changes made in parallel text. `https://github.com/cnap/smt-for-gec`

1.3 Related Publications

The work presented in this document are the author's and have been included in published articles or works under review. I have been fortunate to collaborate with other student and post-doctoral researchers, and have specified below my contributions when a chapter draws from our joint work.

- Chapter 3 extends "Evaluating sentence compression: Pitfalls and suggested remedies" (Napoles, Van Durme, and Callison-Burch, 2011).
- Chapter 4 extends "Paraphrastic Sentence Compression with a Character-based Metric: Tightening without Deletion" (Napoles, Callison-Burch, Ganitkevitch, and Van Durme, 2011), which is an extension of Clarke and Lapata (2006). My contribution in this work is the development and evaluation of the compression approach presented therein (Section 4.1).
- Chapter 4 also describes "Learning Sentential Paraphrases from Bilingual Parallel Corpora for Text-to-Text Generation" (Ganitkevitch, Callison-Burch, Napoles, and Van Durme, 2011), for which my contributions are the use of compression rate in the optimization metric and for evaluation and the evaluation of paraphrastic compressions, and which serves as inspiration for the approaches to text simplification (Chapter 6) and GEC (Chapter 8).
- Chapter 5 contains analysis presented in "Learning Simple Wikipedia: A Cogitation

in Ascertaining Abecedarian Language" (Napoles and Dredze, 2010).

- Chapter 6 and Chapter 5 present research initially described in "Computational Approaches to Shortening and Simplifying Text" (Napoles, 2012), and which was later extended by Wei Xu in "Problems in Current Text Simplification Research: New Data Can Help" (Xu, Callison-Burch, and Napoles, 2015) and "Optimizing statistical machine translation for text simplification" (Xu, Napoles, Pavlick, Chen, and Callison-Burch, 2016), the latter overlapping significantly with Napoles (2012). Unless specified, the research presented in this thesis draws solely from Napoles (2012).

- Chapter 7 reflects research resulting from a collaboration with Keisuke Sakaguchi, Joel Tetreault, and Matt Post. I have outlined my contributions in each of our joint papers below.

  - "Ground Truth for Grammatical Error Correction Metrics" (Napoles, Sakaguchi, Post, and Tetreault, 2015): My primary contribution is development of the GLEU metric that penalizes overlap between system output and unchanged input (Chapter 7).

  - "Reassessing the Goals of Grammatical Error Correction: Fluency Instead of Grammaticality" (Sakaguchi, Napoles, Post, and Tetreault, 2016): For this work, I performed analysis of grammatical edits and automatic metrics (Chapter 7).

  - "There's No Comparison: Reference-less Evaluation Metrics in Grammatical

    Error Correction" (Napoles, Sakaguchi, and Tetreault, 2016): My contribution is development of the grammaticality-based metrics described in Chapter 7 and the CodaLab scoring platform.

  - "JFLEG: A Fluency Corpus and Benchmark for Grammatical Error Correction" (Napoles, Sakaguchi, and Tetreault, 2017): I performed qualitative and quantitative analysis of the annotations we collected, automatic evaluation of system outputs not including the TrueSkill experiments, and qualitative analysis of system output (Chapter 7).

- Chapter 8 describes an MT-based approach to GEC, which is an extension of "Systematically Adapting Machine Translation for Grammatical Error Correction" (Napoles and Callison-Burch, 2017).

1.4 Other Publications

While the entirety of my PhD research has focused on increasing the accessibility of written language by increasing clarity of transforming the text to reach a specific target audience, I have also investigated related problems of evaluating student writing (Napoles and Callison-Burch, 2015), quantifying the effect of ungrammatical text on the performance of models trained on clean text (Napoles et al., 2016c), and automatically identifying constructive conversations in online discourse (Napoles et al., 2017b; Napoles et al., 2017a).

I also compiled and released a large-scale annotated corpus of newswire text, which is available to the community through the Linguistic Data Consortium[5] (Napoles et al., 2012). Annotated Gigaword has been used in support of a variety of related T2T tasks, including abstractive sentence summarization (akin to compression) (Rush et al., 2015) and generating image descriptions (Elliott and Keller, 2013).

Finally, I helped organize the first shared task on the automatic evaluation of scientific writing (AESW16), which had the goal of identifying grammatical and stylistic errors in scientific text (Daudaravicius et al., 2016). Drawing from the research presented in this work, we have also proposed a new shared task for GEC (Sakaguchi et al., 2017a).

[5] `https://www.ldc.upenn.edu`

Chapter 2

Literature Review

This chapter reviews existing work related to this thesis and is organized by task (compression, simplification, and error correction).

2.1 Sentence Compression

One of the earliest T2T tasks is document summarization, which was originally formulated as choosing a set of sentences from a document to create a summary, such that their total length was less than some limit. Knight and Marcu (2000) proposed *sentence compression* (SC), the task of selecting an ordered subset of tokens from a sentence to form a new sentence that is shorter, and thereby more sentences can be included into an *extracted* summary. This is called deletion-based or *extractive compression*, where the only transformation is deletion. In contrast, *abstractive compression* involves reordering and replacements as well as deletions. A good compression is grammatical, retains the most important infor-

Source	Compression
Knight and Marcu (2000)	~~Beyond that basic level,~~ **T**he operations of the three products vary widely.
McDonald (2006)	~~The first new product,~~ ATF Protype~~,~~ is a line of digital postscript typefaces ~~that~~ will be sold in packages of up to six fonts.
Galley and McKeown (2007)	The chemical etching process used for glare protection is effective ~~and will help if your office has the fluorescent-light overkill that's typical in offices~~.
Cohn and Lapata (2008)	*His wife author Margo Kurtz* ~~He~~ died last Thursday ~~at his home~~ from complications ~~following a fall,~~ **after a decline** ~~said his wife author Margo Kurtz~~.
Filippova et al. (2015)	~~State Sen.~~ Stewart Greenleaf discusses his ~~proposed~~ human trafficking bill ~~at Calvery Baptist Church in Willow Grove Thursday night~~.

Table 2.1: Example sentence compressions from other published works. ~~Struck-out~~ text has been deleted, **bolded** text has been inserted, and *italicized* text has been moved.

mation from the original text, and meets some target compression rate. The *compression rate*—which is more accurately described as the compression ratio—describes how much the original text has been shortened. Specifically:

$$\text{Compression rate} = \frac{\text{\# words in compression}}{\text{\# words in original}} \tag{2.1}$$

Most of the previous research on sentence compression focuses on deletion using syntactic information (e.g., Galley and McKeown, 2007; Knight and Marcu, 2002; Nomoto, 2009; Galanis and Androutsopoulos, 2010; Filippova and Strube, 2008; McDonald, 2006;

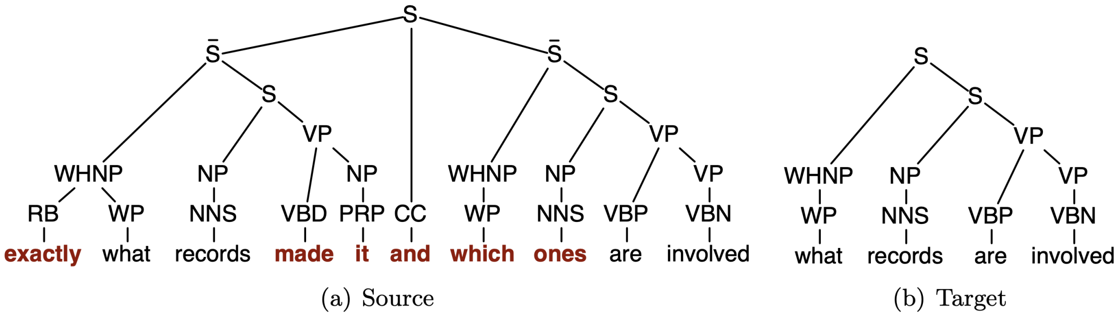


Figure 2.1: Example of a sentence compression generated with tree transduction, from Cohn and Lapata (2009), Figure 1. **Bolded** nodes are deleted.

Yamangil and Shieber, 2010; Turner and Charniak, 2005). These models operated on the sentence level with few exceptions, such as Cohn and Lapata (2007), who used document-level discourse information to inform compression. Closely related tasks to sentence compression are headline and title generation (e.g., Dorr et al., 2003; Vandeghinste and Pan, 2004; Marsi et al., 2009). These have roughly the same goal but the generated sentence should be descriptive of an entire document and does not need to be a complete sentence (Dorr et al., 2003). In these tasks, compression rate is targeted at the character level and not the word level. For instance, Corston-Oliver (2001) deleted characters from words to shorten the character length of sentences and earlier work in subtitling made one-word substitutions to decrease the character length (Glickman et al., 2006).

The leading approaches developed at the time of this work include Clarke and Lapata (2008)'s compression model, which uses a series of constraints in an integer linear programming (ILP) solver, and *Extractive* compression was pursued until Cohn and Lapata (2008), who introduced a tree-transduction model to incorporate paraphrasing into SC (this approach is described in more detail in Cohn and Lapata (2009)). Cohn and Lapata learn a synchronous tree substitution grammar (STSG) from paired monolingual sentences, which they argue is a natural fit for sentence compression, since deletions introduce structural mismatches (Figure 2.1). Little other work has examined abstractive compression, including Zhao et al. (2009), who apply an adaptable paraphrasing pipeline to sentence compression, optimizing for F-measure over a manually annotated set of gold standard paraphrases. Woodsend et al. (2010) incorporate paraphrase rules into a deletion model. In Chapter 4, we demonstrate the viability of paraphrasing for compression, and Section 4.5 applies our unified framework to sentence compression, outperforming other models for abstractive compression.

Following this work, a large-scale abstractive compression corpus has been released (Toutanova et al., 2016) and several deep learning approaches to sentence compression have been reported, both extractive and abstractive, e.g., Filippova et al. (2015) and Rush et al. (2015), respectively.

2.2 Text Simplification

Text simplification is related to sentence compression because deletions are common in simplified text. However, it is a much more nuances and complicated task because lexical and syntactic substitutions are often also necessary. The earliest approaches to automatic text simplification (TS) rely on handwritten rules, e.g., PEST (Carroll et al., 1999), its SYSTAR module (Canning et al., 2000), and the method described by Siddharthan (2006). Later work in sentence simplification favors automatic or semi-supervised methods for acquiring simplifying rules. Woodsend and Lapata (2011) encoded the simplification problem as an integer linear program (ILP), which maximized the likelihood of sentence transformations and rewarded sentences that were shorter than a given target. Their best system, RevILP, used a quasi-synchronous grammar extracted from the revision history of Simple English Wikipedia. More similar to the model we will present in Chapter 6, early more sophisticated approaches were inspired machine translation and learn a phrase table from parallel unsimplified–simplified text (from English Wikipedia and Simple English Wikipedia, respectively). Zhu et al. (2010) used a method inspired by MT to generate simplifications (TSM). Their translation grammar consisted of three operations: (1) splitting syntactic constituents into a new sentence (e.g., SBAR or WHNP), (2) dropping and reordering nodes, and (3) word and phrase substitution. Coster and Kauchak (2011a) applied existing MT methods to extract translation rules, not limiting operations and also introducing phrasal deletions. Wubben et al. (2012) also used phrase-based MT for simplifying text, but further reranks the n-best list to favor output that is less similar to the original text.

Chapter 6 describes how we simplify text in our unified framework, using a combination of paraphrases and deletions from a large database of syntactic and lexical paraphrases. This approach was later extended in Xu et al. (2016). Table 2.2 contains example simplifications. Notice how the output improves from the rule-based, ILP method (Woodsend and Lapata, 2011) to the MT-based approach with rules learned from Wikipedia (Coster and Kauchak, 2011a) to rules from a larger paraphrase corpus (Xu et al., 2016). The para-

Woodsend and Lapata (2011)	Wonder has recorded several critically acclaimed albums and hit singles~~, and~~ **. He** writes**.** ~~and~~ **He** ~~produces~~ **makes** songs for many of his label mates and outside artists as well.
Coster and Kauchak (2011a)	Nicolas Anelka is a French ~~footballer~~ **football player** ~~who currently~~ **. He** plays ~~as a striker~~ for Chelsea ~~in the English premier league~~.
Xu et al. (2016)	Jeddah is the ~~principal~~ **main** gateway to Mecca, Islam's holiest city, which ~~able-bodied~~ **sound** Muslims ~~are required~~ **have** to visit at least once in their ~~lifetime~~ **life**.

Table 2.2: Example sentence simplifications from other published works. ~~Struck-out~~ text has been deleted, **bolded** text has been inserted, and *italicized* text has been moved.

phrases used in this work (Ganitkevitch et al., 2011) were a precursor to the Paraphrase Database (PPDB) (Ganitkevitch and Callison-Burch, 2014). Subsequent to this thesis, the Simple PPDB was released, which contains a subset of the PPDB identified to contain simplifying lexical transformations (Pavlick and Callison-Burch, 2016).

2.3 Grammatical Error Correction

One of the oldest NLP tasks is grammatical error correction (GEC). Beginning in the 1980s, research groups at IBM and Bell Labs focused on spelling and grammar correction (Heidorn et al., 1982; MacDonald et al., 1982; Richardson and Braden-Harder, 1988). Earlier approaches to grammatical error correction developed rule-based systems or classifiers targeting specific error types such as prepositions or determiners (e.g., Chodorow and Leacock, 2000; Eeg-Olofsson and Knutsson, 2003; De Felice and Pulman, 2008; Gamon et al., 2008; Tetreault and Chodorow, 2008; Rozovskaya et al., 2014). In a departure from this limited approach, Park and Levy (2011) proposed a noisy-channel model to perform *whole-sentence* grammatical error correction, but trained models for different error types instead of using a joint model. The 2012 and 2013 shared tasks in GEC both targeted only certain error types (Dale et al., 2012; Ng et al., 2013), to which classification was appropriately suited. However, the 2014 CoNLL Shared Task introduced a new problem of simultaneously correcting 28 types of grammatical errors, encouraging several MT-based approaches to GEC (e.g., Felice et al., 2014; Junczys-Dowmunt and Grundkiewicz, 2014). Two of the best CoNLL 2014 systems used MT, treating it as a black box and reranking output (Felice et al., 2014), and customizing the tuning algorithm and using lexical features (Junczys-Dowmunt and Grundkiewicz, 2014). The other leading system was classification-based and only targeted certain error types (Rozovskaya et al., 2014). Performing less well, Wang et al. (2014) used factored statistical MT, representing words as factored units to more adeptly handle morphological changes. Shortly after the shared task, a system combining classifiers and statistical MT with no further customizations reported better performance than all competing systems (Susanto et al., 2014).

The current leading GEC systems all use MT in some form, including hybrid approaches that use the output of error-type classifiers as MT input (Rozovskaya and Roth, 2016) or include a neural model of learner text as a feature in SMT (Chollampatt et al., 2016); phrase-based MT with sparse features tuned to a GEC metric (Junczys-Dowmunt and Grundkiewicz, 2016); and neural MT (Yuan and Briscoe, 2016).

Corpus	# sents.	Characters/sentence	Sentences changed	LD
AESW	1.2M	133	39%	3
FCE	34k	74	62%	6
Lang-8	1M	56	35%	4
NUCLE	57k	115	38%	6

Table 2.3: Publicly available parallel corpora available for GEC.

Source	Sentence
FCE	Also, in {`SPELLING:` Augost → August} I will be studying a summer {`WORD ORDER:` English course → course in English}.
NUCLE	Dubai will be a good example for this as previously the country got almost no natural water {`MECHANICAL:` ε → .} {`LINK WORD:` and → But} {`PRONOUN:` they → it} {`AGREEMENT` use → uses} irrigation to bring natural water to the country.

Table 2.4: Example sentences from error-coded GEC corpora, with the type of error in `fixed-width font`.

Source	Sentence
AESW	Then each thread **is** compared with the adjacent elements~~,~~ **;** if equal, the corresponding position of **the** bit vector **is** set **to** 0, otherwise **it is** set to 1.
Lang-8	However, ~~only~~ **I have to be careful about** slipping ~~doesn't take care~~.

Table 2.5: Example sentences from parallel GEC corpora. **Bold** text is inserted and ~~struck-out~~ text is deleted.

2.3.1 GEC Corpora

There are two broad categories of parallel data for GEC. The first is error-coded text, in which annotators have coded spans of learner text containing an error, and which includes the NUS Corpus of Learner English (NUCLE; 57k sentence pairs) (Dahlmeier et al., 2013), the Cambridge Learner Corpus (CLC; 1.9M pairs per Yuan and Briscoe (2016)) (Nicholls, 2003), and a subset of the CLC, the First Certificate in English (FCE; 34k pairs) (Yannakoudakis et al., 2011).

The second class of GEC corpora is parallel datasets, which contain the original text and a corrected version of the text, without explicitly coded error corrections. These datasets are cheaper to collect, significantly larger than the error-coded corpora, and may contain more extensive rewrites. The related field of MT uses parallel texts for training and evaluation. Two existing parallel corpora for GEC are the Automatic Evaluation of Scientific Writing (AESW) corpus, with more than 1 million sentences of scientific writing corrected by professional proofreaders (Daudaravicius et al., 2016), and the Lang-8 Corpus of Learner English, which contains 1 million sentence pairs scraped from an online forum for language learners, which were corrected by other members of the `lang-8.com` online community (Mizumoto et al., 2011). Table 2.3 describes the sizes of these corpora and examples from error-coded and parallel corpora are in Table 2.4 and 2.5, respectively. In Chapter 7, we motivate why the GEC community should move away from error-coded corpora and additionally create a new, multiple reference test set without error coding.

2.4 T2T Evaluation

In the three T2T tasks contained in this thesis, little research has focused on evaluation methodologies and meta-evaluation. A common metric used for all three is the F-score, which is a measure of accuracy that combines precision (P) and recall (R). The β value specifies how to weigh precision versus recall, and the default parameter, $\beta = 1$ is the harmonic mean of the precision and recall.

$$F_\beta = (1+\beta^2)\frac{P \cdot R}{\beta^2 \cdot P + R} \tag{2.2}$$

Because sentence compression has primarily been a deletion-based task, evaluation is relatively simple: calculate the accuracy or F-score of words selected for the output compared to the gold standard. (Riezler et al., 2003) proposed evaluating with the F-score over dependency relations of the candidate compression compared to the gold standard. The machine-translation metric, BLEU (Papineni et al., 2002), has also been applied to compression as well as other monolingual generation tasks, including text simplification. BLEU has been shown to correlate with human ratings of machine translation output (Doddington, 2002), however is not universally true for T2T output, as this thesis shows.

For text simplification, F-score over words has also been used as a metric. The Flesch-Kincaid grade level, designed to automatically assess the readability of a text (Kincaid et al., 1975), has also been applied to evaluate the output of TS systems. Wubben et al. (2012) perform a meta-evaluation of BLEU and Flesch-Kincaid compared to human judgments, finding neither captures adequacy. Later, Xu et al. (2016) compares newly proposed TS metrics to human judgments with multiple references for the first time, but also finds BLEU insufficient.

A series of shared tasks starting in 2011 prompted the development and scrutiny of new

Method	**Accuracy**	**Alignment**
`wdiff`	0/1	{Xinhua →} The xinhua portion of the English Gigaword3
M^2	1/2	{ε → The} {Xinhua portion of → xinhua portion of} the English Gigaword3
Reference correction		{ε → The} Xinhua portion of the English Gigaword3

Table 2.6: Example candidate text aligned to the reference with two different methods. Alignment spans are denoted with {}, and *Accuracy* indicates the accuracy of the candidate according to that alignment.

metrics for evaluating GEC systems. The Helping Our Own (HOO) shared tasks evaluated systems using precision, recall, and F-score against annotated gold-standard corrections (Dale and Kilgarriff, 2011; Dale et al., 2012). One issue is the difficulty of aligning candidate text to the error-coded spans in the reference. In HOO, the Unix `wdiff` utility[1] found the alignment, but systems were sometimes penalized because of how the texts were aligned. Dahlmeier and Ng (2012) proposed the MaxMatch (M^2) scorer to address this issue, which calculates the F-score over an edit lattice that captures phrase-level edits, where the lattice finds the optimal alignment between the candidate and reference. Table 2.6 shows the difference between the two alignments.

M^2 was the official metric of the subsequent CoNLL Shared Tasks on GEC (Ng et al., 2013; Ng et al., 2014). Felice and Briscoe (2015) proposed I-measure be an interpretable measure, indicating whether a candidate improved or degraded the grammaticality of the source. I-measure computes the accuracy of a token-level alignment between the original,

[1] `https://www.gnu.org/software/wdiff/`

candidate, and reference sentences. In Section 7.4 we will propose a new metric, Generalized Language Evaluation Understanding (GLEU) that captures both fluency and adequacy with n-gram overlap (Napoles et al., 2015).

Part I

Sentence Compression:

A Sandbox for Exploring T2T

In the next two chapters, we will introduce core concepts that motivate and will be explored in this thesis. To illustrate these concepts, we will apply them to the most simple T2T task, sentence compression—which is simple in terms of it having only one, easily measured objective. This part will establish the pattern that will be replicated for each T2T task examined in this thesis: Chapter 3 examines the data and evaluation measures available for sentence compression and makes proposals enabling more reliable evaluation and *abstractive* sentence compression. Chapter 4 will prove the viability of applying paraphrases for shortening text and introduce a framework for generating paraphrastic compressions, the same framework which will be the bases of our models for generating text simplification and grammatical error correction. The findings and techniques we discuss here will be applied to and expanded in two more sophisticated and complicated tasks, text simplification and grammatical error correction.

Sentence compression is the task of automatically shortening text, while ensuring that the generated text is grammatical and faithful to the meaning of the original sentence. Because the output is shorter than the original, it is natural that some meaning is lost, but the most important meeting of the original sentence should be maintained. Sentence compression is a step of the extractive summarization pipeline, which involves selecting sentences from a document to extract to form a summary. The summary can be further condensed with sentence compression. Table 2.7 shows examples of compressed sentences. We choose sentence compression as the first task explored in this thesis because the objective is easily measured—is the candidate text shorter than the original—and therefore we can focus research on measuring grammaticality and meaning retention of the output while knowing the primary objective of the task is met.

Before discussing methods for applying MT techniques to monolingual translation tasks, we first examine evaluation methodology. In the absence of a shared task, which increases validity by comparing systems under the same conditions, we examine evaluation practices used in sentence compression research with the goal of developing informative and unbiased metrics.[2] The findings of this study are informative for the best practices for evaluating other T2T and NLG tasks more broadly, in particular those visited in Chapters 5 and 7. Specifically, our primary conclusion is that compression rate is a strong predictor of compression quality, and that perceived improvement over another model can be a byproduct of producing longer output. In other words, conservative systems are rewarded

[2] Even shared tasks can lead to bias if systems over-optimize to the data used for that task, as we will discuss in the context of GEC in Chapter 7.

Original	Microsoft alone has lost one-third of its market value.
Extractive	Microsoft lost one-third of its market value.
Abstractive	Microsoft*'s* market value *dropped by 1/3*.
Original	The first new product, ATF ProType, is a line of digital PostScript typefaces that will be sold in packages of up to six fonts.
Extractive	ATF ProType is a line of digital PostScript typefaces that will be sold in packages of up to six fonts.
Abstractive	The *initial* product, ATF ProType, is sold in packages of up to six *PostScript typefaces*.
Original	Actual hardware and maintenance costs have decreased 40 percent over the years, while the number of users supported has increased by over 300 percent since 1983.
Extractive	Actual hardware and maintenance costs have decreased .
Abstractive	*Since 1983,* hardware and maintenance costs decreased while *supporting more* users.

Table 2.7: Examples of compressed sentences from the Ziff–Davis corpus (Knight and Marcu, 2002). *Extractive* is the gold-standard from that corpus; *Abstractive* is included to demonstrate how paraphrasing can impact compression performance. Paraphrased text is indicated in *italics*.

for making fewer changes even if the objective of the task is not fulfilled.

Having identified fair evaluation practices for both extractive *and* paraphrastic sentence compression, Chapter 4 will discuss methods for generating abstractive compressions using paraphrasing. Section 4.1[3] compresses sentences with paraphrasing alone (without deletion), and demonstrates that sentences generated with just paraphrases are more grammatical and retain more meaning than extractive compressions at the same compression rate.

[3] These findings originally appeared in "Paraphrastic Sentence Compression with a Character-based Metric: Tightening without Deletion" (Napoles, Callison-Burch, Ganitkevitch, and Van Durme, 2011).

Section 4.5[4] introduces a hybrid model for sentence compression that uses both deletion and paraphrasing. At its core, this model is a SMT pipeline that has been fully customized for sentence compression. We detail how the pipeline can be adapted for any T2T task, and describe compression-specific modifications that we implement. While earlier works have incorporated MT as a black box in their T2T pipeline, we systematically adapt the MT pipeline for monolingual tasks and demonstrate how to use a paraphrase grammar for MT, with curated feature functions to represent each paraphrase transformation. We show that the MT-based approach is more expressive than a leading extractive model at that time, supporting the use of paraphrasing in sentence compression. Later in this thesis, we will extend this approach for the more complex T2T tasks of simplification (Chapter 6) and grammatical error correction (Chapter 8).

[4] This section first appeared in "Learning Sentential Paraphrases from Bilingual Parallel Corpora for Text-to-Text Generation" (Ganitkevitch, Callison-Burch, Napoles, and Van Durme, 2011), for which my contributions are the use of compression rate in the optimization metric and for evaluation, as well as the evaluation of paraphrastic compressions.

Chapter 3

Unbiased evaluation of sentence compression

A crucial aspect of any task is determining how to evaluate the outcome effectively and efficiently. This chapter surveys previous evaluation methodologies for sentence compression, provides analysis of their shortcomings, and makes concrete recommendations for more effective evaluation. Like other language generation systems, sentence compression systems are evaluated using automatic methods, which are quick and cheap but imprecise, and manual evaluation, which is more reliable but expensive. The majority of compression models before this work focused on *extractive* compressions, but we additionally examine the problems of evaluating paraphrastic compression, for which automatic evaluation is more complicated because simple string matching to a reference is insufficient. We demonstrate that compression rate is a strong predictor of compression *quality* and that perceived improvement over other models is often a side effect of producing longer output. Based on this analysis, we make recommendations for unbiased system comparisons.

Sentence compressions should be judged based on the following three criteria:

1. **Length**. Compressions should have fewer tokens and/or characters than the input. In some situations, a very compact input sentence cannot be shortened any more than it already is. For example, it is difficult to imagine compressing the sentence *He was mad.*

2. **Adequacy**. One important function of a sentence compression system is choosing which part of a sentence can be deleted, which almost always leads to a loss in meaning or nuance. Very few compressions have 100% adequacy, and therefore compressions are evaluated in terms of retention of the *most important* meaning of the input.

3. **Fluency**. Compressed sentences should be grammatical and well-formed. A separate but related task is *headline generation* (e.g., Banko et al., 2000), which has a more lax notion of fluency, since headlines are often incomplete sentences, e.g., *Finding Beauty in Function* (Coleman, 2017).

It is simple to measure length, but *fluency* and *adequacy* are difficult to measure for any artificially generated natural language, and therefore many of the problems identified in this work are relevant for other T2T tasks. (The reader may suggest machine translation metrics for evaluating T2T, however in later chapters (5 and 7) we will provide evidence against their use in a monolingual setting.)

Shared tasks are popular in many areas like machine translation and summarization as a way to compare system performance in an unbiased manner. Unlike other tasks, such as machine translation, there is no shared-task evaluation for compression, even though some compression systems are indirectly evaluated as a part of the Document Understanding Conference (DUC) summarization shared tasks (e.g., Madnani et al., 2007; Zajic et al., 2006). The benefits of shared-task evaluation have been discussed before (e.g., Belz and Kilgarriff, 2006; Reiter and Belz, 2006), and they include comparing systems fairly under the same conditions. Unbiased evaluation, following the guidelines we establish in this chapter, is critical in the absence of shared tasks.

Human evaluation is preferable but frequently not performed due to its cost, making ethical and informative automatic evaluation even more important. We focus primarily on aspects of automatic evaluation, however we will also discuss subtleties to appropriate experimental design for human evaluation, which can give misleading results if not handled properly.

3.1 A Note on Terminology

Sentence compressions are described by their **compression rate**, which is defined as

$$\text{CR} = \frac{\text{Length of candidate sentence}}{\text{Length of source sentence}} \times 100 \qquad (3.1)$$

A compression rate of 90 means that the candidate is 90% the length of the original sentence, or only 10% shorter. Unintuitively, a higher compression rate indicates a longer sentence. In this thesis we will follow convention and refer to the compression rate as a *rate*, even though it would be better described as the *compression ratio*.[1] Length has typically been measured in the number of tokens, however in the following chapter we argue for a character-based compression rate.

3.2 Compression Corpora

We refer the reader to the literature review (Chapter 2) for a more comprehensive discussion of sentence compression, and here provide a brief overview of SC, focusing on evaluation and corpora. There are two types of sentence compressions: extractive and abstractive. **Extractive compression** returns an ordered subset of words from the original sentence that fulfills the three criteria (length, adequacy, fluency), whereas **abstractive compressions** paraphrase the original sentence and may include movement and substitutions in addition to deletions. Because it was developed in support of extractive summarization (Knight and Marcu, 2000), compression has mostly been framed as a deletion task, generating extractive compressions (e.g., McDonald, 2006; Galley and McKeown, 2007; Clarke and Lapata, 2008; Galanis and Androutsopoulos, 2010). There are limited compression corpora due to the infrequency of naturally occurring parallel compressed text. A potential

[1] To our knowledge, the term *compression rate* applied in this way originated in Teufel and Moens (1997), who used it to describe the constant $P(s \in \mathscr{S})$, indicating the probability a sentence is included in an extracted summary (Kupiec et al., 1995, Section 2.2).

source of parallel compressions is in abstracts of full-length articles, such as the Ziff–Davis corpus, which contains news articles related to computers, approximately half of which have human-authored abstracts (Harman and Liberman, 1993). Knight and Marcu (2002) identified 1,067 sentence pairs where a sentence in the abstract was an ordered subset of a sentence in the main article.[2] However, extractive compressions do not occur with high frequency in natural text because humans tend to write abstractive compressions, which are characterized by paraphrases (Marsi et al., 2010). An alternative to this is to create artificial corpora by manually shortening sentences. The Clarke and Lapata (2008) corpus consists of extractive compressions for 3000 sentences from news articles (1443 sentences) and broadcast news transcripts (1370 sentences). These corpora both contain one gold standard for each sentence. Cohn and Lapata (2009) released a small corpus which is better suited: it comprises 30 news articles compressed by one or two annotators with paraphrasing, however it is small with only 575 parallel sentences. Following the completion of the work described in this chapter, a new *abstractive* compression corpus was released, containing more than 6,000 sentences from diverse sources, compressed up to 5 times each (Toutanova et al., 2016).

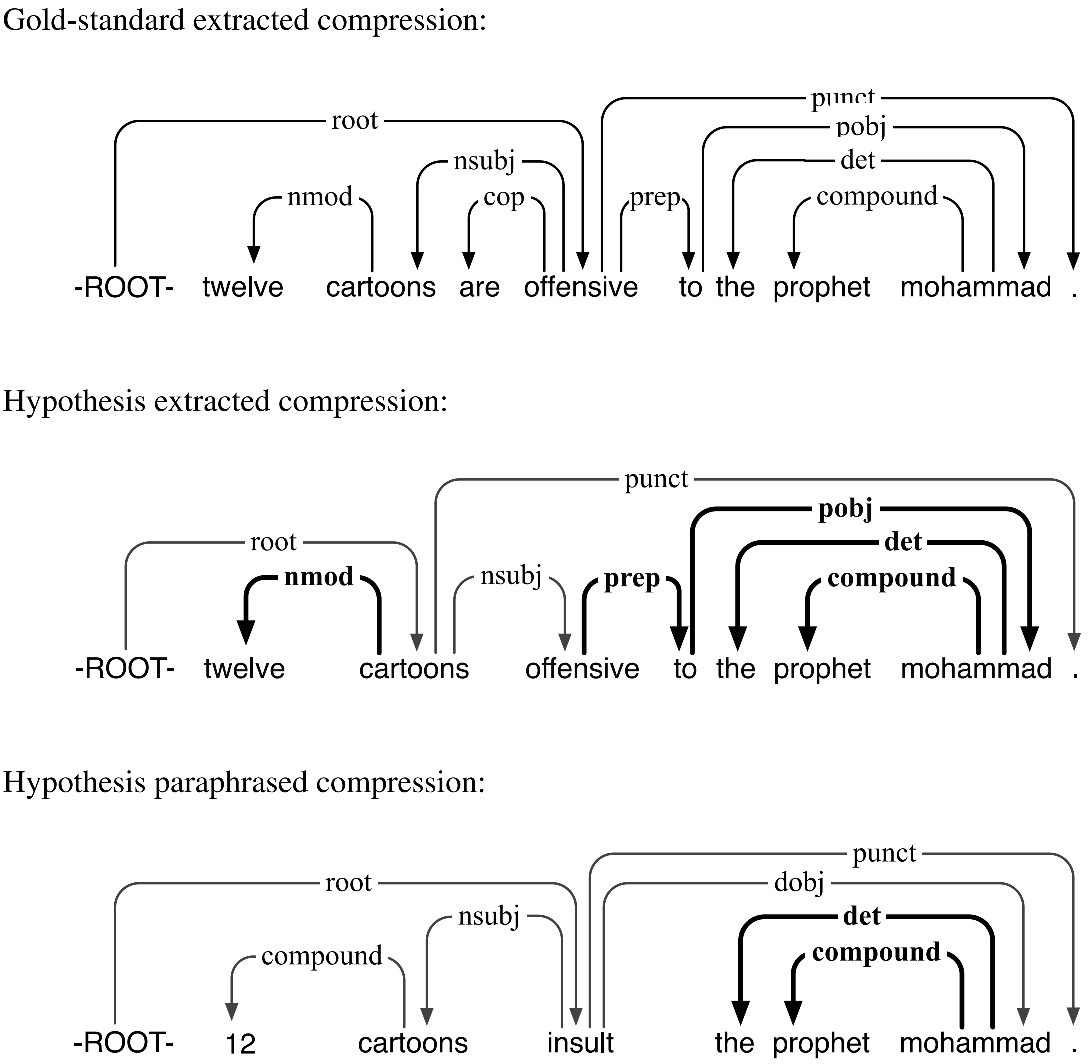
[2] In the context of SC, this extracted corpus is also referred to as the *Ziff–Davis corpus*, not to be confused with the original complete corpus.

3.3 Evaluation Methodologies

While evaluation by human judges is almost always preferable, it is expensive and time consuming, and therefore automatic metrics are often used as a proxy. When possible, the results of both automatic and manual evaluation should be reported. Manual evaluation is more reliable and gives a clearer picture of the system quality, and automatic metric scores provide a measure against which other systems can compare. We will review current practices in automatic and manual evaluation and in the next section identify faulty assumptions of evaluation methodologies, which can skew the result.

Automatic Measures

One of the most widely used automatic metrics for extractive compression is the F_1 measure (Equation 2.2) over grammatical relations of the gold-standard compressions (Riezler et al., 2003), as illustrated in Figure 3.1. F_1 has also been used over token unigrams (Martins and Smith, 2009) and bigrams (Unno et al., 2006). Unno et al. (2006) compared the F_1 measures to BLEU scores (using the gold standard as a single reference) over varying compression rates and found that BLEU behaves similarly to both F_1 measures, which is an unsurprising result because BLEU uses a weighted average of n-gram precision (Papineni et al., 2002). F_1 was also found to have higher correlation with human judgments than Simple String Accuracy, a common metric for evaluating NLG output quality (Bangalore et al., 2000). A syntactic approach to SC evaluation considers the alignment over parse



Gold-standard extracted compression:

Hypothesis extracted compression:

Hypothesis paraphrased compression:

root
nmod
nsubj
cop
prep
punct
pobj
det
compound
-ROOT- twelve cartoons are offensive to the prophet mohammad .
punct
root
nmod
nsubj
prep
pobj
det
compound
-ROOT- twelve cartoons offensive to the prophet mohammad .
punct
root
dobj
nsubj
det
compound
-ROOT- 12 cartoons insult the prophet mohammad .

Figure 3.1: A gold-standard (extractive) compression and candidate extractive and abstractive compressions. The dependency relations in each of the candidates that are also present in the gold dependency parse are bold.

trees (Jing, 2000), and a similar technique has been used with dependency trees to evaluate the quality of sentence fusions (Marsi and Krahmer, 2005). The only metric that has been shown to correlate with human judgments is F_1 over grammatical relations, which are tuples from the dependency parse in the form ⟨relation, governor, dependent⟩ (Clarke and

Lapata, 2006). However, even this is not entirely reliable because it is dependent on parser accuracy and the type of dependency relations used. For instance, the RASP parser uses 16 grammatical dependencies (Briscoe, 2006) while there are over 50 original Stanford Dependencies (Marneffe and Manning, 2008) and 37 Universal Dependencies (De Marneffe et al., 2014, Version 2). Evaluation with fine-grained dependencies would quite possibly result in lower scores than coarse dependencies. These are common methods for evaluation in the community and should not be dismissed, but these potentially confounding factors must be considered. Except for BLEU, all of these evaluations have been applied and tested for evaluating extractive compressions, and as such are unsuitable for abstractive compression. Later in this thesis, we will identify problems with applying BLEU directly to evaluate monolingual T2T and suggest alternatives (Chapters 6 and 7).

Manual Judgments

The goal of automatic metrics is to serve as a proxy for human judgments and therefore, when possible, manual evaluation is preferable to determine semantic and syntactic suitability. The most widely practiced manual evaluation methodology was first used by Knight and Marcu (2002). Judges grade each compressed sentence against the original and make two separate decisions: How grammatical is the compression? How much of the meaning from the original sentence is preserved? Decisions are made along a Likert-like 5-level scale (Likert, 1932). Contextual cues and discourse structure may not be a factor to consider if the sentences are generated for use out of context and except for a few exceptions (e.g., Daumé III and Marcu, 2002; Martins and Smith, 2009; Lin, 2003) most compression systems consider sentences out of context. An example of a context-aware approach considered the summaries formed by shortened sentences and evaluated the compression systems based on how well people could answer questions about the original document from the summaries (Clarke and Lapata, 2007). This technique was previously used to evaluate extractive summarizations (Mani et al., 2002; Morris et al., 1992).

While fluency can be assessed by reviewing the output alone, the original sentence or a reference compression is necessary to judge the adequacy. As we will discuss in the following section, this can influence humans to prefer extractive compressions over abstractive when the extractive compression has higher overlap with the original (or reference) sentence.

3.4 Problematic Assumptions

Having reviewed how compressions are evaluated, we now identify problematic practices. Automatic evaluation operates under two often incorrect assumptions:

The Gold Standard Is Complete and Correct.

In sentence compression, candidates are evaluated against a single gold-standard compression. This makes the faulty assumption that the gold standard is the single best compression, when in fact many equally good compressions may exist. In reality, there may exist acceptable compressions of different lengths or equally good compressions of the same

Length	Original sentence
31 tok. 198 char.	Kaczynski faces charges contained in a 10-count federal indictment naming him as the person responsible for transporting bombs and bomb parts from Montana to California and mailing them to victims .

Length	Gold-standard compressions
17 tok. 113 char.	Kaczynski faces charges naming him responsible for transporting bombs to California and mailing them to victims .
18 tok. 114 char.	Kaczynski faces charges naming him responsible for transporting bombs and bomb parts and mailing them to victims .
18 tok. 113 char.	Kaczynski faces a 10-count federal indictment for transporting bombs and bomb parts and mailing them to victims .

Table 3.1: A sentence and three acceptable extractive compressions created by different annotators.

length expressed in different ways. This holds true for paraphrastic compressions as well as extractive compressions, where multiple sets of deletions may be acceptable. An example of the latter case is found in Table 3.1, which shows three alternate compressions of the same length. For text simplification and grammatical correction, we will also identify instances in which the gold standards are not correct in that they do not exhibit the respective task objectives.

Multiple gold standards would provide references at different compression lengths and reflect different deletion choices (see Section 3.5.1). Since no large corpus with multiple gold standards exists to our knowledge, systems could instead report the quality of compressions at several different compression rates, as Nomoto (2008) did. Alternatively, to fix a length for the purpose of evaluation, systems could evaluate compressions that are of a similar length as the gold standard compression.[3] Output length is controlled for evaluation in some other areas, notably the DUC tasks.

Text Is Compressed with Deletions and Not Substitution.

Several approaches to compression introduce reordering and paraphrase operations (e.g., Cohn and Lapata, 2008; Woodsend et al., 2010; Napoles et al., 2011b; Ganitkevitch et al., 2011). For paraphrastic compressions, manual evaluation alone reliably determines the compression quality. Because automatic evaluation metrics compare candidates to extractive gold standards, they cannot be applied to paraphrastic compression. To apply automatic techniques to substitution-based compression, one would need a gold-standard set of paraphrastic compressions. These are rare. Cohn and Lapata (2009) created an abstractive corpus, which contains word reordering and paraphrasing in addition to deletion, however this corpus is small (575 sentences) and only includes one possible compression for each sentence.

In manual evaluation, bias can be introduced in the following ways:

Experimental Setup

We identify two sources of bias dependent on how candidates are presented to judges. (Details about these experiments are found in Appendix A.4.) First, when the compression is

[3] This obviously prevents a system from deciding the extent of compression, but the target rate is normally hard-coded anyway.

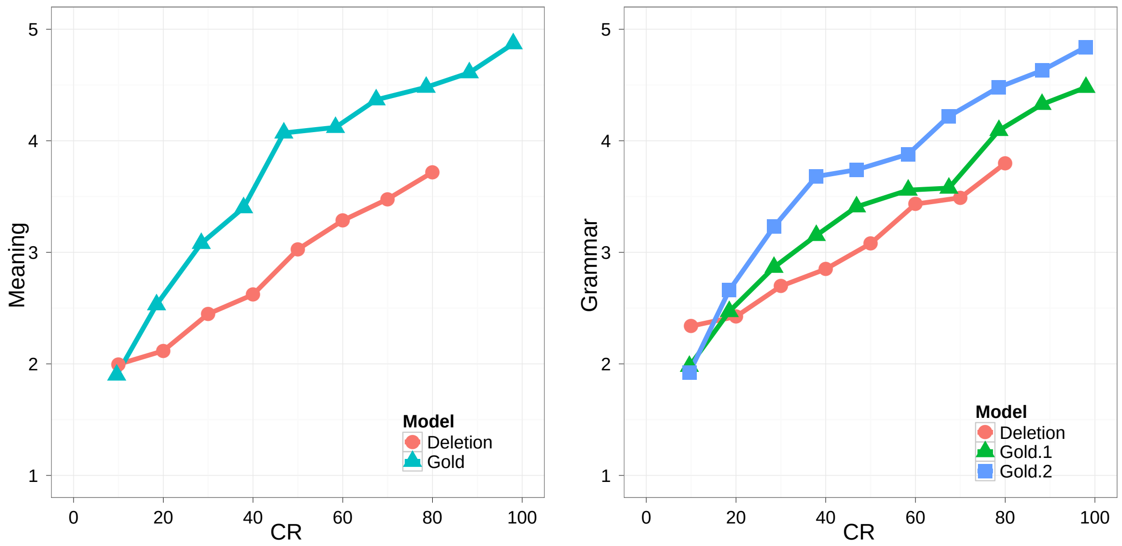


Figure 3.2: Compression rate strongly correlates with human judgments of meaning and grammaticality. *Gold* represents gold-standard compression and *Deletion* the results of a leading deletion model. Gold.1 grammar judgments were made alongside the original sentence and Gold.2 were made in isolation.

presented alongside the original sentence, perceived grammaticality decreases. Figure 3.2 shows that the mean grammar rating for the same compressions is on average about 0.3 points higher when the compression is judged in isolation, without showing the original sentence. Fluency judgments can be made without comparison to the original or a reference statement. Adequacy can be measured by comparing the compression to the original sentence.[4] Secondly, when comparing extractive and paraphrastic candidate compressions, judges prefer sentences with higher overlap with the input sentence, which is almost always the extractive version.

[4] Before 2016, the adequacy of MT systems was judged by comparing the candidate translation to a reference instead of to the original sentence, which would require human raters fluent in both languages. Graham et al. (2017) identified this framework led to a reference-bias, and subsequent WMT evaluations have used bilingual raters who can compare the candidate to the original sentence (Bojar et al., 2016).

	Originally reported			Consistent Compression Rates		
Model	**CR** (%)	**Meaning**	**Grammar**	**CR** (%)	**Meaning**	**Grammar**
C&L′	78	**3.76**	**3.53**	64	3.83	3.66
McD	69	3.50	3.17	64	**3.94**	**3.87**

Table 3.2: Mean quality ratings of two competing models when the compression rates are mismatched versus consistent. The first three columns as originally reported by Clarke and Lapata (2008), with non-equivalent compression rates.

Different Groups of Judges

Another factor is the group of judges. The *McD* entries in Table 3.2[5] represent the same set of sentences generated from the same model evaluated by two different sets of judges, one of which was trained and familiar with the task, and the other was an untrained group of crowdsourced workers (collected using the framework described in Appendix A.4). The mean grammar and meaning ratings in each evaluation setup differ by 0.5–0.7 points.

3.5 Recommendations

We make the following recommendations for fair and informative evaluation of sentence compression:

Multiple Reference Corpora

SC candidates should be evaluated on test sets with multiple references in order to capture allowable variations since there are often many equally valid ways of shortening a sentence.

[5] Specifics about these models will be explained in Section 3.4.

Multiple references should also enable comparisons at different compression rates. We have created a small multi-reference gold-standard collection of 50 sentences at 10 different compression rates, which can be used for testing to benchmark systems (Section 3.5.1). Other alternatives include deriving such corpora from existing corpora of multi-reference translations. The longest reference translation can be paired with the shortest reference to represent a long sentence and corresponding paraphrased gold-standard compression. A similar approach was used in Zhao et al. (2009), who treated the first in a set of multiple-reference translations as the source and the second as the reference for paraphrase evaluation. Additionally, this corpus could be used as a reference for human evaluation to prevent biasing judges towards extractive compressions that contain more overlap with the original sentence than abstractive compressions. In Chapter 4, we address this issue by compressing the longest sentence from each set of reference translations (Huang et al., 2002) and randomly choosing a sentence from the set of reference translations to use as the standard for comparison. In the absence of a multiple-reference corpus, models should be tested on the same sentences, because different corpora will likely have different features that make them easier or harder to compress.

Character-Based Lengths

Few existing efforts have used character-based constraints. Except for DUC 2004 (Over and Yen, 2004), which imposed a character length limit, other DUC tasks had only word limits. Character lengths have been used for application-driven methods, such as subtitling (Glickman et al., 2006) and summarizing for mobile devices (Corston-Oliver, 2001). Although in the past strict word limits have been imposed for various documents, information transmitted electronically is often limited by the number of bytes, which directly relates to the number of characters (1 byte = 1 character). Mobile devices, SMS messages, and microblogging sites such as Twitter are essential for quickly spreading information, and mobile devices have become a standard means of consuming text. In this context, it is important to consider character-based constraints. For extractive techniques, character length is not as relevant because the length of words in the source cannot be shortened. The example in Table 3.1 shows three different extractions that have similar character and token lengths. However, paraphrasing allows original text to be replaced with shorter words and phrases, in which case the token length may not be compressed even though there are fewer characters. As models move beyond exclusive reliance on word-deletion, we suggest switching to characters as the basic unit of measurement. This concept will be demonstrated in the following section (4.1).

Automatic Metrics

In addition to the multiple references identified above, paraphrastic compressions would require different automatic metrics. ROUGE or BLEU could be applied to a set of multiple-reference compressions, although BLEU is not without its own shortcomings (Callison-Burch et al., 2006). One benefit of both ROUGE and BLEU is that they are based on *n*-gram recall and precision (respectively) instead of word-error rate, so reordering and word substitutions can be evaluated. Dorr et al. (2003) used BLEU for evaluating candidates in headline generation which exhibited rewording, and headline generation is closely related to sentence compression.

Consistency in Manual Setup

How candidate sentences are presented to human judges affects their ratings, and so researchers should be careful to state when whether the original or a reference compression is included for judging compressions. Annotator instructions should be made available as well, to reduce variability between groups of judges—or output from existing models should be (re-)evaluated against those of a proposed model.

Control Compression Rate

In order to make non-vacuous comparisons of different models, a system also needs to be constrained to produce the same length output as another system, or report results *at least as good* for shorter compressions. Other methods for limiting quality disparities introduced by the compression rate include fixing the target length to that of the gold standard (e.g., Unno et al., 2006). Alternately, results for a system at varying compression levels can be reported as was done in Nomoto (2008), which included results ranging over compression rates 0.50–0.70. This allows for comparing new systems to the existing at specific lengths and should be emulated, if possible, for comparison against systems that cannot control output length. Compression rate is such an important component of SC evaluation that we will discuss it in more detail in the following section.

3.5.1 Compression Rate Predicts Performance

The dominant assumption in compression research is that the system makes the determination about the optimal compression length. For this reason, compression rates can vary drastically across systems. In order to get unbiased evaluations, systems should be compared only when they are compressing at similar rates.

The only previous work that has seriously considered the effect of the compression rate is Unno et al. (2006), which provides results over a range of different compression rates. It seems intuitive that sentence quality diminishes in relation to the compression rate. Each word deleted increases the probability that errors are introduced. However this relies on the assumption that every word carries equal importance for the meaning or grammaticality. Alternately, if some words do not have as much effect on the meaning and/or grammaticality, equally good compressions of the same sentence should exist at different compression rates. To verify this notion, we generated compressions at decreasing compression rates of 250 sentences randomly chosen from the written corpus of Clarke and Lapata (2008), generated by our implementation of a leading extractive compression system (Clarke and Lapata, 2008). We collected human judgments using the Likert-like scales of meaning and grammar described above. Both quality judgments decreased linearly with the compression rate (see *Deletion* in Figure 3.2).

As this behavior could have been an artifact of the particular model employed, we next developed a unique gold-standard corpus for 50 sentences selected at random from the same corpus described above. The authors manually compressed each sentence at compression rates ranging from less than 10 to 100. Using the same setup as before, we collected human judgments of these gold standards to determine an upper bound of perceived quality at a wide range of compression rates. Figure 3.2 demonstrates that meaning and grammar ratings decay more drastically at compression rates below 40 (see *Gold*). Manual analysis suggests that people are often able to practice "creative deletion" to tighten a sentence up to a certain point, before hitting a compression barrier, shortening beyond which leads to significant meaning and grammatically loss.

3.5.2 Mismatched Comparisons

Several results reported in earlier research have not heeded the observations made in Section 3.4. We have observed that a difference in compression rates as small as 5 percentage points can influence the quality ratings by as much as 0.1 points and conclude that *systems must be compared using similar levels of compression*. In particular, if system A's output is of higher quality but longer than system B's, then it is not necessarily the case that A is better than B. Conversely, if B has results at least as good as system A, one can claim that B is better, since B's output is shorter.

Here are some examples in the literature of mismatched comparisons:

- Nomoto (2009) concluded that their system significantly outperformed that of Cohn and Lapata (2008). However, the compression rate of their system ranged from 45 to 74, while the compression rate of Cohn and Lapata (2008) was 35. This claim is unverifiable without further comparison.

- Clarke and Lapata (2008) (an extension of the EMNLP 2007 best paper (Clarke and Lapata, 2007)) reported significantly better results than a competing model (McDonald, 2006) even though the compression rate was 5 points higher. At first glance, this does not seem like a remarkable difference. However, the study evaluated the quality of summaries containing automatically shortened sentences. The average document length in the test set was 20 sentences, and with approximately 24 words per sentence, a typical 65.4% compressed document would have 80 more words than a typical 60.1% McDonald compression. The aggregate loss from 80 words can be considerable (more than 3 uncompressed sentences!), which suggests that this comparison is inconclusive. Furthermore, nearly 10% of the generated sentences were identical to the original sentence—meaning that no compression was done whatsoever. This artificially inflates the results of human rating, because unaltered sentences will have the identical meaning and fluency as the original sentence.

We re-evaluated the leading model of Clarke and Lapata (2008) against the McDonald (2006) model with global constraints but fixed the compression rates to be equal. We randomly selected 100 sentences from that same corpus and generated compressions with our implementation of Clarke and Lapata (2008) (C&L′), fixing the compression rate to be the same as the sentences generated by the McDonald model (MCD). The sentences were then evaluated by human judgments on 5-level scales of grammaticality and meaning retention. At a consistent compression rate, compression rate, humans judged the McD output to be more grammatical and meaning preserving (Table 3.2). Although not statistically significant, this new evaluation reversed the polarity of the results reported by Clarke and Lapata. This again stresses the importance of using similar compression rates to draw accurate conclusions about different models.

An example of unbiased evaluation is found in Cohn and Lapata (2009). In this work, their model achieved results significantly better than the McD system above. Recognizing that their compression rate was about 15 percentage points higher than the competing system, they fixed the target compression rate to one similar to McD's output, and still found significantly better performance using automatic measures. This work is one of the few that controls their output length in order to make an objective comparison. Another example is found in McDonald (2006).

3.6 Best Practices

We have identified the shortcomings of widely practiced evaluation methodologies and provided justification for the following practices in evaluating compressions:

- Compare systems at similar compression rates.
- Provide results across multiple compression rates when possible.
- Report that system A surpasses B if and only if
  - A and B have the same compression rate and A does better than B, or

  - A produces shorter output than B and A does at least as well B.

- New corpora for compression should have multiple gold standards for each sentence.

Many readers may find this discussion to be intuitive, these points have not previously been delineated and are not consistently considered in earlier research. Therefore it is crucial to enumerate them in order to improve the scientific validity of the task.

Chapter 4

Paraphrastic Sentence Compression

While sentence compression has been framed as an extractive task, a few works before this have applied paraphrasing for SC (please refer to the literature review in Section 2.1). Following our discussion of the importance of character lengths in evaluation (Section 3.5), we further motivate the use of a character-based compression rate for generating compressions. We then demonstrate that compressions generated with paraphrasing alone have better adequacy and fluency than extractive compressions at the same compression rate. Later, Section 4.5 extends that approach by further including deletion rules, and lays out the T2T framework that will be revisited later for text simplification (Chapter 6) and grammatical error correction (Chapter 8).

4.1 Compression Without Deletion

We distinguish two non-identical notions of sentence compression: making a sentence substantially shorter by removing less crucial meaning versus “tightening” a sentence by removing unnecessary verbiage or substituting shorter words/phrases without sacrificing semantic content. Previous work in sentence compression has taken the former approach, while related work in subtitling have taken the latter, by making one-word substitutions to decrease character length (Glickman et al., 2006). We present a substitution-only approach to sentence compression which *tightens* a sentence by reducing its character length. While other work has applied paraphrasing to compression, there is no systematic investigation into the relationship between compression quality and presence of paraphrases or deletion operations. Replacing phrases with shorter paraphrases yields compressions as short as 60% of the original length. At high compression rates, paraphrastic compressions outperform a state-of-the-art deletion model in an oracle experiment. For further compression, deleting from oracle paraphrastic compressions preserves more meaning than deletion alone. In either setting, paraphrastic compression shows promise for surpassing deletion-only methods.

While not currently the standard, character-based lengths have been considered before in compression, and we believe that it is relevant for current and future applications (Section 3.5). When considering paraphrastic compressions, we further support the use of character compression rate by showing that paraphrastic compressions with a high token compression rate can have lower character compression rates.

4.2 Sentence Tightening

The distinction between tightening and compression can be illustrated by considering how much space needs to be preserved. In the case of microblogging, often a sentence has just a few too many characters and needs to be “tightened.” On the other hand, if a sentence is much longer than a desired length, more drastic compression is necessary. Some sentences may not be compressible beyond a certain limit with deletion alone. For example, we found that almost 10% of the compressions generated by Clarke and Lapata (2008) were identical to the original sentence. In situations where the sentence *must* meet a minimum length, tightening can be used to meet these requirements.

We examine whether paraphrastic compression allows more information to be conveyed in the same number of characters as deletion-only compressions. For example, the length constraint of Twitter posts (*tweets*) is 140 characters,[1] and the lead sentence of many articles exceed this limit. Table 4.1 contains example ledes shortened for Twitter, contrasting paraphrastic and extractive compressions of the same length. The first set of examples is compressed to 75% of the original length (162 to 122 characters; the first is the original) The compressed tweet is 140 characters, including spaces and a 17-character shortened link to the original article.[2] In contrast, using deletion to compress to the same length is not as

[1] The limit was doubled to 280 characters in 2018.

[2] Taken from the main page of `wsj.com` (Bendavid and Hook, 2011).

Original (162 char.)	Congressional leaders reached a last-gasp agreement Friday to avert a shutdown of the federal government, after days of haggling and tense hours of brinksmanship.
Paraphrase (123 char.)	Congress made a final agreement Fri. to avoid government shutdown, after days of haggling and tense hours of brinkmanship. on.wsj.com/h8N7n1
Delete (122 char.)	Congressional leaders reached agreement Friday to avert a shutdown of federal government, after haggling and tense hours. on.wsj.com/h8N7n1
Original (147 char.)	With 61 senators on board, the repeal of the military's ban on gay service members is looking more likely, although opponents may yet stall action.
Paraphrase (140 char.)	With 61 senators aboard, the end of the military's ban on gay service members seems likely, yet foes may stall action. nyti.ms/eBUBtz #DADT
Delete (139 char.)	With 61 senators, repeal of the military's ban on gay service members is likely, although opponents may stall action. nyti.ms/eBUBtz #DADT

Table 4.1: Two sentences from news articles that were compressed with paraphrasing alone and with deletion.

expressive. The second set has a higher compression rate of 80% (147 characters to 119), and the loss of meaning/fluency in the extractive compression are not as pronounced.[3]

Multi-reference translations provide instances of the natural length variation of human-generated sentences. These translations represent different ways to express the same foreign sentence, so there should be no meaning lost between the different reference translations. We consider the Multiple-Translation Chinese corpus, which contains Mandarin Chinese sentences from news articles along with four human-authored English translations for each (Huang et al., 2002). Comparing the character length of the shortest sentence in each set to the longest sentence, we find that the average compression rate is 80. When

[3] From the main page of `nytimes.com` (Shear, 2010).

CR	**Length**	**Reference**
100	73 char	Britain and France consulted with each other about this crisis in London.
93	68 char	Meanwhile, in London, Britain and France are discussing this crisis.
84	61 char	Meanwhile, Britain and France discussed the crisis in London.
68	50 char	Britain and France discussed the crisis in London.
100	142 char	According to Kyodo News, it is the first occasion that this kind of defenses were rejected by court in lawsuits for War related compensations.
84	119 char	Kyodo News reported that this is the first time such argument has been rejected in lawsuits involving war compensation.
76	108 char	Kyodo News reported that in war compensation lawsuits, this was the first appeal of its kind to be rejected.
71	101 char	As Kyoto Press Agency reports, this is the first suit that is overruled in all war compensation case.

Table 4.2: Example reference sets from the Chinese multiple reference translation corpus. The sentences are in descending order by length, and CR indicates their compression rate compared to the longest sentence in each set.

comparing the shortest sentence to all three other reference sentences, the average compression rate is 86.[4] Table 4.2 contains example reference sets illustrating this phenomenon. This provides evidence that sentences can be tightened to some extent without losing any meaning.

Through the lens of sentence tightening, we consider whether paraphrase substitutions alone can yield compressions competitive with a deletion at the same length. A character-based compression rate is crucial in this framework, as two compressions having the same

[4] This value will vary by collection and with the number of references: for instance, the NIST 2005 Arabic reference set (NIST, 2010) has a mean compression rate of 0.92 with 4 references per set.

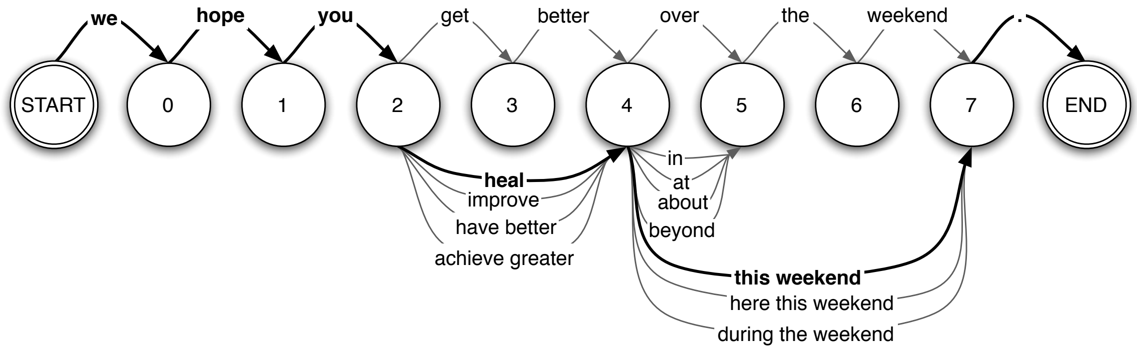


Figure 4.1: A paraphrastic compression lattice, with the path indicating the shortest compression in bold. The cost of each edge is the character length (not shown).

character-based compression rate may have different *word*-based compression rates. The advantage of a character-based substitution model is in choosing shorter words when possible, freeing space for more content words. Going by word length alone would exclude the many paraphrases with fewer characters than the original phrase and the same number of words (or more).

4.3 Framework for Sentence Tightening

Our sentence tightening approach finds the combination of non-overlapping paraphrases that minimizes a sentence's character length. Given a sentence, we construct a lattice of all potential paraphrases of a sentence, where each edge represents a phrase (which can be one word), and the cost of each edge is the number of characters within the phrase. The shortest compression can be found with a shortest-path algorithm through the lattice. Figure 4.1 contains an example.

The paraphrases used in this work were extracted by Juri Ganitkevich from the French-

English Europarl corpus (Koehn, 2005) using the bilingual pivoting method (Bannard and Callison-Burch, 2005) (see Ganitkevitch et al. (2011) for complete details and analysis), resulting in more than 42-million paraphrase pairs. Bilingual pivoting has been shown to capture a greater number of paraphrase transformations than extracting from other sources (Zhao et al., 2008). Bannard and Callison-Burch (2005) ranked paraphrases by the paraphrase probability $p(e_2|e_1)$, which is defined in terms of the translation model probabilities $p(f|e)$ and $p(e|f)$:

$$\begin{aligned} p(e_2|e_1) &= \sum_f p(e_2,f|e_1) \\ &= \sum_f p(e_2|f,e_1)p(f|e_1) \\ &\approx \sum_f p(e_2,f)p(f|e_1). \end{aligned} \tag{4.1}$$

Paraphrases eligible for replacement are thresholded by their paraphrase probability, $p(e_2|e_1)$, and the threshold can be increased for higher precision or lowered for greater recall. Without any threshold, sentences be compressed at a compression rate as low as 60.

Because the lattice approach can generate multiple paraphrased sentences of equal length, we apply two layers of filtering to generate a single output. First, we calculate a word-overlap score between the original and candidate sentences to favor compressions similar to the original sentence. Next, from among the sentences with the highest word overlap, we select the compression with the best language model score.[5]

[5] The language model used in this chapter is a 5-gram model trained on English Gigaword with Kneyser-Ney smoothing.

4.3.1 Scoring Paraphrases

Although many appropriate paraphrases are extracted from parallel corpora, many others are unsuitable and the translation score calculated during pivoting does not always accurately distinguish the two. Therefore, we re-ranked the paraphrase pairs based on their monolingual distributional similarity, employing the method described by Van Durme and Lall (2010) to derive approximate cosine similarity scores over feature counts using single token, independent left and right contexts from a web-scale n-gram corpus (Lin et al., 2010). As 5-grams are the highest order n-gram in this corpus, the allowable set of paraphrases have at most four words (allowing for at least one word of context on either side). To our knowledge this is the first time such techniques have been used in combination in order to derive higher quality paraphrase candidates. Table 4.3 contains an example ranking of a paraphrase by the monolingual similarity (MS) contrasted with the paraphrase probability.

The monolingual-filtering technique we describe is by no means limited to paraphrases extracted from bilingual corpora. It could be applied to other data-driven paraphrasing techniques (see Madnani and Dorr (2010) for a survey), but is particularly well suited to the bilingual extracted corpora, since the information that it adds is orthogonal to that model, and would presumably add less to paraphrasing techniques that already take advantage of monolingual similarity (Pereira et al., 1993; Lin and Pantel, 2001; Barzilay and Lee, 2003). An extension of this work demonstrated the viability of this approach for domain adaptation

Paraphrase	**Monolingual**	**Bilingual**	**CR**
study in detail	1.00	0.70	100
scrutinise	0.94	0.08	67
consider	0.90	0.20	53
keep	0.83	0.03	27
learn	0.57	0.10	33
study	0.42	0.07	33
studied	0.28	0.01	47
studying it in detail	0.16	0.05	140
undertook	0.06	0.06	60

Table 4.3: Candidate paraphrases for *study in detail* with corresponding approximate monolingual cosine similarity (Monolingual), paraphrase probability (Bilingual), and character compression rate.

of paraphrases (Chan et al., 2011).

In order to evaluate monolingual scoring of paraphrase candidates, we randomly selected 1,000 paraphrase sets where the source phrase was present in the corpus described in Clarke and Lapata (2008) and extracted all sentences from that corpus in which the source phrases appeared. A paraphrase set consists of the original *source* phrase and each paraphrase candidate of no more than 4 tokens. We extracted all of the contexts from the corpus in which the source phrase appeared, and human participants ranked each paraphrase based on the extent to which it preserved the meaning and affected the grammaticality of the sentence on two Likert-like scales from 1 to 5, with 5 being perfect (Appendix A.3).

We averaged the scores of each dimension and inferred rankings of each paraphrase set from the scores. We compared the rankings from the mean, grammar, bilingual, and monolingual scores and calculated correlation between the human and automatic scores with

Human	**Bilingual**	**Monolingual**
Grammar	0.15	0.31
Meaning	0.19	0.28

Table 4.4: Correlation between human scores with automatic scores calculated from bilingual and monolingual data. Correlation is computed with Kendall's τ, which operates on the absolute ranking of each paraphrase and not the numerical score.

Kendall's rank correlation coefficient (τ). Both the bilingual paraphrase probability and monolingual similarity positively correlated with human judgments, but there was stronger agreement between the ranking imposed by the monolingual score and human ratings than the original ranking as derived during the bilingual extraction (Table 4.4).

The monolingual scores have stronger correlation with our human ratings of meaning and grammar is higher: $\tau = 0.28$ and 0.31 for monolingual scores and 0.19 and 0.15 for bilingual scores. In our substitution framework, we ignore the translation probabilities and use only the monolingual similarity score in the paraphrase decision task. Higher paraphrase thresholds guarantee more appropriate paraphrases but yield longer compressions. Thresholding the MS score at a cosine similarity of 0.95, the average compression rate is 0.968, which is considerably longer than the compressions using no threshold (0.60).

4.4 Results

In the absence of a multiple reference abstractive compression corpus, we compressed the longest sentence from each set of reference translations (Huang et al., 2002). Paraphrastic compressions were generated at MS thresholds ranging from 0.60 to 0.95. We

System	**CR** (%)	**Cosine threshold**	**Grammar**	**Meaning**
Paraphrase only	97	0.95	3.8	3.7
C&L′	97	–	4.1	4.0
Paraphrase only	89	0.85	3.4	3.2
C&L′	89	–	4.0	3.8
Paraphrase only	85	0.75	3.1	3.0
C&L′	85	–	3.9	3.7
Paraphrase only	82	0.65	2.9	2.9
C&L′	82	–	3.8	3.5

Table 4.5: Mean ratings of compressions using just deletion or substitution at different monolingual cosine similarity thresholds. Deletion performed better in all settings.

re-implemented the deletion model of (Clarke and Lapata, 2008), referred to here as C&L′, to generate deletion-only compressions. The compression length was fixed to ±5 characters of the length of each paraphrastic compression in order to isolate quality from the effect of compression rate (Section 3.4). To avoid potential bias in human judges to favor compressions that have high word overlap with the original sentence, we randomly chose a sentence from the set of reference translations to use as the standard for comparison in human evaluation. We collected meaning and grammar judgments were collected using two 5-point scales (Appendix A.4).

The initial results of our substitution system show room for improvement in future work (Table 4.5). We believe this is due to erroneous paraphrase substitutions, even with the improved scoring technique; phrases with the same syntactic category and distributional similarity are not necessarily semantically identical. Illustrative examples include *WTO* for *United Nations* and *east* or *west* for *south* or *north*. Additionally, manual exam-

Original Sentence
科学家为攸关初期失智症的染色体完成定序
Reference Translations
Scientists Complete Sequencing of the Chromosome Linked to Early Dementia
Scientists Decrypted the Sequence of a Chromosome Crucial to Symptoms of Early Disorders
Scientists Set Sequence For Chromosome Linked To Early-Stage Low IQ
Scientists Completed the Study on the Order of Chromosomes Critical to Early Arzhermes Syndrome

Table 4.6: An example Chinese sentence and English translations from Huang et al. (2002).

ination revealed that the quality of the multi-reference translations is not uniformly high, which would affect the human ratings of all system outputs. Table 4.6 illustrates one such low-quality reference set. The original Chinese sentence is about early-onset Alzheimer's, and only the first reference translation correctly is appropriate but not perfect. *Dementia* is an acceptable replacement for *Alzheimer's*, however *Early* is ambiguous and does not necessarily mean early onset. The subsequent translations do not appropriately identify Alzheimer's or have a spelling error, in the case of the last translation.

To control against these errors and test the viability of a substitution-only approach, we performed an experiment using oracle paraphrases on a set of 20 sentences. We compared three models: paraphrase-only with oracle paraphrases, deletion only (C&L′), and a pipeline of oracle paraphrases then deletion.

In order to identify oracle paraphrases, we generated all possible paraphrase substitutions above a threshold of MS ≥ 0.80 within a set of 20 randomly chosen sentences from the written corpus of Clarke and Lapata (2008). This data set is cleaner than the multiple-reference sentences (although it only contains one gold standard compression per sentence and would not be suitable for evaluating paraphrastic compressions). We solicited humans to make a ternary decision of whether a paraphrase was (*good*, *bad*, or *not sure* in the context (Appendix A.3). We applied our model to generate compressions using only paraphrase substitutions on which all three annotators agreed that the paraphrase was *good*. The oracle generated compressions with an average compression rate of 0.90.

On the same set of original sentences, we used the C&L$'$ deletion model to generate compressions constrained to ±5 characters of the length of the oracle compression. Next, we examined whether passing paraphrastic compressions to C&L$'$ would improve compression quality. In manual evaluation (Appendix A.4) along the dimensions of grammar and meaning, both the oracle compressions and oracle-plus-deletion compressions outperformed the deletion-only compressions at uniform lengths (Table 4.7). Paraphrastic compressions were rated significantly higher for meaning by the sign test, $p < 0.05$. These results suggest that improvements in paraphrase acquisition will make our system competitive with deletion-only models.

This work shows promise for the use of paraphrasing in the task of sentence compression. The oracle results show that a better paraphrase selection technique coupled with deletion outperforms the state-of-the-art deletion model. In the next section, we build upon these findings with a joint model of deletion and paraphrasing that optimizes for more

Model	Grammar	Meaning	CR
Oracle paraphrase	4.1	4.3	90
C&L′	4.0	4.1	90
Oracle → C&L′	3.4	3.7	80
C&L′	3.2	3.4	80
Gold	4.3	3.8	75

Table 4.7: Mean ratings of compressions generated by a paraphrase oracle, deletion only (C&L′), deletion on the oracle paraphrastic compression, and the gold standard.

sophisticated features than length.

4.5 A Joint Model: Compressing with Deletion and Paraphrasing

Supported by the oracle results in the last section, here we describe a joint paraphrase and deletion model for compressing sentences, which is inspired by machine translation and will serve as an inspiration for subsequent models in this thesis.[6] We will discuss how this framework can be adapted to many text generation tasks by augmenting its feature set, optimizing to a specifically tailored metric, and selectively generating artificial translation rules. This model outperforms earlier deletion and paraphrasing methods for sentence compression.

A set of paraphrases can be seen as a (monolingual) translation grammar, with the original phrase analogous to the source language and the paraphrase the target language.

[6] Work originally published as "Learning Sentential Paraphrases from Bilingual Parallel Corpora for Text-to-Text Generation" (Ganitkevitch, Callison-Burch, Napoles, and Van Durme, 2011), which was joint work with Juri Ganitkevich, who developed this approach for sentential paraphrasing.

While the paraphrases first used in Section 4.3 are not specific to compression, it is straightforward to choose lexical paraphrases satisfying the length constraint, as demonstrated in Section 4.4. This section describes a model that harnesses SMT advancements, namely in decoding and parameter optimization, for compressing text using a rich source of paraphrases and artificially generated deletion rules.

Under this framework, we adapt the following components of a SMT pipeline for monolingual sentential paraphrasing:

1. A monolingual synchronous **grammar** that contains transformations reflecting the goals of the task.

2. Task-specific **feature functions** to score each rule, the weights of which are optimized to a specific objective function.

3. An **objective function** that scores how well the output of the model meets task constraints. In MT, this would be a metric such as BLEU or TER, for example.

4. A **development set** of sentence pairs exhibiting transformations fulfilling the task for parameter optimization.

Later chapters will describe how to extend this framework to the more complex of tasks of simplification (Chapter 6) and grammatical error correction (Chapter 8).

We now explain each of these aspects and provide examples of how they can be adapted for sentence compression.

1. Grammar

We will use the paraphrase grammar described in Section 4.3 for our model. Since these paraphrases were extracted from a large corpus of bilingual text, we expect many types of transformations to be present, a subset of which will be appropriate for sentence compression. However, the SCFG formalism does not accommodate deletions of constituents, as was present in the tree-transduction model of Cohn and Lapata (2007), and therefore we experiment with augmenting the paraphrase grammar with a small set of artificial rules. Recall that a synchronous paraphrase rule has the form:

$$\mathrm{X} \rightarrow \langle \mathrm{LHS}, \mathrm{RHS} \rangle \tag{4.2}$$

A deletion rule of the following form is asyncrhonous and therefore not present in the paraphrase grammar, so we artificially generate deletion rules for each adjective, adverb, and determiner in our paraphrase corpus.

$$\mathrm{X} \rightarrow \langle \mathrm{LHS}, \varepsilon \rangle \tag{4.3}$$

2. Task-Specific Feature Functions

Each paraphrase rule has a corresponding series of scores returned by feature functions $\vec{\varphi} = \{\varphi_1, \ldots, \varphi_n\}$. As a result of the paraphrase extraction procedure, several MT features are already assigned, such as translation probability, and the paraphrase probability. These features represent the paraphrase quality but not the suitability of the paraphrase for the constraints of a given task. Therefore, to guide the decoder to choose compressing paraphrases, we augment each grammar rule with additional length-aware features. Specifically:

- The count of words in the source (c_s) and the target side (c_t).

$$c_s = \text{count}(word,\ s)$$
$$c_t = \text{count}(word,\ t)$$

- The difference between the word counts.

$$c_{\Delta_{tok}} = c_t - c_s$$

- The difference between the average word length (in characters) between the source and target sides.

$$c_{\Delta_{char}} = \frac{\text{count}(char, t)}{c_t} - \frac{\text{count}(char, s)}{c_s}$$

We do not include approximate monolingual distributional similarity as a feature function in this model since our grammar contains deletion and synchronous operations such as movement for which it is not possible to calculate distributional similarity.

The translation and compression-specific features are all calculated specific to each transformation (*stateless features*), however the decoder also uses *stateful features* that depend on earlier decoding decisions. Our model uses a language model, the weight of which is also set during parameter optimization.

3. Objective Function

In SMT, the objective function scores candidate translations from the decoder by comparing them to a reference or set of reference translations. Each feature $\vec{\varphi}$ is combined in a log linear model, the parameters of which are set to optimize the objective function. Because our framework has its basis in MT, the obvious metric for parameter optimization would be a metric such as BLEU. BLEU compares the n-grams in common between the candidate and reference translations and has been shown to capture both fluency and adequacy (Papineni et al., 2002). For a candidate C and a reference R with lengths c and r, BLEU is defined as:

$$\mathrm{BLEU}(C,R) = \mathrm{BP} \cdot \exp\left(\sum_{n=1}^{N} w_n \log p_n\right) \tag{4.4}$$

where p_n is the modified n-gram precision of C against R, where typically the highest order n-gram is $N = 4$ and weights are uniform, $w_n = \frac{1}{N}$. BP is the *brevity penalty*, which is added to penalize short candidates (in an extreme example, short candidate can have perfect precision while it is a bad translation for only representing a small portion of the output).

$$\mathrm{BP} = \begin{cases} 1 & \text{if } c > r \\ e^{(1-r/c)} & \text{otherwise} \end{cases} \tag{4.5}$$

BLEU is an ideal metric for optimization because it represents two of the criteria of sentence compression (fluency, adequacy) and it can be calculated efficiently. In the case of monolingual T2T, optimizing for BLEU will result in a system that only produces minimal paraphrases because there is significant overlap between the source and reference sentences (as is especially evident with extractive compression). BLEU is additionally not aware of the relative length of the candidate and therefore is not guaranteed to produce sentences shorter than the input. To design a metric for SC, we can merely scale BLEU with a "verbosity penalty" which penalizes candidates with length in excess of a desired compression rate. The function, PRÉCIS, is defined as follows:

$$\text{PRÉCIS}_{\lambda,\text{CR}}(S,C,R) = \text{VP} \cdot \text{BLEU}(C,R) \tag{4.6}$$

$$\text{VP} = \begin{cases} e^{\lambda(\text{CR}-c/s)} & \text{if } c/s \leq \text{CR} \\ 1 & \text{otherwise} \end{cases} \tag{4.7}$$

For a source sentence S, output C, and reference compression R (with lengths s, c, and r), PRÉCIS combines the precision estimate of BLEU with a verbosity penalty that penalizes compressions that fail to meet a target compression rate CR. Figure 4.2 illustrates the verbosity penalty of PRÉCIS. The scaling term λ determines how severely to penalize deviations from CR, which was set to $\lambda = 10$ in these experiments.

4. Parallel Data

The final customization we need is a parallel corpus that models compression. As discussed earlier, existing compression corpora contain deletion operations and therefore are not suitable for training a paraphrase system. The abstractive corpus of Cohn and Lapata

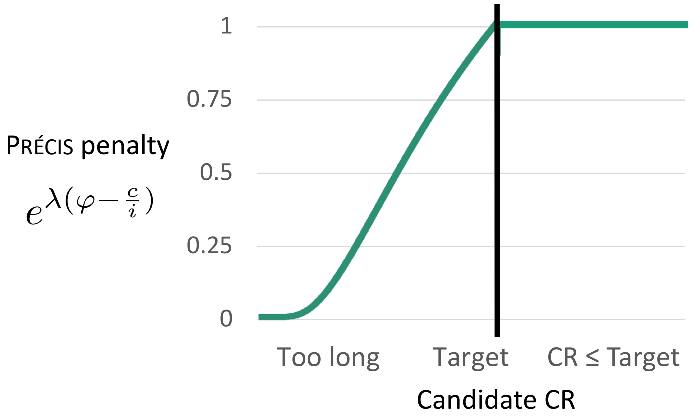


Figure 4.2: The length penalty of PRÉCIS, which penalizes any candidate over the target compression rate.

(2009) is too small for tuning, as it contains only 575 sentence pairs, when parameter optimization in SMT is performed on the order of thousands of sentence pairs. In Section 3.5 we recommend as an existing source of parallel abstractive sentences to use multiple reference translations used for MT evaluation. From a multiple-reference corpus (Huang et al., 2002), we select the longest and shortest sentences to form the input and reference in our tuning set. Of 9,570 sentence pairs, 1,496 have a compression rate within the range of $0.5 < \text{CR} \leq 0.8$. 936 of these sentence pairs are used as a development set and 560 are head out for a blind test set. We tokenize and lowercase the text for all steps of the experiments reported in the next section.

4.5.1 Experimental Results

The following specifications were used to train our model. Parameters were tuned to PRÉCIS with minimal error rate training implemented in the Z-MERT toolkit (Zaidan, 2009). We used the Joshua decoder (Li et al., 2010), making no alterations to the decoder, and a 5-gram English Gigaword language model.

First, we investigate the impact of our customizations to a baseline system using a Hiero grammar, and without syntactic rules, custom features, or augmented artificial grammar rules. We performed human evaluation alone, following the approach described in Section 4.1. Crowdsourced participants judged compressions on two Likert-like scales of grammaticality. Judges are instructed to decide how much the meaning from a reference translation is retained in the compressed sentence, with a score of 5 indicating that all of the important information is present, and 1 being that the compression does not retain any of the original meaning. Similarly, a grammar score of 5 indicates perfect grammaticality, and a grammar score of 1 is assigned to sentences that are entirely ungrammatical (Appendix A.4). Systems were compared only at the same compression rates; specifically, when comparing systems A and B, a sentence was included in the evaluation if $|\textsc{cr}_A - \textsc{cr}_B| < 0.05$.

Table 4.8 contains the results of human evaluation of a set of pairwise system comparisons for $\textsc{cr} \approx 50$.[7] Going from a Hiero-based (Hiero) to a syntactic paraphrase grammar (Syntax) yields greater meaning retention and a significant improvement in grammaticality. Compression-specific features (Syntax+Feat) improves grammaticality even further. Augmenting the grammar with deletion rules (Syntax+Feat.+Aug.) significantly improves the

[7] The compression rates vary for each pairwise comparison due to only including sentences where each pair of sentences reach approximately the same compression rate, and we chose the compression rate for which there were the most sentences.

Model	**CR**	**Meaning**	**Grammar**
Hiero	56	2.57	2.35
Syntax	56	2.76	2.67*
Syntax	53	2.70	2.49
Syntax+Feat.	53	2.71	2.54*
Syntax+Feat.	54	2.79	2.71*
Syntax+Feat.+Aug.	54	2.96*	2.52
Syntax+Feat.+Aug.	52	2.87	2.40
ILP	52	2.83	3.09*
Syntax+Feat.+Aug.	50	2.41	2.20*
T3	50	2.01	1.93

Table 4.8: Human evaluation for variations of our compression system, compared to earlier state-of-the-art models. * indicates a statistically significant difference with a sign test ($p < 0.05$).

core meaning retention but negatively impacts grammaticality.

Finally, we compare our full model to existing state-of-the-art models, the extractive ILP model of Clarke and Lapata (2008) (C&L′) and the tree transduction model of Cohn and Lapata (2007) (T3). We retrain the T3 model on our 936-sentence development set, which is similar in size to the training corpus used in Cohn and Lapata and is also in the same domain as our evaluation set. Our approach significantly outperforms the T3 system but does not match C&L′ in terms of grammaticality (Table 4.8).

Thus far, we have evaluated the model at a substantial compression rate, with the input compressed about 50%. We hypothesize that different systems may perform better at different compression rates, so we next compare Syntax+Feat and C&L′ at a higher compression rate, CR = 80 (Table 4.9). We also include upper and lower bounds of performance, with human evaluation of the reference sentence and a version of the sentence with tokens

Model	**CR**	**Meaning**	**Grammar**
Reference (upper bound)	0.73	4.26	4.35
Syntax+Feat.	0.80	3.67*	3.38
ILP	0.80	3.50	3.49
Random (lower bound)	0.50	1.94	1.57

Table 4.9: Results of the human evaluation on longer compressions: pairwise compression rates (CR), meaning and grammaticality scores. * indicates a statistically significance difference at $p < 0.05$.

randomly deleted. At this compression rate, we significantly outperform ILP in meaning retention ($p < 0.0001$ with the sign test) while achieving comparable results in grammaticality. These results indicate that this framework for paraphrastic transformations performs better than other task-specific models for sentence compression with regards to meaning retention, and the input is similarly grammatical.

Table 4.10 contains example sentences from our test set alongside compressions produced by each system. The paraphrase and ILP systems both produce good-quality compressions, however the paraphrastic compressions have greater meaning retention.

4.6 Conclusion

In this chapter, we have discussed a straightforward approach for paraphrastic sentence compression. We have systematically applied this concept in the development of our systems and comparisons with earlier state-of-the-art approaches. We demonstrate that paraphrastic compressions preserve more meaning than extractive compressions and propose adapting SMT for this task. We have laid out a compelling argument for how our para-

Source	he also expected that he would have a role in the future at the level of the islamic movement across the palestinian territories , even if he was not lucky enough to win in the elections .
Reference CR = 70	he expects to have a future role in the islamic movement in the palestinian territories if he is not successful in the elections .
Syntax+Feat. CR = 88	he also expected that he would have a role in the future of the islamic movement in the palestinian territories , although he was not lucky enough to win elections .
ILP CR = 88	he also expected that he would have a role at the level of the islamic movement , even if he was not lucky enough to win in the elections .
Source	in this war which has carried on for the last 12 days , around 700 palestinians , which include a large number of women and children , have died .
Reference CR = 81	about 700 palestinians , mostly women and children , have been killed in the israeli offensive over the last 12 days .
Syntax+Feat. CR = 71	in this war has done for the last 12 days , around palestinians , including women and children , died .
ILP CR = 71	in this war which has carried for the days palestinians , include a number of women and children died .
Source	hala speaks arabic most of the time with her son , taking consideration that he can speak english with others .
Reference CR = 64	hala speaks to her son mostly in arabic , as he can english to others .
Syntax+Feat. CR = 81	hala speaks arabic most of the time with her son , that he can speak english with others .
ILP CR = 80	hala speaks arabic most of the time , taking into that he can speak english with others .

Table 4.10: Example compressions produced by the two systems in Chapter 4.9.

phrastic system outperforms a leading deletion-based model at moderate and extreme compression rates (CR = 80 and 50). Later chapters demonstrate how this unified framework applies to take to more complex T2T tasks, text simplification (Chapter 6) and grammatical error correction (Chapter 8). As the task increases in complexity, we will particularly be considering the objective function and how more abstraction necessitates an input-aware metric.

Part II

Text Simplification

This part of the thesis examines a T2T task that is more complex than compression. Text simplification is the process of changing vocabulary and grammatical structure to increase the readability of a text while maintaining the original information and content. Automated tools for text simplification are a practical way to make large corpora of text accessible to a wider audience lacking high levels of fluency in the original language. In Chapter 5, we examine a novel source of parallel text for simplification, perform a corpus analysis, and develop a classifier to automatically identify simple English text.[8] We also identify problems with this corpus and create a clean development and test set of 4,000 parallel sentences.

Chapter 6 extends the unified framework introduced in Chapter 4 to simplification, examines measures for automatically evaluating simplified output, and proposes a novel metric. The simplification system we develop is significantly better than other approaches for

[8] Originally presented in "Learning Simple Wikipedia: A Cogitation in Ascertaining Abecedarian Language" (Napoles and Dredze, 2010).

simplicity and meaning retention of the input text, and it does not depend on the availability of a large parallel corpus.[9]

The goal of text simplification (TS) is to increase the readability of a given text, with applications in areas such as education and public health, and safety. TS can also be applied for generating text in an earlier step of a NLP-pipeline, such as information extraction or machine translation, but studies have found that text simplified for machine and human comprehension are categorically different (Chae and Nenkova, 2009). This part considers TS for human readers, but the findings can be applied for either application.

Historically, simplification has been done by hand, which is time consuming and expensive, especially when dealing with material that requires expertise, such as legal documents. Prior to this work, most systems rely on handwritten rules, e.g., PEST (Carroll et al., 1999), its SYSTAR module (Canning et al., 2000), and the method described by Siddharthan (2006).[10] Systems using handwritten rules can be susceptible to changes in domains and need to be modified for each new domain or language. Before this work, there was some research into automatically learning the rules for simplifying text using aligned corpora (Daelemans et al., 2004; Yatskar et al., 2010), but these have yet to match the performance hand-crafted rule systems. Chapter 5 represents one of the earliest explorations of a parallel corpus for simplified English text, Simple English Wikipedia.

Table 4.11 contains an example of a manually simplified text. This example displays

[9] Based on work presented in “Computational Approaches to Shortening and Simplifying Text” (Napoles, 2012).

[10] Subsequent to the work presented in this chapter, several new approaches have been applied to text simplification, including neural machine translation (e.g., Wang et al., 2016).

the three operations into which the process of TS can be divided (Aluísio et al., 2008):

1. *Deleting* unnecessary or distracting text
2. *Substituting* complex lexical and syntactic forms
3. *Inserting* text for further clarification where needed

In this regard, TS is related to several different natural language generate tasks such as summarization, compression, machine translation, and paraphrasing.

Simple English Wikipedia	English Wikipedia
Stephen William Hawking, CH CBE FRS (born January 8 1942) is an English theoretical physicist and mathematician. He is one of the world's leading theoretical physicists.	Stephen William Hawking, CH, CBE, FRS, FRSA (born 8 January 1942[1]) is a British theoretical physicist, whose world-renowned scientific career spans over 40 years.
A theoretical physicist is someone who uses information from experiments to make predictions about the world.	
Hawking writes many science books for the public, or the people who are not scientists.	His books and public appearances have made him an academic celebrity and he is an Honorary Fellow of the Royal Society of Arts,[2] a lifetime member of the Pontifical Academy of Sciences,[3] and in 2009 was awarded the Presidential Medal of Freedom, the highest civilian award in the United States.[4]
Hawking was a professor of mathematics at the University of Cambridge (a position that Isaac Newton once had[1]). He retired on October 1st 2009.[2]	Hawking was the Lucasian Professor of Mathematics at the University of Cambridge for thirty years, taking up the post in 1979 and retiring on 1 October 2009.[5][6]
He has ALS or Lou Gehrig's disease, and because of that he can not move or talk very well. The illness has gotten worse over the years and he is now almost completely paralyzed. He uses a wheelchair to move and an Intel computer to talk for him.	Hawking has a neuro-muscular dystrophy that is related to amyotrophic lateral sclerosis (ALS), a condition that has progressed over the years and has left him almost completely paralysed.
He is one of the most clever living people.	

Table 4.11: The article, *Stephen Hawking*, from Simple English Wikipedia (Wikipedia, 2010a) alongside comparable sentences from the English Wikipedia article (Wikipedia, 2009d). Citations are preserved from the original articles.

Chapter 5

Discriminating between Simple and Complex English

This chapter investigates the potential of Simple English Wikipedia to assist automatic text simplification by identifying the most discriminative features of simple English across multiple domains, and develop a classifier that discriminates *simple* English from *ordinary* English.

Text simplification systems built with hand-written rules face limitations scaling and transferring across domains. The potential for using Simple English (SE) Wikipedia as a comparable corpus for text simplification is significant: it contains nearly 60,000 articles across diverse domains.[1] Articles contain revision histories, which can be mined for simplifying operations. More significantly, many articles are aligned to the original English (OE)

[1] There were nearly 60,000 articles in 2010 at the time of this work. By 2018, Simple English Wikipedia more than doubled in size, containing 133,210 sentences.

Wikipedia. OE and SE Wikipedia do not have a one–one alignment between sentences or even articles, and therefore form a comparable corpus. Some articles in SE Wikipedia have been modified from the OE article, and other work simultaneous to this research extracted aligned sentence pairs from aligned articles (Zhu et al., 2010). This chapter uses statistical learning techniques to identify the most discriminative features of SE and "ordinary" English in Wikpiedia articles and compare these to the simple English authorship guidelines.

The example in Table 4.11 illustrates some differences characteristic of simplifications.

- The sentences in the simplified text have fewer words than the original (5 fewer words on average).
- Syntactic structures and lexical items are simplified.
- Complex sentences are split into multiple simple sentences.
- Some of the simplified text is longer than the original text because illustrative examples are included to fully describe complex terms such as *theoretical physicist* and *the public*.

We capture these differences with cognitively motivated features as well as automatic measurements of a document's lexical, syntactic, and surface features. Our study demonstrates the validity and potential benefits of using SE Wikipedia as a resource for TS research.

The goal of this thesis is not to develop methods for sentence alignment, so therefore we first examine Wikipedia as a comparable corpus and then use a sentence-level alignment developed in separate work (Zhu et al., 2010). We then discuss why an automatically aligned parallel Wikipedia corpus is problematic for developing and evaluating automatic simplification systems, identifying 64% sentence pairs where the target sentence does not represent a simplification of the original. We refine a 4,000-sentence subset of the aligned corpus to create a clean development and test set for simplification.

5.1 Wikipedia as a Corpus

Wikipedia is a unique resource for natural language processing tasks due to its sheer size, accessibility, language diversity, article structure, inter-document links, and inter-language document alignments. Denoyer and Gallinari (2007) introduced the Wikipedia XML Corpus, with 1.5 million documents in eight languages from Wikipedia, that stored the rich structural information of Wikipedia with XML. YAWN (Schenkel et al., 2007), a Wikipedia XML corpus with semantic tags, is another example of exploiting Wikipedia's structural information. Wikipedia has been used as a corpus for nearly all areas of NLP, including word sense disambiguation (Mihalcea, 2007), classification (Gantner and Schmidt-Thieme, 2009), machine translation (Smith et al., 2010), coreference resolution (Versley et al., 2008; Yang and Su, 2007), sentence extraction for summarization (Biadsy et al., 2008), information retrieval (Müller and Gurevych, 2008), semantic role labeling (Ponzetto and Strube, 2006), entity linking (Han et al., 2011), paraphrasing (Yatskar et al., 2010), and grammatical error correction (Grundkiewicz and Junczys-Dowmunt, 2014). Concurrent with the work of Yatskar et al. (2010), this is the first application of Wikipedia for text simplification.

The Simple Wikipedia project[2] was founded in 2003. SE Wikipedia uses basic vocabulary and less complex grammar to make the content of Wikipedia accessible to students, children, adults with learning difficulties, and non-native English speakers. In addition to being a large corpus, these articles are linked to their English Wikipedia counterparts, so for each article both a simple and a more complex version are available. Furthermore, many articles in SE Wikipedia appear to be copied and edited from the corresponding OE Wikipedia article. This information, together with revision history and flags signifying unsimplified text, can provide a scale of information on the text-simplification process previously unavailable. Example sentences from SE Wikipedia and OE Wikipedia are shown in Table 4.11.

We create a large comparable corpus of aligned SE and OE Wikipedia articles for our experiments, across nine domains: Everyday Life, Geography, History, Knowledge, Literature, Media, People, Religion, and Science. A total of 55,433 OE and 42,973 SE articles were extracted and processed. We limit documents to include a minimum of two sentences and strip all elements (including *wiki markup*) except for the main body text. The preprocessing tools used are the Punkt sentence tokenizer (Kiss and Strunk, 2006) in NLTK (Bird and Loper, 2004) and the PCFG parser of Huang and Harper (2009), a modified version of the Berkeley parser (Petrov et al., 2006; Petrov and Klein, 2007).

[2] `http://simple.wikipedia.org/`

5.2 Task Setup

Using the two document collections, we constructed a binary classification task of labeling whether a piece of text was SE or OE. Labels were inherited from the source of the text (SE or OE Wikipedia). We extracted the following cognitively motivated features based on a document's lexical, syntactic, and surface features.

5.2.1 Features

The guidelines for writing Simple Wikipedia pages (Wikipedia, 2009c)[3] suggest

- Only use the 1000 most common and basic English words.
- Words should appear on lists of basic English words, such as the Voice of America Special English words list (America, 2009) or the Ogden Basic English list (Ogden, 1930).
- Use simple grammar and short sentences.
- Write short articles unless they need to explain vocabulary words necessary to understand the topic.
- Avoid idioms, compounds, and the passive voice.

To capture these properties in the text, we created four classes of features: lexical, part-of-speech, surface, and syntactic. Several of our features have previously been used for

[3] The guidelines in 2018 are largely unchanged.

measuring text fluency (Aluísio et al., 2008; Chae and Nenkova, 2009; Feng et al., 2009; Petersen and Ostendorf, 2007).

1. Lexical

Previous work suggests that the document vocabulary is a good predictor of document readability (Feng et al., 2009). Simple texts are more likely to use basic words more often as opposed to more complicated, domain-specific words used in ordinary texts. To capture these features we used a unigram bag-of-words representation. We note that lexical features are unlikely to be useful unless we have access to a large training corpus that allowed the estimation of the relative frequency of words (Chae and Nenkova, 2009). Additionally, we can expect lexical features to be very fragile for cross-domain experiments as they are especially susceptible to changes in domain vocabulary. Nevertheless, we include these features as a baseline in our experiments.

2. Parts of Speech

A clear focus of the simple text guidelines is grammar and word type. One way of representing this information is by measuring the relative frequency of different types of parts of speech. We consider simple unigram part-of-speech tag information. We measured the normalized counts and relative frequency of part-of-speech tags and counts of bigram part-of-speech tags in each piece of text. Word order (subject verb object (SVO), object verb subject (OVS), etc.) is correlated with readability (Devlin and Unthank, 2006), we also

Coarse Tag	**Penn Treebank Tags**
ADJ	JJ, JJR, JJS
ADV	RB, RBR, RBS
DET	DT, PDT
N	NN, NNS, NP, NPS, PRP, FW
V	VB, VBN, VBG, VBP, VBZ, MD
WH	WDT, WP, WP$, WRB

Table 5.1: A mapping of the Penn Treebank tags to a coarse tagset used to generate features.

Feature	**SW**	**EW**
Tokens	158	4332
Types	100	1446
Sentences	10	172
Average sentence length	16	25
Type-token ratio	0.63	0.33
Percent simple words	0.31	0.08
BE850 type-token ratio	0.59	0.67
Not BE850 type-token ratio	0.65	0.30

Table 5.2: A comparison of the feature values from the *Stephen Hawking* article in Simple English Wikipedia (SW) and English Wikipedia (EW).

included a reduced tagset to capture grammatical patterns (Table 5.1). We also included normalized counts of these reduced tags in the model.

3. Surface Features

While lexical items may be important, more general properties can be extracted from the lexical forms. We can also include features that correspond to surface information in the text. These features include document length, sentence length, word length, numbers of lexical types and tokens, and the ratio of types to tokens. All words are labeled as basic or not basic according to Ogden's Basic English 850 (BE850) list (Ogden, 1930).[4] In order to measure the lexical complexity of a document, we include features for the number of BE850 words, the ratio of BE850 words to total words, and the type-token ratio of BE850 and non-BE850 words. Investigating the frequency and productivity of words not in the BE850 list will hopefully improve the flexibility of our model to work across domains and not learn any particular jargon. We also hope that the relative frequency and productivity measures of simple and non-simple words will codify the lexical choices of a sentence while avoiding the aforementioned problems with including specific lexical items.

Table 5.2 shows the difference in some surface statistics in an aligned document from Simple and ordinary Wikipedia. In this example, nearly one-third of the words in the simple document are from the BE850 while less than a tenth of the words in the ordinary document are. Additionally, the productivity of words, particularly non-BE850 words, is much higher in the ordinary document. There are also clear differences in the length of the documents, and on average documents from ordinary Wikipedia are more than four times longer than documents from Simple Wikipedia.

4. Syntactic

As previously mentioned, a number of Wikipedia's writing guidelines focus on general grammatical rules of sentence structure. Evidence of these rules may be captured in the

[4] Ogden proposed a controlled English language with simplified a simplified lexicon and grammar, with the goal of assisting non-native English speakers and advancing English as an international auxiliary language. Ogden also provides extended Basic English vocabulary lists, totaling 2000 Basic English words, but these words tend to be more specialized or domain specific. For the purposes of this study only words in BE850 were used.

Category	**Documents**	**Sentences**
Everyday Life	15,124	7,392
Geography	10,470	5,852
History	5,174	1,644
Literature	992	438
Media	502	429
People	4,326	1,562
Religion	1,863	1,581
Science	25,787	21,054
All	64,238	39,952

Table 5.3: The number of examples available in each category. To compare experiments in each category we used at most 2000 instances in each experiment.

syntactic parse of the sentences in the text. Chae and Nenkova (2009) studied text fluency in the context of machine translation and found strong correlations between parse tree structures and sentence fluency.

In order to represent the structural complexity of the text, we collected extracted features from the parse trees. Our features included the frequency and length of noun phrases, verb phrases, prepositional phrases, and relative clauses (including embedded structures). We also considered relative ratios, such as the ratio of noun to verb phrases, prepositional to noun phrases, and relative clauses to noun phrases. We used the length of the longest noun phrase as a signal of complexity, and we also sought features that measured how typical the sentences were of English text. We included some of the features from the parser reranking work of Charniak and Johnson (2005): the height of the parse tree and the number of right branches from the root of the tree to the furthest right leaf that is not punctuation.

Feature class	**Features**
Lexical	522,153
Part of speech	2,478
tags	*45*
tag pairs	*1,972*
tags (reduced)	*22*
tag pairs (reduced)	*484*
Parse	11
Surface	9

Table 5.4: The number of features in each feature class.

5.3 Experiments

Using the feature sets described above, we evaluated a simple/ordinary text classifier in several settings on each category. First, we considered the task of document classification, where a classifier determines whether a full Wikipedia article was from ordinary English Wikipedia or Simple Wikipedia. For each category of articles, we measured accuracy on this binary classification task using 10-fold cross-validation. In the second setting, we considered the performance of a sentence-level classifier. The classifier labeled each sentence as either ordinary or simple and we report results using 10-fold cross-validation on a random split of the sentences. For both settings we also evaluated a single classifier trained on all categories.

We next considered cross-category performance: how would a classifier trained to detect differences between simple and ordinary examples from one category do when tested on another category? Example text from two different categories, *History* and *Relition* is

Category	Text
History	The British Empire was a global power that contained territories owned by the United Kingdom. The empire was the largest empire in history, and at its peak controlled one quarter of the world's surface. More than 500 million people were brought under the control of the British Empire. Today, most of its members are in the Commonwealth of Nations.
Religion	Confucianism is the philosophy based on the teachings of Confucius (511 BC - 479 BC), who was an important Chinese philosopher. Confucianism has a complex system of moral, social, political, and religious thought, and has had a large influence on the history of Chinese civilization.

Table 5.5: Example text from two categories of Simple English Wikipedia (Wikipedia, 2009a; Wikipedia, 2009b).

included in Table 5.5. In this experiment, we trained a single classifier on data from a single category and used the classifier to label examples from each of the other categories. We report the accuracy on each category in these transfer experiments.

For learning we require a binary classifier training algorithm. We evaluated several learning algorithms for classification and report results for each one:

MIRA a large margin online learning algorithm (Crammer et al., 2006). Online learning algorithms observe examples sequentially and update the current hypothesis after each observation.

Confidence Weighted (CW) learning a probabilistic large margin online learning algorithm (Dredze et al., 2008).

Maximum Entropy a log-linear discriminative classifier (Berger et al., 1996).

Support Vector Machines (SVM) a large margin discriminator (Joachims, 1998b).

For each experiment, we used default settings of the parameters and 10 online iterations for the online methods (MIRA, CW). To create a fair comparison for each category, we limited the number of examples to a maximum of 2000.

5.4 Results

For the first task of document classification, we saw at least 90% mean accuracy with each of the classifiers. Using all features, SVM and Maximum Entropy performed almost perfectly. The online classifiers, CW and MIRA, displayed similar preference to the larger feature sets, lexical and part-of-speech counts. When using just lexical counts, both CW and MIRA were more accurate than the SVM and Maximum Entropy (reporting 92.95% and 86.55% versus 75.00% and 78.75%, respectively). For all classifiers, the models using the counts of part-of-speech tags did better than classifiers trained on the surface features and on the parse features. This is surprising since we expected the surface features to be robust predictors of the document class, mainly because the average ordinary Wikipedia article in our corpus is about four times longer than the average Simple Wikipedia article. We also expected the syntactic features to be a strong predictor of the document class since more complicated parse trees correspond to more complex sentences.

For each classifier, we looked at its performance without its less predictive feature categories, and for CW the inclusion of the surface features decreased performance noticeably. The best CW classifiers used either part-of-speech and lexical features (95.95%) or just part-of-speech features (95.80%). The parse features, which by themselves only yielded

		Feature group			
Classifier	**All features**	**Lexical**	**POS**	**Surface**	**Parse**
CW	86.4	93.0	**95.8**	69.8	64.6
MIRA	**97.5**	86.6	94.6	79.7	66.9
MaxEnt	**99.5**	78.8	96.3	86.9	80.7
SVM	**99.9**	75.0	96.6	89.8	82.7

Table 5.6: Mean accuracy of all classifiers on the document classification task (in percentages).

		Feature group		
Classifier	**All features**	**POS**	**Surface**	**Parse**
CW	73.2	**74.5**	57.4	62.3
MIRA	71.2	**72.7**	56.5	56.5
MaxEnt	**80.8**	77.7	71.3	69.0
SVM	**77.0**	76.4	72.6	73.0

Table 5.7: Mean accuracy of all classifiers on the sentence classification task (in percentages).

64.60% accuracy, when combined with part-of-speech and lexical features showed high accuracy as well (95.60%). MIRA also showed higher accuracy when surface features were not included (from 97.50% mean accuracy with all features to 97.75% with all but surface features).

The best SVM classifier used all four feature classes, but had nearly as good accuracy with just part-of-speech counts and surface features (99.85% mean accuracy) and with surface and parse features (also 99.85% accuracy). Maximum Entropy, on the other hand, improved slightly when the lexical and parse features were not included (from 99.45% mean accuracy with all feature classes to 99.55%).

We examined the weights learned by the classifiers to determine the features that were effective for learning. We selected the features with the highest absolute weight for a MIRA classifier trained on all categories. The most predictive features for document classification were the sentence length (shorter favors Simple), the length of the longest NP (longer favors ordinary), the number of sentences (more favors ordinary), the average number of prepositional phrases and noun phrases per sentence, the height of the parse tree, and the number of adjectives. The most predictive features for sentence classification were the ratio of different tree non-terminals (VP, S, NP, S-Bar) to the number of words in the sentence, the ratio of the total height of the productions in a tree to the height of the tree, and the extent to which the tree was right branching. These features are consistent with the rules described above for simple text.

Next we looked at a pairwise comparison of how the classifiers performed when trained on one category and tested on another. Surprisingly, the results were robust across categories, across classifiers. Using the best feature class as determined in the first task, the average drop in accuracy when trained on each domain was very low across all classifiers (the mean accuracy rate of each cross-category classification was at least 90%). Table 5.8 shows the mean change in accuracy from CW models trained and tested on the same category to the models trained and tested on different categories. When trained on the Everyday Life category, the model actually showed a mean increase in accuracy when predicting other categories.

In the final task, we trained binary classifiers to identify simple sentences in isolation. The mean accuracy was lower for this task than for the document classification task, and

Category	**Mean accuracy change**
Everyday life	+1.4
Geography	−4.3
History	−1.0
Literature	−1.8
Media	−0.6
People	−0.2
Religion	−0.6
Science	−2.5

Table 5.8: Mean accuracy drop for a CW model trained on one category and tested on all other categories. Negative numbers indicate a decrease in performance (in percentage points).

we anticipated individual sentences to be more difficult to classify because each sentence only carries a fraction of the information held in an entire document. It is common to have short, simple sentences as part of ordinary English text, although they will not make up the whole. However results were still promising, with between 72% and 80% mean accuracy. With CW and MIRA, the classifiers benefited from training on all categories, while MaxEnt and SVM in-category and all-category models achieved similar accuracy levels, but the results on cross-category tests were more variable than in the document classification. There was also no consistency across features and classifiers with regard to category-to-category classification. Overall the results of the sentence classification task are encouraging and show promise for detecting individual simple sentences taken out of context.

5.5 Discussion

The classifiers performed robustly for the document-level classification task, although the corpus itself may have biased the model due to the longer average length of ordinary documents, which we tried to address by filtering out articles with only one or two sentences. Cursory inspection suggests that there is overlap between many Simple Wikipedia articles and their corresponding ordinary English articles, since a large number of Simple Wikipedia documents appear to be generated directly from the English Wikipedia articles with more complicated subsections of the documents omitted from the Simple article.

The sentence classification task could be improved by better labeling of sentences. In these experiments, we assumed that every sentence in an ordinary document would be "ordinary" (i.e., not simple) and vice versa for simple documents. However it is not the case that ordinary English text contains only complicated sentences. In future research we can use human annotated sentences for building the classifiers. The features we used in this research suggest that simple text is created from categorical lexical and syntactic replacement, but more complicated, technical, or detailed oriented text may require more rewriting, and would be of more interest in future research.

5.6 Conclusion

We have demonstrated the ability to automatically identify texts as either simple or ordinary at both the document and sentence levels using a variety of features based on the word usage and grammatical structures in text. Our statistical analysis has identified relevant features for this task accessible to computational systems. In the following chapter, we will apply the features to select simplifying paraphrases in our unified T2T framework.

Chapter 6

Sentence Simplification by Paraphrasing

This chapter[1] applies our unified framework to simplification, spelling out the aspects of simplification that are analogous to machine translation, and highlight where it diverges. Unlike previous MT-based simplification systems (see Section 2.2) that derive translation grammars from aligned English and Simple English Wikipedia sentences, we create paraphrases by pivoting over bilingual data that do not directly model simplifications. Our paraphrase grammar contains a rich set of paraphrases and syntactic paraphrase rules, from which simplification rules can be selected. We identify issues with using a machine translation metric, BLEU, for a monolingual T2T, as the unaltered source sentence is a strong baseline that has a high overlap with the reference. We propose a new metric for simplification, GLiB, that penalizes text that is not more readable than the source while also capturing fluency and adequacy with n-gram overlap. We design feature functions that appropriately

[1] Work originally presented in "Computational Approaches to Shortening and Simplifying Text" (Napoles, 2012).

model simplifications and demonstrate that it is better to optimize their weights with GLiB than a standard MT metric. We further re-rank decoder output with additional syntactic grammaticality and readability scores. Our model chooses the best simplification over a large space of possible phrase and sentence-level paraphrases. It increases he readability of sentences while preserving meaning and producing grammatical output, and it outperforms state-of-the-art simplification systems.[2]

Simplification is related to sentence compression, which automatically shortens sentences by deleting words (Chapter 4), because many simplifying operations involve deletion (Coster and Kauchak, 2011b). Other simplifying operations include lexical and syntactic paraphrases. A lexical paraphrase substitutes a complicated word or phrase for a simpler phrase. Simplifying lexical paraphrases might rewrite Latin legal terms into plain English, such as

erratum → *mistake*
affidavit → *sworn statement*
inter alia → *among other things*

Syntactic transformations are more general changes that involve rearranging the syntactic categories or nodes in a parse tree, like in the English possessive rule *the* NN_1 *'s* NNP_2 → *the* NNP_2 *'s* NN_1. Some learned instances of the possessive rule arguably simplify, like *the* NNS_1 *produced by* NNS_2 → NNS_2 *'s* NNS_1. The example in Table 4.11 (page 82) illustrates the diverse requirements of making a text easier to understand.

[2] This work was later extended and published as “Optimizing statistical machine translation for text simplification” (Xu, Napoles, Pavlick, Chen, and Callison-Burch, 2016).

This chapter demonstrates how our unified framework generates paraprhastic simplifications. In contrast to previous work in which simplification rules are handwritten or learned from a parallel simple corpus (Zhu et al., 2010; Bach et al., 2011; Woodsend and Lapata, 2011), our model learns to choose simplifying transformations from a multipurpose paraphrase grammar. We define parameters tailored to improving readability and use an existing statistical MT decoder to generate our output. Finally, we develop a model for re-ranking the decoder output to favor more grammatical and more readable sentences.

Our model, **Sim**plification by **P**araphrasing to **Pl**ainer **E**nglish (SIMPPLE), produces grammatical output that is more readable than the source side, while retaining its meaning. In manual evaluation, participants judge our output to be more grammatical, more readable, and preserve more meaning than leading systems. This chapter will

- show how to adapt the MT-inspired paraphrasing framework for simplifying sentences,
- develop a new metric for optimizing sentence simplicity and grammaticality,
- demonstrate the expressivity of a paraphrase grammar, and
- rerank candidate simplifications with a machine learning approach.

6.1 Adapting the Unified Framework To Simplification

To simplify sentences with our T2T framework, we define custom, rule-level features that represent readability characteristics. By acquiring our paraphrases from a large corpus, we are able to capture many transformations that other approaches do not given their limited nature (limited either by hand-written rules or the corpus size). The following sections describe our customizations to the objective function, grammar, features, and tuning data.

6.1.1 Objective Function

The first step is defining an objective function, to which we optimize the weights of the grammar feature functions. Given a set of feature functions $\vec{\varphi} = \{\varphi_1 \ldots \varphi_n\}$ combined in the following linear model:

$$w = -\sum_{i=1}^{N} \lambda_i \varphi_i, \tag{6.1}$$

we optimize the weights with minimum error rate training (MERT) to iteratively adjust the weight λ_i of each feature function are φ_i until a local optimum is reached. Features used in this work are described below.

BLEU (Equation 4.4, page 71) does not capture the readability of a sentence, and therefore is insufficient for simplification. One of the most widely used metrics for measuring readability is the Flesch–Kincaid Grade Level (Kincaid et al., 1975), which is defined for a document as

$$\text{FKGL} = 0.39\left(\frac{\#\text{ words}}{\#\text{ sentences}}\right) + 11.8\left(\frac{\#\text{ syllables}}{\#\text{ words}}\right) - 15.19, \tag{6.2}$$

with a lower value indicating a higher level of simplicity. The constants were empirically set so that the FKGL roughly equates to the grade at which a student should be able to read a text.

Since we are simplifying sentences out of context, we modify FKGL to evaluate each sentence individually. Additionally, we count tokens instead of words and assign one syllable to each non-word token, to prevent the system from superficially decreasing the grade level by deleting punctuation. This adaptation, GL, is defined as follows:

$$\mathrm{GL}(C) = 0.39c_{tok} + 11.8\left(\frac{c_{syll}}{c_{tok}}\right) - 15.19, \tag{6.3}$$

with c_{tok} the number of tokens in a candidate C and c_{syll} the number of syllables plus the number of non-word tokens.

Pilot experiments used GL as an objective function, but the output was not very fluent, so we also considered *n*-gram overlap with the target sentence, in the form of BLEU. BLEU encourages reasonable output with regard to meaning retention and grammaticality, and we use a penalty dependent on GL, which is sensitive to the readability of a sentence. When we trained using BLEU with a GL penalty, the output was only minimally changed from the source, if at all. The MT system tended to make no changes when optimizing to BLEU since the BLEU score of the source side was already high (0.50, see Table 6.1.4). Therefore, we use *i*BLEU (Sun and Zhou, 2012), a paraphrase-specific extension to BLEU that is *input*-aware. *i*BLEU penalizes *n*-gram overlap with the input source sentence *S* in order to reduce self-paraphrases and promote diversity in the paraphrased output. Given a candidate C and reference R, iBLEU is defined

$$i\text{BLEU}(S,R,C) = \alpha\text{BLEU}(C,R) - (1-\alpha)\text{BLEU}(C,S). \tag{6.4}$$

The weight α adjusts how much penalty source overlap should contribute.

The objective function used for training our SIMPPLE system is a combination of GL and iBLEU, defined as

$$\text{GLiB} = \begin{cases} i\text{BLEU} & \text{if GL}(pp) < target \\ 0 & \text{otherwise} \end{cases} \tag{6.5}$$

We use a [0,1] penalty if the target GL is not reached to strictly enforce more readable output during parameter estimation. The target GL can be set to any value, and for this study the target was always set to the FKGL of the reference sentences. Future work can explore whether different target FKGLs affect the readability of the output.

6.1.2 Grammar

Earlier simplification work (Zhu et al., 2010; Woodsend and Lapata, 2011) learned a translation grammar from the Parallel WiKiPedia corpus(PWKP), which contains 188-thousand aligned sentence pairs from English Wikipedia and Simple English Wikipedia (Zhu et al., 2010). However, this corpus is relatively small when compared to corpora used for MT, so our framework uses the paraphrase grammar extracted with bilingual pivoting (described

Corpus	**Sentences**	**Rules**
Fr-En Europarl	1,695,060	5,014,336
Wikipedia	187,948	700,428

Table 6.1: The size of the corpora from which each grammar was generated and the number of rules used from each for our 100-sentence test set.

in Section 4.3). Using this data, we extracted an order-of-magnitude more rules for decoding our 100-sentence test set than using the PWKP corpus (see Table 6.1). While our paraphrase grammar does not represent any asynchronous transformations, we believe that our paraphrases capture a wide variety of simplifying lexical and syntactic transformations, supported by our findings reported in Section 6.3.

Since automatically extracted paraphrases may contain erroneous substitutions (as first discussed in Section 4.1), we augment the grammar with rules to discourage paraphrasing closed-class words, such as pronouns ($PP \rightarrow \langle \text{he} \mid \text{she} \rangle$), articles ($DT \rightarrow \langle \text{every} \mid \text{any} \rangle$), and prepositions ($IN \rightarrow \langle \text{to} \mid \text{at} \rangle$). We add identity rules with no cost for every word appearing in these classes to encourage identity paraphrases. We further augment the grammar by adding deletion rules that allow adjectives to be deleted since deletion is a common operation in simplification (Coster and Kauchak, 2011b) but an asynchronous operation not present in the paraphrase grammar. The identity rules added to our grammar of the type

$$DT \rightarrow \langle \text{the} \mid \text{the} \rangle$$

include all words of the in the categories DT, IN, and PRP. All adjectives (JJ) are added to the grammar with a deletion rule, such as

$$JJ \rightarrow \langle \text{beautiful} \mid \varepsilon \rangle$$

Even with these augmentations, the grammar is still noisy, but instead of filtering it, we rely on the optimization step to choose well-formed, simplifying rules.

6.1.3 Features

By acquiring our paraphrases from a large corpus, we are able to capture many transformations that other approaches cannot given their limited nature (either by hand-written rules or the corpus size). To illustrate, Table 6.2 compares the paraphrases for *ancient* that are included in the PPDB (an extension of our paraphrase grammar) compared to the single paraphrasing rule learned from a parallel Wikipedia corpus, *ancient* → *old* (Zhu et al., 2010). Using GLiB as our objective function and a set of features specific to this task, we let MERT choose appropriate weights to indicate the simplifying properties of paraphrases.

In order to model simplifying operations, we define a set of rule-level feature functions that calculate readability characteristics over the target-side terminals. Our features are taken from the features from our discriminative classifier described in Chapter 5 that can be easily calculated at the phrase level. These features are the number of terminals, the number of Basic English words,[3] and the average number of syllables per terminal. Additionally, boolean variables indicate when the target side has fewer of each of these counts than the

[3] As defined in Ogden (1930)

ancient	longtime
old	long-standing
former	centuries-old
antique	archaic
age-old	classical
ancestral	old-fashioned
immemorial	antiquated
past	outdated
dated	historical
archaeological	older
traditional	primeval

Table 6.2: Some of the 214 paraphrases in the PPDB for *ancient* (Ganitkevitch and Callison-Burch, 2014).

source, as well as whether the rule is an identity paraphrase or if it is lexical (i.e., each side has no non-terminals). We also use the approximation in Equation 4.1 (page 60) to estimate lexical and phrasal paraphrase probabilities for each rule.

6.1.4 Data

For our experiments, we use the aligned sentences in the PWKP as our corpus. We randomly selected 4000 sentence pairs from PWKP for development. In pilot experiments using this development set, we noticed that the system tended to produce output that was *less* readable than the input.

Because of the collaborative nature of Wikipedia, it contains noise in the form of sentence fragments, ungrammatical text, and poorly formed markup, all of which affect the quality of the corpus. We removed two sentences which were clearly table formatting and another sentence because it was over 100 tokens long. More importantly, there is no guar-

Data	**FKGL**	**Length**
SimpWiki	10.46	21.17
EnWiki	12.50	17.78
FilteredSimpWiki	9.48	16.18
Filtered EnWiki	13.33	23.92

Table 6.3: FKGL and average sentence length of the tuning data, before and after filtering.

antee that a Simple English sentence is more readable than its aligned English sentence, because the PWKP corpus was aligned automatically.

In order to quantify the relative readability of an aligned sentence pair, we asked human judges to compare the remaining 3997 sentence pairs on Amazon's Mechanical Turk (Appendix A.5.1), and we found that 64% of the Simple sentences were perceived to be at least as complicated as the aligned English sentence. Judges made a ternary decision on each sentence pair: sentence A is more readable than sentence B, A is less readable than B, or A and B are of the same difficulty. Each sentence pair was evaluated by three different participants. There were 1434 sentences pairs for which the majority of judges agreed, including 305 where the Simple side was judged to be *less* readable than the English side. We reversed the labels of these pairs.

We used these human-verified 1434 sentences as our tuning corpus and found that the distance between the FKGL of each side of the corpus increased after verification (Table 6.1.4).

We also trained a 5-gram language model on the simple PWKP corpus to model common *n*-grams in a simple corpus, and a 5-gram English Gigaword (Parker et al., 2011) language model to represent grammaticality.

6.2 Ranking Output

In machine translation, the decoder produces translations based on a score over the parameters of the applied transformations. Re-ranking has been investigated as a method for improving MT output, and generally uses orthogonal information, such as edit distance (Akiba et al., 2001), language-model probability (Callison-Burch and Flournoy, 2001; Zhang et al., 2006), and grammatical features (Avramidis et al., 2011).

Because the decoder does not explicitly encode any simplifying syntactic transformations, we use syntactic information to re-rank decoder output. We include features from machine translation, including the unmodified *n*-gram precision for $1 \leq n \leq 4$ and BLEU score, as well as readability features and descriptive surface statistics, borrowed from the classifier developed in the previous chapter. The features we use for scoring the output fall into the following categories:

Surface features include the length of the sentence in tokens and syllables, the average syllables per word, *n*-gram precision for $1 \leq n \leq 4$ compared to a source sentence, BLEU score compared to the source, and GL (16 features in total).

Corpus-based features include the language-model probabilities of the sentence given Gigaword and Simple Wikipedia 5-gram language models, and the number of OOVs in the sentence according to each (4 features).

Syntactic features were used by Charniak and Johnson (2005) for re-ranking parse trees. Some of these features were used in Chapter 5, and a binary classifier trained over the C&J features is a good predictor of grammaticality (Post, 2011). We extract all of the features defined by C&J that appear at least 5 times in our corpus (56k features).

With these features, we trained a discriminative support vector machine with SVM-Light (Joachims, 1998a). Unlike machine translation, we do not have annotated rankings of simplifications to use for training a re-ranker. Therefore, sentences were ranked based on their distance from the decision boundary. This approach is similar to Albrecht and Hwa (2007), who trained regression and discriminative classifiers to identify good machine-translation output, and whose model correlated better with human judgments than some automatic metrics.

Positive examples were the Simple Wikipedia side of our tuning corpus (1434 sentences), and for negative examples we randomly generated 1434 sentences from a trigram Gigaword language model, following Post (2011). In our model, features representing both grammaticality and readability were heavily weighted (see Table 6.4 for the most positive and negative features).

Human judges compared the best translation chosen by the re-ranker to that chosen by the objective function, and judged the unranked translation to be worse 37.67% of the time (unranked translations were better 16.10% of the time and tied with the ranked translations 46.23% of the time). In our experiments, we reranked our system output using this syntactic classifier.

Positive	**Negative**
1,2,3,4-gram precision	Gigaword LM probability
PWKP LM probability	PWKP OOVs
Length	$\langle$VP S$\rangle$
$\langle$S NP VP$\rangle$	SBAR
$\langle$NP DT NN$\rangle$	$\langle$NP NN NN$\rangle$
Grade level	$\langle$NP VBG$\rangle$

Table 6.4: Highly weighted positive and negative features for identifying grammatical, simple sentences.

Model	**BLEU**	**FKGL**	**Meaning**	**Grammar**	**Tokens/ sentence**	**Sentences**
SIMPPLE	0.56	12.50	**4.23**	**3.97**	25.13	100
RevILP	0.34	9.60	4.09	3.90	12.90	183
TSM	0.34	8.12	$3.54^{\dagger}$	$2.80^{\dagger}$	12.09	180
Moses	0.56	12.89	4.16	$2.93^{\dagger}$	23.40	100
Reference	–	9.60	$3.66^{\dagger}$	4.54	16.16	131
Source	0.50	14.18	–	–	25.49	100

Table 6.5: Automatic and manual measures of system performance. $\dagger$ indicates SIMPPLE is significantly better with $p < 0.001$.

6.3 Experiments

In our T2T framework, we treated existing statistical machine translation tools as a black box for parameter estimation and decoding: Z-MERT (Zaidan, 2009) and Joshua (Weese et al., 2011), respectively.

A variety of features help the decoder score its output. These features include standard MT parameters, paraphrase probabilities, and features we defined for this task (Section 6.1.3). MERT determines an optimal weight vector for these features, using GLiB as an objective function (Equation 6.5, $\alpha = 0.95$). The decoder outputs n candidates with the

System	Output
Source	Many consider the flavor to be very agreeable , but it is generally bitter if steeped in boiling water , so it is made using hot but not boiling water .
SIMPPLE	some people think the flavor to be very nice , but it is often bitter if steeped in boiling water , so it is made using hot but not boiling water .
RevILP	Many say the flavor to be very agreeable . It is made using hot but not boiling water .
TSM	Many consider the flavor to be agreeable , but it is horrible . If steeped in boiling water .
Moses	many consider the flavor to be very agreeable , but it is generally bitter if steeped in boiling water , so it is made using hot but not boiling water .
Reference	Many consider the flavor to be very agreeable , but it is generally bitter if steeped in boiling water . It is therefore made using hot but not boiling water .
Source	The combination of new weapons and tactics have caused many historians to consider this battle the beginning of the end of chivalry.
SIMPPLE	The mix of weapons and tactics have caused many historians to see this battle the beginning of the end of chivalry .
TSM	The combination of weapons and tactics have caused many historians. To consider this battle the beginning of the end of knight.
RevILP	It was The combination of new weapons and tactics . It have caused many historians to consider this battle the start of the end of chivalry .
Moses	the combination of new weapons and some historians consider this battle the beginning of the end of chivalry .
Reference	Because of this change some historians call this battle the beginning of the end of chivalry.

Table 6.6: Sample output from different systems.

lowest cost according to the weighted feature functions. We set the target GL to be the average TKGL of the target side of the tuning corpus (9.48).

We evaluated SIMPPLE on the PWKP test corpus, which was used to test the earlier TSM and RevILP models described in Section 2.2. We evaluated the output of SIMPPLE against RevILP and TSM. As a baseline, we used the Moses machine translation system (Koehn et al., 2007), with a translation grammar extracted from the PWKP corpus and optimizing to BLEU. For automatic evaluation we calculated the BLEU and FKGL scores.

Following previous work in evaluating sentence compression described in Chapter 3, we solicited manual judgments of the grammaticality and meaning retention of the simplifications (Appendix A.5.2). Judges were presented with the source sentence and each simplification, and evaluated them on two Likert-like five-point scales of meaning retention and grammaticality (5 being the best). All of our experiments were carried out using Mechanical Turk, with three-way redundancy and controls to filter out spammers. Table 6.5 contains a summary of the automatic and manual evaluation results, and example output is found in Table 6.2.

Our next experiment evaluated human perceptions of readability. For each source sentence, the output of all four systems and the reference sentence were presented in random order. We asked participants to rank the simplifications in order of how readable they were, allowing ties. In a pairwise comparison of system rankings, SIMPPLE was found to have more readable output compared to the other systems (see Figure 6.1). SIMPPLE was significantly ranked more readable than Moses and TSM ($p < 0.001$).

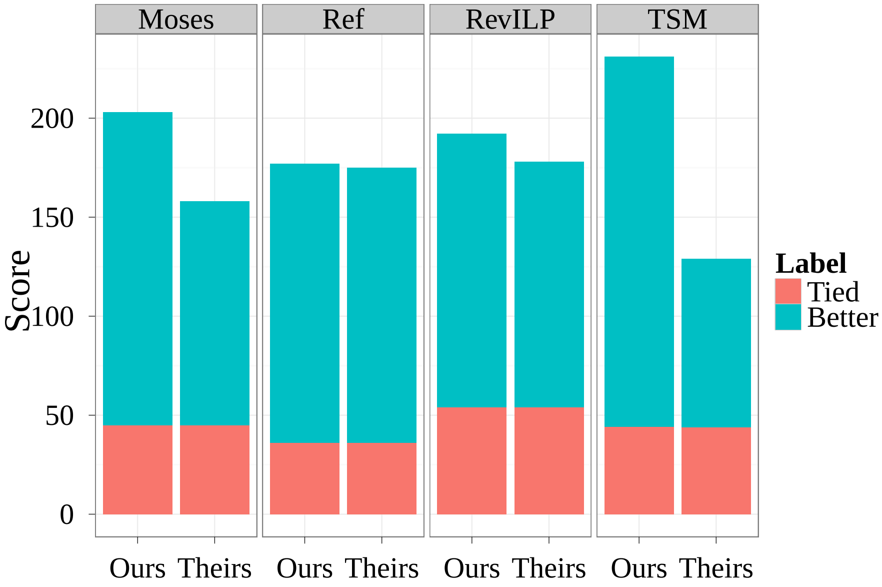


Figure 6.1: Pairwise comparison of how human judges ranked the readability of different systems; across the board SIMPPLE was preferred more often.

6.4 Analysis

In manual evaluation, SIMPPLE was perceived to be more grammatical and preserve more meaning than competing systems. These results are statistically significant for TSM (meaning and grammar), and Moses (grammar). Compared to RevILP, our results were only significant at $p < 0.1$ for meaning. SIMPPLE also had significantly higher average meaning retention than the reference sentences, but we imagine this was because the reference sentences were on average shorter than the SIMPPLE output (Napoles et al., 2011a). For automatic metrics, Moses and SIMPPLE had higher BLEU scores than the other systems, most likely because they optimize to BLEU, directly or indirectly. The FKGL of SIMPPLE and Moses was lower than that of the source sentence, but not nearly as low as that of

RevILP and TSM. The latter two methods both used quasi-synchronous grammars, which allowed sentence splitting, and for a given number of words, FKGL is lower when there are more sentences (see Equation (6.2)). However, humans still ranked SIMPPLE output more readable than all other systems the majority of the time, even without asynchronous, sentence-splitting transformations. These results provide strong evidence in favor of taking a machine translation approach to monolingual text-to-text generation.

6.5 Conclusion

This chapter demonstrates how our universal T2T framework can be extended to a complex text-to-text generation task. This framework outperformed existing automatic text simplification systems by using a rich set of transformation rules to improve sentence simplification and defined a small set of features and a metric by which text can be simplified. Our manual evaluation does not consider how well SIMPPLE aids people with specific readability needs. However, the parameter of the objective function can be adjusted to favor more drastic increases in readability. Our T2T framework is a rich, data-driven model that can be further improved and customized. We have shown how the framework creates better transformations that are grammatical, meaning-preserving, and further meet fulfill the goal of each task. In the next part, we turn to ill-formed text and apply the framework to grammatical error correction.

Part III

Grammatical Error Correction

We turn to the final monolingual rewriting task, grammatical error correction (GEC). Unlike simplification and compression, the input to GEC is ungrammatical, and a general-purpose paraphrase resource will not contain corrections to most of the errors. In Chapter 7, we will introduce the task, discuss existing evaluation practices and resources for the task, and argue for a different approach, presenting new metrics and parallel corpora to support the new approach. Chapter 8 applies methods from machine translation to GEC, highlighting the impact that artificially generated data can make when training data is limited.

The field of grammatical error correction (GEC) has grown substantially in the 2010s, with several shared tasks (Dale and Kilgarriff, 2011; Dale et al., 2012; Ng et al., 2013; Ng et al., 2014) and GEC models that perform whole-sentence correction (e.g., Junczys-Dowmunt and Grundkiewicz, 2016) instead of targeting specific errors such as prepositions (e.g., Chodorow and Leacock, 2000; Dale and Kilgarriff, 2011; Leacock et al., 2014)). Approaches to grammatical error correction have been developed and evaluated with *error-coded* corpora, which contain corrections to discretely labeled spans of text, and research has focused on developing better evaluation metrics. However, the field has not questioned the reliance of GEC evaluation on *error-coded* corpora, which contain labeled corrections. We examine current practices and show that GEC's reliance on such corpora unnaturally constrains annotation and automatic evaluation, resulting in (a) sentences that do not sound acceptable to native speakers and (b) system rankings that do not correlate with human judgments. In light of this, we propose an alternate approach using *fluency edits*, which more closely reflect human perceptions of grammaticality and are substantially cheaper to collect.

Fluency edits can be viewed as a type of paraphrase, and we adapt our unified T2T framework to grammatical error correction, focusing on holistic fluency correction instead of local error-coded operations. Our framework rivals the current state-of-the-art system but requires half the training data, demonstrating the expressivity and robustness of this framework for diverse T2T tasks with different requirements.

Chapter 7

Reframing Grammatical Error Correction

This chapter motivates the need for fluency corrections. We describe the creation of fluency corpora for evaluating GEC (Section 7.1) and new metrics that work with either error-coded or fluency edits (Section 7.4). We demonstrate that automatic evaluation with fluency corrections and a new fluency-centered metric have very strong correlation with human expert judgments ($\rho = 0.82$).[1]

GEC is often viewed as a matter of correcting isolated grammatical errors but is much more complicated, nuanced, and subjective than that. As discussed in Chodorow et al.

[1] This chapter is based on work done in collaboration with Keisuke Sakaguchi, originally published in Napoles et al. (2015), Napoles et al. (2016d), Sakaguchi et al. (2016), and Napoles et al. (2017c). My contributions to this work are creation of automatic metrics for GEC (Sections 7.4 and 7.7.3), evaluation and analysis of metrics and annotations, and creating annotations. Keisuke Sakaguchi's contributions were collecting annotations and human judgments, probabilistic inference of system rankings with TrueSkill, calculating agreement levels, and development of a hybrid metric.

Original Sentence During that period, if one of the family member reflects genetic disorder symptoms, he will fell in an ethical dilenma for sure.
Correction 1 During that period, if one of the family member reflects **the** genetic disorder symptoms, he will **fall** in an ethical dilenma for sure.
Correction 2 During that period, if one of the family **members** reflects *a* genetic disorder symptoms, he will **feel** in an ethical dilenma for sure.

Table 7.1: Two proposed corrections of a sentence rewritten by an English language learner from the CoNLL 2014 Shared Task evaluation set (Ng et al., 2014).

(2012), there is often no single correction for an error (e.g., whether to correct a subject-verb agreement error by changing the number of the subject or the verb), and errors cover a range of factors including style, register, venue, audience, and usage questions, about which there can be much disagreement. For example, the two candidate corrections in Table 7.1 contain edits addressing different errors, and attempt to correct the sentence in different ways. Neither correction fixes the misspelling of *dilemma*, and neither is representative of fluent standard English.

At this point, we should consider the goal of GEC. In NLP, GEC was originally tasked with correcting targeted error types in order to provide feedback to non-native writers (e.g., Chodorow and Leacock, 2000; Dale and Kilgarriff, 2011; Leacock et al., 2014). With publicly available annotated datasets and more advanced methods applied to the task, systems have improved and now aim to correct all errors of every error type (Ng et al., 2014).

Original
From this scope , social media has shorten our distance .
Technically Grammatical
From this scope , social media has shortened our distance .
Fluent
From this perspective , social media has shortened the distance between us .

Table 7.2: An *Original* sentence and candidate *Corrected* version from the NUCLE corpus, followed by a *Fluent* correction from our new annotations.

With *whole-sentence correction* the standard GEC task, we encourage the community to reevaluate the definition of GEC.

Learner text often exhibits problems that are not easily categorized into a single, span-delimited error. Consider the example in Table 7.2. The second version of the sentence was corrected by a human annotator using error coding and corrects the verb error. This sentence is "technically" grammatical, but still unacceptable to a native or fluent speaker. However, when we aim to correct the sentence as a whole, the most conspicuous error has to do with how unnaturally this sentence reads. The meanings of words and phrases like *scope* and the corrected *shortened our distance* are clear, but this is not how a native English speaker would use them. A more fluent version of this sentence is the third version in Table 7.2.

This issue motivates a broader definition of grammaticality that we will term *native-language fluency*, or simply *fluency*. One can argue that traditional understanding of grammar and grammar correction encompasses the idea of native-language fluency. However, the metrics commonly used in evaluating GEC undermine these arguments. The performance of GEC systems is typically evaluated using metrics that compute corrections against *error-coded* corpora, which impose a taxonomy of types of grammatical errors. Assigning these codes can be difficult, as evidenced by the low agreement found between annotators of grammatical errors (Rozovskaya and Roth, 2010; Bryant and Ng, 2015). It is also quite expensive. But most importantly, as we show in the following section, annotating for explicit error codes places a downward pressure on annotators to find and fix concrete, easily-identifiable grammatical errors (such as *wrong verb tense*) in lieu of addressing the native fluency of the text.

7.1 Correcting for Fluency

We hypothesize that human judges, when presented with two versions of a sentence, will favor *fluent* versions over ones that exhibit only *technical grammaticality*.

By *technical grammaticality*, we mean adherence to an accepted set of grammatical conventions. In contrast, we consider a text to be *fluent* when it looks and sounds natural to a native-speaking population. Both of these terms are hard to define precisely, and fluency especially is a nuanced concept for which there is no checklist of criteria to be met.[2] To demonstrate these intuitions, Table 7.3 contains examples of sentences that are one, both,

[2] It is important to note that both grammaticality and fluency are determined with respect to a particular speaker population and setting. In this thesis, we focus on Standard Written (American) English, which is the standard used in education, business, and journalism. While judgments of individual sentences would differ for other populations and settings (for example, spoken African-American Vernacular English), the distinction between grammaticality and fluency would remain.

	Technically Grammatical	Not Technically Grammatical
Fluent	In addition, it is impractical to make such a law.	I don't like this book, it's really boring.
Not Fluent	Firstly, someone having any kind of disease belongs to his or her privacy.	It is unfair to release a law only point to the genetic disorder.

Table 7.3: Examples of sentences that are different combinations of fluent and technically grammatical.

or neither. A text does not have to be technically grammatical to be considered fluent, although in almost all cases, fluent texts are also technically grammatical. This chapter will demonstrate how they are quantifiably different with respect to GEC.[3]

Annotated corpora for GEC are almost exclusively error coded. The Cambridge Learner Corpus, a proprietary collection of essays written by English language learners, is coded for 80 error types (Nicholls, 2003). A subset of that corpus, consisting of essays by students sitting for the First Certificate Exam (FCE), is freely available to the community (Yannakoudakis et al., 2011). The NUS Corpus of Learner English (NUCLE) consists of essays written by advanced English language learners at the National University of Singapore (Dahlmeier et al., 2013). NUCLE is annotated with a smaller set of errors than CLC (28), and was used in the 2013 and 2014 shared tasks. No substantial parallel corpora exist for GEC without error coding. The two that we are aware of are the Lang-8 corpus, containing automatically aligned sentences written and corrected by ELLs (Mizumoto et al., 2011), and the AESW corpus, which has proofreading corrections on scientific writ-

[3] The work in this section was originally published in Sakaguchi et al. (2016).

ing (Daudaravicius et al., 2016). The former is noisy and has been used for additional training data for MT-based approaches (like the one described in Chapter 8), and the latter contains very minimal errors, the majority of which address punctuation.

Annotating coded errors encourages a minimal set of edits because more substantial edits often address overlapping and interacting errors. For example, the annotators of the NUCLE corpus, which was used for the CoNLL shared tasks, were explicitly instructed to select the minimal text span of possible alternatives (Dahlmeier et al., 2013). There are situations where error-coded annotations are useful to help students correct specific grammatical errors. The ability to do this with the non-error-coded, fluent annotations we advocate here is no longer direct but it is not lost entirely. For this purpose, some studies have proposed *post hoc* automated error-type classification methods (Swanson and Yamangil, 2012; Xue and Hwa, 2014; Bryant et al., 2017), which compare the original sentence to its correction and deduce the error types. A similar method could be applied to non-coded fluency edits.

We speculate that by removing the error-coding restraint, we can obtain edits that sound more fluent to native speakers while also reducing the expense of annotation, with diminished time and training requirements. Chodorow et al. (2012) and Tetreault et al. (2014) suggested that it is better to have a large number of annotators to reduce bias in automatic evaluation. Following this recommendation, we collected additional annotations without error codes, written by both experts and non-experts.

	Edit type		
Annotator	*Minimal*	*Fluency*	
Expert	1,312 × 2	1,312 × 2	
Non-expert	1,312 × 2	1,312 × 2	
Total	5,248	5,248	**10,496**

Table 7.4: New corrections collected in this work.

7.2 A Corpus of Fluency Corrections

In collecting a new fluency corpus, we have two goals: to determine which type of corrections humans prefer whether reliable corrections can be collected from untrained annotators using crowdsourcing. We collected 8 additional corrections for each sentence in the CoNLL-2014 shared task test set, using both expert and crowdsourced annotators, and fluency and minimal edit corrections. The expert annotators were three native speaking authors of this work and the crowdsourced annotations were collected from Mechanical Turk workers in the United States.[4] Work in machine translation shows that more references are better for evaluation (Finch et al., 2004) and, concurrent to this work and also recognizing the importance of multiple references, Bryant and Ng (2015) collected 8 additional minimal-edit corrections for each of the CoNLL-14 sentences.

We compared the quality of the correction by different groups of annotators on Mechanical Turk, and found that when people (both experts and non-experts) are asked to make minimal edits, they make few changes to the sentences and also change fewer of the

[4] Annotation instructions are included in Appendices A.6.1 and A.6.2, and further detail about annotation collection can be found in Sakaguchi et al. (2016)

Original	Genetic disorder may or may not be hirataged hereditary disease and it is sometimes hard to find out one has these kinds of diseases .
Expert fluency	A genetic disorder may or may not be □ a hereditary disease , and it is sometimes hard to find out whether one has these kinds of diseases .
Non-expert fluency	Genetic □ factors can manifest overtly as disease □ , or simply be carried , making it □ hard , sometimes , to find out if one has □ a genetic predisposition to disease .

Table 7.5: An example sentence with expert and non-expert fluency edits. Moved and changed or inserted spans are underlined and □ indicates deletions.

sentences. Fluency edits show the opposite effect, with non-experts taking more liberties than experts with both the number of sentences changed and the degree of change within each sentence (see Table 7.5 for an extreme example of this phenomenon). In order to quantify the extent of changes made in the different annotations, we look at the percent of sentences that were left unchanged as well as the number of changes needed to transform the original sentence into the corrected annotation. To calculate the number of changes, we used a modified Translation Edit Rate (TER), which measures the number of edits needed to transform one sentence into another (Snover et al., 2006). An edit can be an insertion, deletion, substitution, or shift. We chose this metric because it counts the movement of a phrase (a *shift*) as one change, which the Levenshtein distance would heavily penalize. TER is calculated as the number of changes per token, but instead we report the number of changes per *sentence* for ease of interpretation, which we call *sTER*.[5]

[5] TER is better than edit distance for this analysis because it considers moving a phrase a single edit.

We compare the original set of sentences to the new annotations and the existing NUCLE and BN15 reference sets to determine the relative extent of changes made by the fluency and minimal edits (Figure 7.1). Compared to the original, non-experts had a higher average sTER than experts, meaning that they made more changes per sentence. For fluency edits, experts and non-experts changed approximately the same number of sentences, but the non-experts made about seven edits per sentence compared to the experts' four. Minimal edits by both experts and non-experts exhibit a similar degree of change from the original sentences, so further qualitative assessment is necessary to understand whether the annotators differ. Table 7.6 contains an example of how the same ungrammatical sentence was corrected using both minimal and fluency edits, as well as one of the original NUCLE corrections.

The error-coded annotations of NUCLE and BN15 fall somewhere in between the fluency and minimal edits in terms of sTER. The most conservative set of sentences is the system output of the CoNLL 2014 shared task, with sTER = 1.4, or approximately one change made per sentence. In contrast, the most conservative human annotations, the minimal edits, edited a similar percent of the sentences but made about two changes per sentence.

When there are multiple annotators working on the same data, one natural question is the inter-annotator agreement (IAA). For GEC, IAA is often low and arguably not an appropriate measure of agreement (Bryant and Ng, 2015). Additionally, it would be difficult, if possible, to reliably calculate IAA without coded alignments between the new and original sentences. Therefore, we look at two alternate measures: the percent of sentences to which

Original

Some family may feel hurt , with regards to their family pride or reputation , on having the knowledge of such genetic disorder running in their family .

NUCLE

Some family members may feel hurt □ with regards to their family pride or reputation □ on having the knowledge of a genetic disorder running in their family .

Expert fluency

On □ learning of such a genetic disorder running in their family , some family members may feel hurt □ regarding their family pride or reputation .

Non-expert fluency

Some relatives may □ be concerned about the family 's □ reputation – not to mention their own pride – in relation to this news of □ familial genetic defectiveness □ .

Expert minimal

Some families may feel hurt □ with regards to their family pride or reputation , on having □ knowledge of such a genetic disorder running in their family .

Non-expert minimal

Some family may feel hurt □ with regards to their family pride or reputation □ on having the knowledge of such genetic disorder running in their family .

Table 7.6: An example sentence with the original NUCLE correction and fluency and minimal edits written by experts and non-experts. Moved and changed or inserted spans are underlined and □ indicates deletions.

different annotators made the same correction(s) and the sTER between two annotators' corrections, reported in Table 7.7.

As we expect, there is notably lower agreement between the annotators for fluency edits than for minimal edits, due to the presumably smaller set of required versus optional stylistic changes. Expert annotators produced the same correction on 15% of the fluency edits,

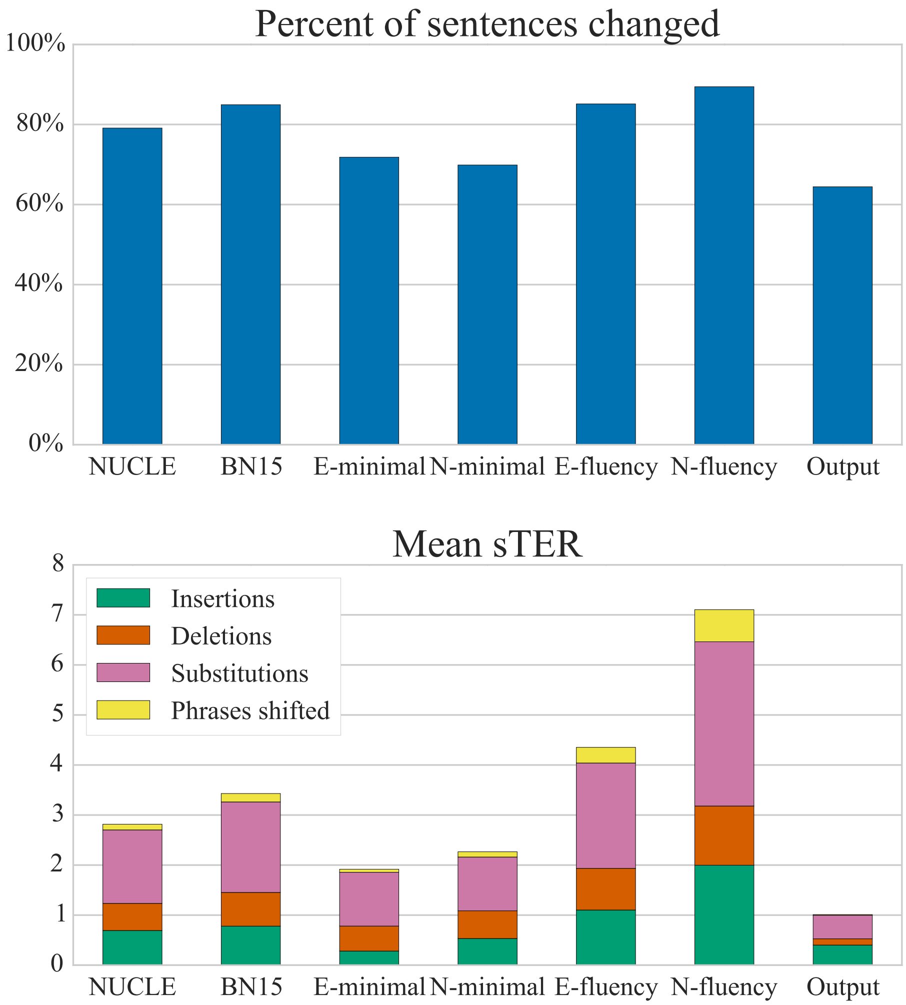


Figure 7.1: Amount of changes made by different annotation sets compared to the original sentences.

but more than 38% of their minimal edits were identical. Half of these identical sentences were unchanged from the original. There was lower agreement between non-expert annotators than experts on both types of edits. We performed the same calculations between the two NUCLE annotators and found that they had agreement rates similar to the non-expert minimal edits. However, the experts' minimal edits have much higher consensus than both the non-experts' and NUCLE, with twice as many identical corrected sentences and half

Edit type	**Annotators**	**Identical**	**sTER**
Fluency	E_1 v. E_2	15.3%	5.1
	N_1 v. N_2	5.9%	10.0
	E v. N	8.5%	7.9
Minimal	E_1 v. E_2	38.7%	1.7
	N_1 v. N_2	21.8%	2.9
	E v. N	25.9%	2.4
NUCLE	A v. B	18.8%	3.3

Table 7.7: A comparison of annotations across different annotators (E for expert, N for non-expert). Where there were more than two annotators, statistics are over the full pairwise set. *Identical* refers to the percentage of sentences where both annotators made the same correction and *sTER* is the mean sTER between the annotators' corrections.

the sTER.

From this analysis, one could infer that the expert annotations are more reliable than the non-expert because there are fewer differences between annotators and fewer changes per sentence. Finally, we performed a manual evaluation of the corrections from different annotation groups. For 300 randomly selected sentences, two judges on Mechanical Turk ranked the new annotations, one of the NUCLE references, and the original sentence (details on the interface are found in Appendix A.7.1). We inferred the relative rank of the annotation groups using a method called TrueSkill, which will be discussed in more detail in Section 7.5. The inferred ranking of annotation groups is found in Table 7.8. There is no significant difference in the perceived quality of the fluency annotations from either group, but the minimal-edit corrections by experts are significantly better than those by the non-experts.

In Section 7.4, we will re-evaluate the results of the CoNLL-2014 shared task on the

#	Score	Range	Annotation type
1	1.164	1–2	Expert fluency
	0.976	1–2	Non-expert fluency
3	0.540	3	NUCLE
4	0.265	4	Expert minimal
5	-0.020	5	Non-expert minimal
6	-2.925	6	Original sentence

Table 7.8: Human ranking of the new annotations by grammaticality. Lines between systems indicate clusters according to bootstrap resampling at $p \leq 0.05$. Each system is represented as a Gaussian and *Score* is the mean of the distribution. Systems in the same cluster are considered to be tied.

newly collected fluency corpora.

7.3 JFLEG: A New Fluency Evaluation Set

The NUCLE corpus represents the types of grammatical errors made by English language learners with similar proficiency levels and (predominantly) the same native language. To accurately assess the current state of GEC, an evaluation corpus should contain text written by speakers of different native languages with different proficiencies. We present a new parallel corpus to fill this need, **J**HU **FL**uency-**E**xtended **G**UG corpus (JFLEG), for more completely evaluating and developing GEC systems. JFLEG adds four fluency annotations to 1,511 sentences from the Grammatical-UnGrammatical corpus (GUG) (Heilman et al., 2014). GUG contains sentences from English-language proficiency exams and annotations for how grammatical each sentence is. The GUG score ranges from 0–4, with 0 being unintelligible or an unfinished sentence, 1 being very ungrammatical, and 4 perfectly

Please correct the following sentence to make it sound natural and fluent to a native speaker of (American) English. The sentence is written by a second language learner of English. You should fix grammatical mistakes, awkward phrases, spelling errors, etc., following standard written usage conventions, but your edits must be conservative. Please keep the original sentence (words, phrases, and structure) as much as possible. The ultimate goal of this task is to make the given sentence sound natural to native speakers of English without making unnecessary changes. Please do not split the original sentence into two or more. Edits are not required when the sentence is already grammatical and sounds natural.

Table 7.9: JFLEG annotation instructions.

grammatical. With added fluency annotations, JFLEG represents a broad range of language proficiency levels and uses holistic *fluency edits* to not only correct grammatical errors but also make the original text more native sounding. We describe the types of corrections made and benchmark four leading GEC systems on this corpus, identifying specific areas in which they do well and how they can improve.[6]

In Section 7.2, we found that there was no significant difference in the perceived quality of fluency corrections made by experts and non-experts (Figure 7.8), and so we used crowdsourcing to collect four corrections of each of 1,511 sentences in the GUG development and test sets. We additionally used quality control measures, specifically requiring workers to take a qualifying task and manually reviewing their corrections. The instructions are found in Table 7.9, and we refer the reader to Napoles et al. (2017c) for additional information about the annotation process.

[6] This work was originally published in Napoles et al. (2017c).

		Error type in original		
		Awkward	**Orthographic**	**Grammatical**
Edit type	Fluency	38%	35%	32%
	Minimal	82%	89%	85%

Table 7.10: Percent of sentences by error type that were changed with fluency or minimal edits. Each row indicates that annotators made a fluency edit or minimal edit, and each column indicates the type of error the edit was addressing.

7.3.1 Types of Annotations in JFLEG

As previously discussed, it is not possible to directly measure inter-annotator agreement for fluency corrections, and so we perform a series of statistical and manual analyses of the JFLEG corrections. The mean LD between the original and corrected sentences is more than twice that of existing corpora (Table 2.3). LD negatively correlates with the GUG grammaticality score ($r = -0.41$) and the annotation difficulty score (-0.37), supporting the intuition that less grammatical sentences require more extensive changes, and it is harder to make corrections involving more substantive edits. Because there is no clear way to quantify agreement between annotators, we compare the annotations of each sentence to each other. The mean LD between all pairs of annotations is greater than the mean LD between the original and corrected sentences (15 characters), however 36% of the sentences were corrected identically by at least two participants.

Next, we examined 100 randomly selected original and human-corrected sentence pairs and labeled them with the type of error present in the sentence and the type of edit(s) applied in the correction. The three error types are sounds *awkward* or has an *orthographic* or *grammatical* error.[7] The majority of the original sentences have at least one error (81%), and 68% of these sentences are completely corrected in the annotations/ Few annotated sentences have orthographic (4%) or grammatical (10%) errors, but awkward errors are more frequent (23% of annotations were labeled *awkward*)—which is not very surprising given how garbled some original sentences are and the dialectal variation of what sounds awkward.

The corrected sentences were also labeled with the type of changes made (minimal and/or fluency edits). Minimal edits reflect a minor change to a small span (1–2 tokens) addressing an immediate grammatical error, such as number agreement, tense, or spelling. Fluency edits are more holistic and include reordering or rewriting a clause, and other changes that involve more than two contiguous tokens. 69% of annotations contain at least one minimal edit, 25% a fluency edit, and 17% both fluency and minimal edits. The distribution of edit types is fairly uniform across the error type present in the original sentence (Table 7.10). Notably, fewer than half of awkward sentences were corrected with fluency edits, which may explain why so many of the corrections were still *awkward*.

In the following section, we will examine metrics for evaluating GEC and propose a new fluency metric. We will demonstrate that the fluency metric is more reliable than previous metrics, and we will evaluate leading GEC systems on the fluency annotations we have collected for the NUCLE and GUG corpora.

[7] Due to their frequency, we separate orthographic errors (spelling and capitalization) from other grammatical errors.

7.4 Fluency Evaluation of GEC

As of 2018, M^2 is the standard automatic metric for GEC, having been used to rank error correction systems in the 2013 and 2014 CoNLL shared tasks (Ng et al., 2013; Ng et al., 2014). M^2 was evaluated by comparing its output against that of the official Helping Our Own (HOO) scorer (Dale and Kilgarriff, 2011), itself based on the `wdiff` tool. In other words, it was evaluated under the assumption that evaluating GEC can be reduced to checking whether a set of predefined errors have been changed into a set of associated corrections.

Beyond having never been validated against human scores, M^2 is not without its own issues. First, phrase-level edits can be gamed because the lattice treats a long phrase deletion as one edit.[8] capture the difference between "no change" and "wrong edits" made by systems. Chodorow et al. (2012) also list other complications arising from using F-score or M^2, depending on the application of GEC. Citing issues with M^2, Felice and Briscoe (2015) proposed I-measure. Comparing the metric scores on the CoNLL 2014 test set has striking results: there is a negative correlation between the M^2 and I-measure scores (Pearson's $r = -0.694$).

These metrics were developed without consulting human judgments. Additionally, they require detailed annotations of the offsets and error type of each correction in response to an explicit error annotation scheme. Due to the inherent subjectivity and poor definition of the

[8] For example, when we put a single character 'X' as system output for each sentence, we obtain $P = 0.27, R = 0.29, M^2 = 0.28$, which would be ranked 6/13 systems in the 2014 CoNLL shared task.

task (Section 7.1), it is difficult for annotators to reliably produce these annotations (Bryant and Ng, 2015). However, this requirement can be relinquished by treating GEC as a text-to-text rewriting task and borrowing metrics from machine translation, as Park and Levy (2011) did with BLEU (Papineni et al., 2002) and METEOR (Lavie and Agarwal, 2007).[9] Applied off-the-shelf, these metrics yield unintuitive results. For example, BLEU ranks the *source* sentence as second place in the CoNLL-2014 shared task.[10]

The problem is partially due to the subtle but important difference between machine translation and monolingual text-rewriting tasks. In MT, an untranslated word or phrase is almost always an error, but in grammatical error correction and other T2T tasks, this is not the case. Some, but not all, regions of the source sentence should be changed. This observation motivates a small change to BLEU that computes n-gram precisions over the reference but assigns more weight to n-grams that have been correctly changed from the source. This revised metric, Generalized Language Evaluation Understanding (GLEU), rewards corrections while also correctly crediting unchanged source text.

BLEU(C,R) is computed as the geometric mean of the modified precision scores of the test sentences C relative to the references R, multiplied by a brevity penalty to control for recall (Papineni et al., 2002). The precisions are computed over bags of n-grams derived from the candidate translation and the references. Each n-gram in the candidate sentence is "clipped" to the maximum count of that n-gram in any of the references, ensuring that no

[9] The work described in this section originally appeared in Napoles et al. (2015).

[10] Of course, it could be the case that the source sentence is actually the second best, but our human evaluation (Section 7.5) confirms that this is not the case.

$$p'_n = \frac{\sum\limits_{n-gram \in C} Count_{R \setminus S}(n-gram) - \lambda \left(Count_{S \setminus R}(n-gram)\right) + Count_R(n-gram)}{\sum\limits_{n-gram' \in C'} Count_S(n-gram') + \sum\limits_{n-gram \in R \setminus S} Count_{R \setminus S}(n-gram)} \tag{7.1}$$

precision is greater than 1.

Similar to I-measure, which calculates a weighted accuracy of edits, we calculate a weighted precision of n-grams. In our adaptation, we modify the precision calculation to assign extra weight to n-grams present in the candidate that overlap with the reference *but not* the source (the set of n-grams $R \setminus S$). The precision is also penalized by a weighted count of n-grams in the candidate that are in the source but not the reference (false negatives, $S \setminus R$). Unlike *i*BLEU, which penalizes overlap between the candidate and the source agnostic to the overlap between the source and the reference (Equation 6.4), GLEU penalizes false negatives explicitly.

Originally, GLEU was formulated with a weight λ which determined how much incorrectly changed n-grams should be penalized (GLEU^0; Equation 7.5). For a correction candidate C with a corresponding source S and reference R, the modified n-gram precision for GLEU^0(C,R,S) is shown in Equation 7.1. The weight λ determines by how much incorrectly changed n-grams are penalized. Equations 7.2–7.3 describe how the counts are collected given a bag of n-grams B.

$$Count_B(n-gram) = \sum_{n-gram' \in B} d(n-gram, n-gram') \tag{7.2}$$

$$d(n-gram, n-gram') = \begin{cases} 1 & \text{if } n-gram = n-gram' \\ \\ 0 & \text{otherwise} \end{cases} \tag{7.3}$$

$$\mathrm{BP} = \begin{cases} 1 & \text{if } c > r \\ \\ e^{(1-c/r)} & \text{if } c \leq r \end{cases} \tag{7.4}$$

$$\mathrm{GLEU}^0\,(C,R,S) = \mathrm{BP} \cdot \exp\left(\sum_{n=1}^{N} w_n \log p'_n\right) \tag{7.5}$$

GLEU^0 double-counts n-grams in the reference that do not appear in the source, and it subtracts a weighted count of n-grams that appear in the source (S) and not the reference (R). Following the publication of Napoles et al. (2015), we observed that the weight needed to be re-tuned as the number of references changed. With more references, more variations of the sentence are seen which results in a larger set of reference n-grams. Larger sets of reference n-grams tend to have higher overlap with the source n-grams, which decreases the number of n-grams that were seen in the source but not the reference. Because of this, the penalty term decreases and a large weight is needed for the penalty term to have the same magnitude as the penalty when there are fewer references. Because the number of possible reference n-grams increases as more reference sets are used, we calculate an intermediate GLEU by randomly sample from one of the references for each sentence,

$$p_n^* = \frac{\left(\sum\limits_{ngram \in \{C \cap R\}} count_{C,R}(ngram) - \sum\limits_{ngram \in \{C \cap S\}} \max\left[0, count_{C,S}(ngram) - count_{C,R}(ngram)\right] \right)}{\sum\limits_{ngram \in \{C\}} count(ngram)}$$

$$count_{A,B}(ngram) = \min(\text{\# occurrences of } ngram \text{ in A}, \text{\# occurrences of } ngram \text{ in B})$$

Equation 1: Modified precision calculation of GLEU.

and report the mean score over 500 iterations. It takes less than 30 seconds to evaluate 1,000 sentences using 500 iterations. Precision is simply the number of reference n-gram matches, minus the counts of n-grams found more often in the source than the reference (Equation 7.6) (Napoles et al., 2016a). GLEU follows the same intuition as the original GLEU^0: overlap between S and R should be rewarded and n-grams that should have been changed in S but were not should be penalized. We report the results of both versions of the metric in this section for completeness, however we discourage use of GLEU^0 and will not use it in subsequent parts of this work.

7.5 Ground Truth Evaluation with TrueSkill

When scoring the twelve publicly released system outputs from the CoNLL-2014 Shared Task[11] we find a negative correlation between two automatic metrics, Max-Match (M^2) (Dahlmeier and Ng, 2012) and BLEU scores ($r = -0.772$) and positive correlation between I-measure and BLEU scores ($r = 0.949$). With the earlier-reported negative correlation between I-measure and M^2, we have a troubling picture: which of these metrics is

[11] www.comp.nus.edu.sg/~nlp/conll14st.html

best? Which one actually captures and rewards the behaviors we would like our systems to report? Despite these many proposed metrics, no prior work has attempted to answer these questions by comparing them to human judgments. We propose to answer these questions by producing a definitive human ranking, against which the rankings of different metrics can be compared. Four judges familiar with the task ranked system outputs of each input sentence. The instructions given to participants are included in Appendix A.7.1. An absolute system ranking was found using the TrueSkill approach (Herbrich et al., 2006; Sakaguchi et al., 2014), following the setup of the Workshop on Machine Translation (Bojar et al., 2014; Bojar et al., 2015). For each competing system, TrueSkill infers the absolute system quality from the pairwise comparisons, representing each as the mean of a Gaussian. These means can then be sorted to rank systems. By running TrueSkill 1,000 times using bootstrap resampling and producing a system ranking each time, we collect a range of ranks for each system. We can then cluster systems according to non-overlapping rank ranges (Koehn, 2012) to produce the final ranking, allowing ties. Details about collecting human judgments and inferring an absolute system ranking can be found in Napoles et al. (2015).[12]

[12] Human judgments collected and system ranking calculated by Keisuke Sakaguchi.

Metric	r	ρ
GLEU^0	0.542	**0.555**
GLEU	**0.549**	0.401
M^2	0.358	0.429
I-measure	−0.051	−0.005
BLEU	−0.125	−0.225

Table 7.11: Correlation of metrics with the human ranking (excluding the reference), as calculated with Pearson's r and Spearman's ρ.

Rank	**Human**	**BLEU**	**I-measure**	**M^2**	**GLEU^0**	**GLEU**
1	CAMB	UFC	UFC	CUUI	CUUI	CAMB
2	AMU	source	source	CAMB	AMU	CUUI
3	RAC	IITB	IITB	AMU	UFC	AMU
4	CUUI	SJTU	SJTU	POST	CAMB	UMC
5	source	UMC	CUUI	UMC	source	PKU
6	POST	CUUI	PKU	NTHU	IITB	POST
7	UFC	PKU	AMU	PKU	SJTU	SJTU
8	SJTU	AMU	UMC	RAC	PKU	NTHU
9	IITB	IPN	IPN	SJTU	UMC	UFC
10	PKU	NTHU	POST	UFC	NTHU	UUTB
11	UMC	CAMB	RAC	IPN	POST	source
12	NTHU	RAC	CAMB	IITB	RAC	RAC
13	IPN	POST	NTHU	source	IPN	IPN

Table 7.12: System outputs scored by different metrics, ranked best (1) to worst (13).

7.6 Meta-Evaluation of Metrics

We compute the goodness of each metric by calculating the correlation of the system ranking by metric score with the human ranking (Table 7.11).[13] The ranked systems of each metric are found in Table 7.12.

The human ranking is considerably different from those of most of the metrics, a fact

[13] The parameters of GLEU^0 are $N = 4$, $w_n = \frac{1}{N}$, and $\lambda = 0$.

System	Sentence	Scores
Original	We may in actual fact communicating with a hoax Facebook acccount of a cyber friend , which we assume to be real but in reality , it is a fake account .	–
Ref. 1	We may in actual fact **be** communicating with a hoax Facebook acccount of a cyber friend , which we assume to be real but in reality , it is a fake account .	–
Ref. 2	We may in actual fact **be** communicating with a **fake** Facebook **account** of **an online** friend , which we assume to be real but , in reality , it is a fake account .	–
UMC	We may **be** in actual fact communicating with a hoax Facebook acccount of a cyber friend , we assume to be real but in reality , it is a fake account .	GLEU = 0.62 $M^2 = 0.00$ TS rank = 1
AMU	We may in actual fact communicating with a hoax Facebook **account** of a cyber friend , which we assume to be real but in reality , it is a fake **accounts** .	GLEU = 0.64 $M^2 = 0.39$ TS rank = 2
NTHU	We may of actual fact communicating with a hoax Facebook acccount of a cyber friend , which we **assumed** to be real but in reality , it is a fake account .	GLEU = 0.60 $M^2 = 0.00$ TS rank = 4

Table 7.13: Examples of system output (changes are in bold), the sentence-level scores assigned by different metrics, and the TrueSkill rank.

that is also captured in correlation coefficients (Table 7.11).[14] From the human evaluation, we learn that the source falls near the middle of the rankings, even though the BLEU, I-measure and M^2 rank it among the best or worst systems.

M^2, the metric that has been used for the CoNLL shared tasks, only correlates moderately with human rankings, suggesting that it is not an ideal metric for judging the results

[14] Pearson's measure assumes the scores are normally distributed, which may not be true here.

of a competition. Even though I-measure perceptively aims to predict whether an output is better or worse than the input, it actually has a slight negative correlation with human rankings. GLEU$_0$ is the only metric that strongly correlates with the human ranks, and performs closest to the range of human-to- human correlation ($0.73 \leq r \leq 0.81$) GLEU$_0$ correctly ranks four out of five of the top human-ranked systems at the top of its list, while the other metrics rank at most three of these systems in the top five.

All metrics deviate from the human rankings, which may in part be because automatic metrics equally weight all error types, when some errors may be more tolerable to human judges than others. For example, inserting a missing token is rewarded the same by automatic metrics, whether it is a comma or a verb, while a human would much more strongly prefer the insertion of the latter. An example of system outputs with their automatic scores and human rankings is included in Table 7.13.

This example illustrates some challenges faced when using automatic metrics to evaluate GEC. The automatic metrics weight all corrections equally and are limited to the gold-standard references provided. Both automatic metrics, M^2 and GLEU, prefer the AMU output in this example, even though it corrects one error and *introduces* another. The human judges rank the UMC output as the best for correcting the main verb even though it ignored the spelling error. The UMC and NTHU sentences both receive M$^2 = 0$ because they make none of the gold-standard edits, even though UMC correctly inserts *be* into the sentence. M^2 does not recognize this since it is in a different location from where the annotators placed it. However, GLEU awards UMC partial credit for adding the correct unigram, and further assigns all sentences a real score.

We have described a new metric for GEC and shown that it correlates more strongly with human judgments than M^2 on the CoNLL-14 shared task results, using 2 minimal edit references. Next we show that GLEU used with fluency judgments is the most reliable approach to evaluating GEC output.

7.7 Fluency References and GLEU Are the Best Evaluation

We have demonstrated that humans prefer fluency edits to error-coded and minimal-edit corrections, but it is unclear whether these annotations are an effective reference for automatic evaluation. The broad range of changes that can be made with non-minimal edits may make it especially challenging for current automatic evaluation metrics to use. In this section, we investigate the impact that different reference sets have on the system ranking found by different evaluation metrics. With reference sets having such different characteristics, the natural question is: which reference and evaluation metric pairing best reflects human judgments of grammaticality?

To answer this question, we performed a comprehensive investigation of existing metrics and annotation sets to evaluate the 12 system outputs made public from the 2014 CoNLL Shared Task. To our knowledge, this is the first time that the interplay of annotation scheme and evaluation metric, as well as the rater expertise, has been evaluated

$$p_n^* = \frac{\left(\sum\limits_{ngram \in \{C \cap R\}} count_{C,R}(ngram) - \sum\limits_{ngram \in \{C \cap S\}} \max\left[0, count_{C,S}(ngram) - count_{C,R}(ngram)\right]\right)}{\sum\limits_{ngram \in \{C\}} count(ngram)}$$
$$count_{A,B}(ngram) = \min\left(\text{\# occurrences of } ngram \text{ in } A, \text{\# occurrences of } ngram \text{ in } B\right) \tag{7.6}$$

jointly for GEC.

7.7.1 Experimental Setup

The four automatic metrics that we investigate are M^2, I-measure,[15] GLEU, and BLEU. We include the machine-translation metric BLEU because evaluating against our new non-coded annotations is similar to machine-translation evaluation, which considers overlap instead of absolute alignment between the output and reference sentences.

For the M^2 and I-measure evaluations, we aligned the fluency and minimal edits to the original sentences using a Levenshtein edit distance algorithm.[16] Neither metric makes use of the annotation labels, so we simply assigned dummy error codes.

We compare the system outputs to each of the six annotation sets and a seventh set containing all of the annotations, using each metric. We ranked the systems based on their scores using each metric–annotation-set pair, and thus generated a total of 28 different rankings (4 metrics × 7 annotation sets).

[15] We ran I-measure with the `-nomix` flag, preventing the algorithm from finding the optimal alignment across all possible edits. Alignment was very memory-intensive and time consuming, even when skipping long sentences.

[16] Costs for insertion, deletion, and substitution are set to 1, allowing partial match (e.g., same lemma).

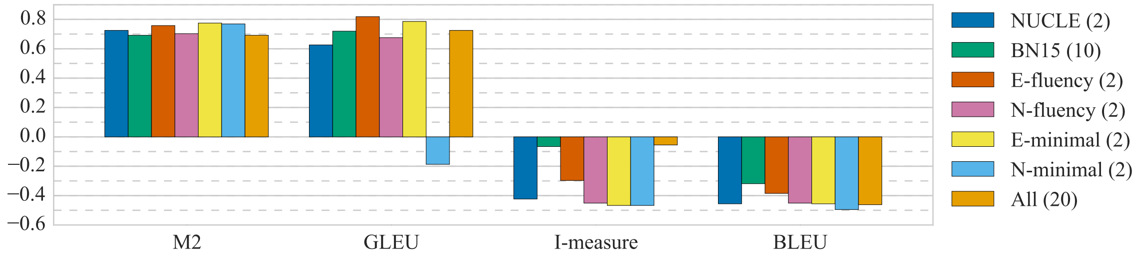


Figure 7.2: Correlation of the human ranking with metric scores over different reference sets (Spearman's ρ). The number of annotations per sentence in each set is in parentheses. See Table 7.14 for the numeric values.

Metric	M^2	GLEU	I-measure	BLEU
NUCLE	0.725	0.626	−0.423	−0.456
	0.677∗	0.646	−0.313	−0.310
BN15	0.692	0.720	−0.066	−0.319∗
	0.641	0.697	−0.007	−0.255
E-fluency	0.758	**0.819**∗	−0.297	−0.385
	0.665	**0.731**∗	−0.256	−0.230∗
N-fluency	0.703	0.676	−0.451	−0.451
	0.655	0.668	−0.319	−0.388
E-min.	0.775∗	0.786	−0.467	−0.456
	0.655	0.676	−0.385	−0.396
N-min.	0.769	−0.187	−0.467	−0.495
	0.641	−0.110	−0.402	−0.473
All	0.692	0.725	−0.055∗	−0.462
	0.617	0.724	0.061∗	−0.314

Table 7.14: Correlation between the human ranking and metric scores over different reference sets. The first line of each cell is Spearman's ρ and the second line is Pearson's r. The strongest correlations for each metric are starred, and the overall strongest correlations are in bold.

To determine the best metric, we compared the system-level ranking obtained from each evaluation technique against the expert human ranking reported in Grundkiewicz et al. (2015, Table 3c).

7.7.2 Results

Figure 7.2 and Table 7.14 show the correlation of the expert rankings with all of the evaluation configurations. For the leading metrics, M^2 and GLEU, the expert annotations had stronger positive correlations than the non-expert annotations. Using just two expert fluency annotations with GLEU has the strongest correlation with the human ranking out of all other metric–reference pairings ($\rho = 0.819$, $r = 0.731$), and it is additionally cheaper and faster to collect. E-fluency is the third-best reference set with M^2, which does better with minimal changes: the reference sets with the strongest correlations for M^2 are E-minimal ($\rho = 0.775$) and NUCLE ($r = 0.677$). Even though the non-expert fluency edits had more changes than the expert fluency edits, they still did reasonably well using both M^2 and GLEU.

The GLEU metric has strongest correlation when comparing against the E-fluency, BN15, E-minimal, and "All" reference sets. One could argue that, except for E-minimal, these references all have greater diversity of edits than NUCLE and minimal edits. Although BN15 has fewer changes made per sentence than the fluency edits, because of the number of annotators, the total pool of n-grams seen per sentence increases. We experimented with increasing the number of references used for calculating GLEU in Sakaguchi et al. (2016) and found that the GLEU score increases as the number of references increases from 1 to 10, with the increase leveling off once there are 4 references. E-minimal edits also have strong correlation, suggesting there may be a trade-off between quantity and quality of references.

The reference sets against which M^2 has the strongest correlation are NUCLE, expert fluency, and expert minimal edits. Even non-expert fluency annotations result in a stronger correlation with the human metric than BN15. These findings support the use of fluency edits even with a metric designed for error-coded corpora.

One notable difference between M^2 and GLEU is their relative performance using non-expert minimal edits as a metric. M^2 is robust to the non-expert minimal edits and, as a reference set, this achieves the second strongest Spearman's correlation for this metric. However, pairing the non-expert minimal edits with GLEU results in slightly *negative* correlation. This is an unexpected result, as there is sizable overlap between the non-expert and expert minimal edits (Table 7.7). We speculate that this difference may be due to the quality of the non-expert minimal edits. Recall that humans perceived these sentences to be worse than the other annotations, and better only than the original sentence (Table 7.8).

I-measure and BLEU are shown to be unfavorable for this task, having negative correlation with the human ranking, which supports the findings of Napoles et al. (2015) and Grundkiewicz et al. (2015). Even though BLEU and GLEU are both based on the n-gram overlap between the hypothesis and original sentences, GLEU has strong positive correlations with human rankings while BLEU has a moderate negative correlation. The advantage of GLEU is that it penalizes n-grams in the system output that were present in the input sentence and absent from the reference. In other words, a system loses credit for missing n-grams that should have been changed. BLEU has no such penalty and instead only

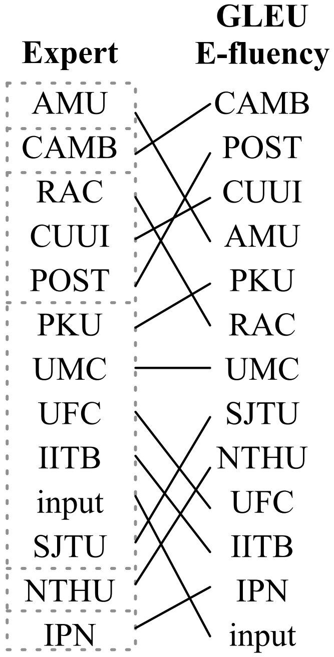


Figure 7.3: System rankings produced by GLEU with expert fluency (E-fluency) as the reference compared to the expert human ranking.

rewards n-grams that occur in the references and the output, which is a problem in same-language text rewriting tasks where there is significant overlap between the reference and the original sentences, as we found for the task of sentence simplification in Section 6.1.1. For this GEC data, BLEU assigns a higher score to the original sentences than to any of the systems.[17]

7.7.3 Handling Unseen Corrections

The metrics we have discussed depend on gold-standard corrections and therefore have a notable weakness: systems are penalized for making corrections that do not appear in

[17] Of course, it could be that the input sentences are the best, but the human ranking in Figure 7.3 suggests otherwise.

Original:
However , people now can contact with anyone all over the world who can use computer at any time , and there is no time delay of messages .

Candidate:
However , people now can ~~contact~~ **communicate** with ~~anyone~~ **people** all over the world who can use computer**s** at any time , and there is no time delay of messages .

Table 7.15: Sample corrected output with low reference-based metric scores (GLEU = 0.43, IM = 18.83, M^2 = 0.0).

the references.[18] For example, the output shown in Table 7.15 has low metric scores even though three appropriate corrections were made to the input.[19] These changes (in red) were not seen in the references and therefore the metrics GLEU and M^2 (described in Section 2.4) score this output worse than 75% of 15,000 other generated sentences.

While grammaticality-based, reference-less metrics have been effective in estimating the quality of machine translation (MT) output, the utility of such metrics has not been investigated previously for GEC. We hypothesize that such methods can overcome this weakness in reference-based GEC metrics.

GLEU, I-measure, and M^2 are calculated based on comparison to reference corrections. These Reference-Based Metrics (*RBMs*) credit corrections seen in the references and penalize systems for ignoring errors and making bad changes (changing a span of text in an ungrammatical way or introducing errors to grammatical text). However, RBMs make two strong assumptions: that the annotations in the references are *correct* and that they are *complete*. An exhaustive list of all possible corrections would be time-consuming, if

[18] We refer to the gold-standard corrections as *references* because *gold standard* suggests just one accurate correction.

[19] This work was originally published in Napoles et al. (2016d).

not impossible. As a result, RBMs penalize output that has a valid correction that is not present in the references or that addresses an error not corrected in the references. The example in Table 7.15 has low GLEU and M^2 scores, even though the output addresses two errors (GLEU=0.43 and M^2 = 0.00, in the bottom half and quartile of 15k system outputs, respectively). To address these concerns, we propose three metrics to evaluate the grammaticality of output without comparing to the input or a gold-standard sentence (Grammaticality-Based Metrics, or *GBMs*). We expect GBMs to score sentences, such as the aforementioned example, more highly. The first two metrics are scored by counting the errors found by existing grammatical error detection tools. The error count score is simply calculated: $1 - \frac{\text{\# errors}}{\text{\# tokens}}$. Two different tools are used to count errors: e-rater®'s grammatical error detection modules (ER) and Language Tool (Miłkowski, 2010) (LT). We choose these because, while e-rater® is a large-scale, robust tool that detects more errors than Language Tool,[20] it is proprietary whereas Language Tool is publicly available and open-sourced.

For our third method, we estimate a grammaticality score with a linguistic feature-based model (LFM), which is our implementation of Heilman et al. (2014).[21] The LFM score is a ridge regression over a variety of linguistic features related to grammaticality, including the number of misspellings, language model scores, OOV counts, and PCFG and link grammar features. It has been shown to effectively assess the grammaticality of learner writing. LFM predicts a score for each sentence while ER and LT, like the RBMs, can be calculated with either sentence- or document-level statistics. To be consistent with LFM,

[20] In the data used for this work, e-rater® detects approximately 15 times more errors than Language Tool.

[21] Our implementation is slightly modified in that it does not use features from the PET HPSG parser.

Metric	**Spearman's ρ**	**Pearson's r**
GLEU	**0.852**	**0.838**
ER	**0.852**	0.829
LT	0.808	0.811
I-measure	0.769	0.753
LFM	0.780	0.742
M^2	0.648	0.641

Table 7.16: Correlation between the human and metric rankings.

for all metrics in this work we score each sentence individually and report the system score as the mean of the sentence scores. Consistent with our hypothesis, ER and LT score the Table 7.15 example in the top quartile of outputs and LFM ranks it in the top half.

To assess the proposed metrics, we apply the RBMs, GBMs, and interpolated metrics to score the output of 12 systems participating in the CoNLL-2014 Shared Task on GEC (Ng et al., 2014). The interpolated metrics, developed by Keisuke Sakaguchi, combine GBM and RBMs to capture meaning retention (described in Napoles et al. (2016d)). We use the same 20 references for scoring with RBMs as in Section 7.7.1. Unlike previous GEC evaluations, all metrics reported here use the *mean* of the sentence-level scores for each system because interpolated scores are calculate on the sentence level.

Results are presented in Table 7.16. The error-count metrics, ER and LT, have stronger correlation than all RBMs except for GLEU, which is the current state of the art. GLEU has the strongest correlation with the human ranking ($\rho = 0.852$, $r = 0.838$), followed closely by ER, which has slightly lower Pearson correlation ($r = 0.829$) but equally as strong rank correlation, which is arguably more important when comparing different sys-

		ER	**LFM**	**LT**
	not interpolated	0.852 (0)	0.780 (0)	0.808 (0)
GLEU	0.852 (1)	**0.885** (0.03)	0.874 (0.27)	0.857 (0.04)
I-measure	0.769 (1)	0.874 (0.19)	0.863 (0.37)	0.852 (0.01)
M^2	0.648 (1)	0.868 (0.01)	0.852 (0.05)	0.808 (0)

(a) Spearman's ρ values

		ER	**LFM**	**LT**
	not interpolated	0.829 (0)	0.742 (0)	0.811 (0)
GLEU	0.838 (1)	**0.867** (0.27)	0.845 (0.84)	0.867 (0.09)
I-measure	0.753 (1)	0.837 (0.02)	0.791 (0.22)	0.828 (0.01)
M^2	0.641 (1)	0.829 (0.00)	0.754 (0.04)	0.811 (0)

(b) Pearson's r values

Table 7.17: Oracle correlations between interpolated metrics and the human rankings. The λ value for each metric is in parentheses.

tems. I-measure and LFM have similar strength correlations, and M^2 is the lowest performing metric, even though its correlation is still strong ($\rho = 0.648$, $r = 0.641$). Hybrid metrics interpolating RBMs and GBMs does even better (see Napoles et al. (2016d) for more details).

7.8 Conclusion

Attending to how humans perceive the quality of the grammatical error corrections, we propose fluency corrections, created two new fluency evaluation sets for GEC, and developed a new fluency metric, GLEU. GLEU more closely models human judgments than previous metrics because it rewards correct edits while penalizing ungrammatical edits, while capturing fluency and grammatical constraints by virtue of using n-grams. We have shown that

Rank		
GLEU	**Interpolated**	**Candidate sentence**
1	2	*Genectic* testing is a personal decision , with the knowledge that there is a *possiblity* that one could be a carrier or not .
2	3	*Genectic* testing is a personal decision , the *kowledge* that there is a *possiblity* that one could be a carrier or not .
3	1	Genetic testing is a personal decision , with the knowledge that there is a possibility that one could be a carrier or not .

Table 7.18: An example of system outputs ranked by GLEU and GLEU interpolated with ER. Words in italics are misspelled.

fluency references together with GLEU are the most robust evaluation for GEC. As GEC systems get more paraphrastic, we also perform an initial investigation into reference-less evaluation for GEC, which will not penalize systems with correct but unseen suggestions.

Chapter 8

Machine Translation and Artificial Data for GEC

This chapter extends our T2T framework to GEC.[1] At the time of this work, the best GEC systems all use machine translation in some form, whether statistical MT as a component of a larger pipeline (Rozovskaya and Roth, 2016) or neural MT (Yuan and Briscoe, 2016). These approaches require a great deal of resources, and in this chapter we propose a lighter-weight approach to GEC by extending our universal framework to the task. We determine that

- Artificially generated rules improve performance by nearly 10%.

- Introducing custom features describing morphological and lexical changes provide a

[1] This work originally appeared as "Systematically Adapting Machine Translation for Grammatical Error Correction" (Napoles and Callison-Burch, 2017).

small performance gain.

- Tuning to a specialized GEC metric is slightly better than tuning to a traditional MT metric (separately found in Sakaguchi et al. (2017b)).
- Larger training data leads to better performance and there is no conclusive difference between training on a clean corpus with minimal corrections and a noisy corpus with potential sentence rewrites.

We have developed and released a tool to automatically characterize the types of transformations made in a corrected text, which are used as features in our model. The features identify general changes such as insertions, substitutions, and deletions, and the number of each of these operations by part of speech. Substitutions are further classified by whether the substitution contains a different inflected form of the original word, such as change in verb tense or noun number; if substitution has the same part of speech as the original; and if it is a spelling correction. These features can be automatically extracted from any aligned English text at the token, sentence or document level. We additionally use them to analyze the outputs generated by different systems and characterize their performance not just with an automatic metric score, but also by the types of transformations it makes, and how they compare to manually written corrections.

Our approach, Specialized Machine translation for Error Correction (SMEC), represents a single model that handles morphological changes, spelling corrections, and phrasal substitutions, and it rivals the performance of the leading neural MT system in 2016 (Yuan and Briscoe, 2016), which uses twice the amount of training data, most of which is not publicly available. For GEC, the analysis provided in this work will help improve future efforts and can be used to inform approaches rooted in both neural and statistical MT. More broadly, we demonstrate that our unified framework can be applied to novel T2T tasks, and with some thoughtful customizations, perform at least as well as data-hungry approaches.

8.1 Customizing the T2T Framework

This section describes how we customized each of the components of our unified framework for GEC.

8.1.1 Grammar

For sentence compression and text simplification, we exploit the expressiveness of a large-scale paraphrase resource and guide the SMT decoder to choose paraphrases appropriate for the respective task (Chapters 4 and 6). However, while some transformations in GEC are paraphrastic, the majority have a corrupted source text that contains a mistake, such as to the spelling or word form. These changes will not be found in a paraphrase corpus, which has well-formed text on both sides of each paraphrase. Therefore we take a hybrid approach and extract a grammar from a parallel GEC corpus and augment that grammar with automatically generated rules to address spelling and morphological mistakes.

A limiting factor on MT-based GEC is the available training data, which is small when compared to the data available for bilingual MT, which commonly uses 100s of thousands or millions of aligned sentence pairs. We hypothesize that artificially generating transformation rules may overcome the limitations imposed by lack of sufficiently large training data and improve performance. Particularly, the prevalence of spelling errors is amplified in sparse data due to the potentially infinite possible misspellings and large number of OOVs. Previous work has approached this issue by including spelling correction as a step in a pipeline (e.g., Rozovskaya and Roth, 2016).

Our solution is to artificially generate grammar rules for spelling corrections and morphological changes. For each word in the input, we query the Aspell dictionary with PyEnchant[2] for spelling suggestions and create new rules for each correction, such as

$$\textit{publically} \rightarrow \textit{public ally}$$

$$\textit{publically} \rightarrow \textit{publicly}$$

Additionally, sparsity in morphological variations may arise in data sets. Wang et al. (2014) approached this issue with factored MT, which translates at the sub-word level. Instead, we also generate artificial translation rules representing morphological transformations using RASP's morphological generator, `morphg` (Minnen et al., 2001). We perform POS tagging with the Stanford POS tagger (Toutanova et al., 2003) and create rules to switch the plurality of nouns (e.g., singular ↔ plural). For verbs, we generate rules that change that verb to every other inflected form, specifically the base form, third-person singular, past tense,

[2] `https://pythonhosted.org/pyenchant/`

past participle, and progressive tense (e.g., *wake*, *wakes*, *woke*, *woken*, *waking*). Generated words that did not appear in the PyEnchant dictionary were excluded.

8.1.2 Features

Junczys-Dowmunt and Grundkiewicz (2016) used a large number of sparse features for a phrase-based MT system that achieved state of the art performance on the CoNLL-2014 test set. Unlike that work, which uses a potentially infinite amount of sparse features, we choose to use a discrete set of feature functions that are informed by this task. Our feature extraction relies on a variety of pre-existing tools, including fast-align for word alignment (Dyer et al., 2013), trained over the parallel FCE, Lang-8, and NUCLE corpora; PyEnchant for detecting spelling changes; the Stanford POS tagger; the RASP morphological analyzer, `morpha` (Minnen et al., 2001); and the NLTK WordNet lemmatizer (Bird et al., 2009). Our toolkit is available from `https://github.com/cnap/smt-for-gec`.

Given a grammatical rule and an alignment between tokens on the LHS and RHS, we tag the tokens with their part of speech and label the lemma and inflection of nouns and verbs with `morpha` and the lemma of each adjective and adverb with the WordNet lemmatizer. We then collect features for individual operations and rule-level qualities. An operation is defined as a deletion, insertion, or substitution of a pair of aligned tokens (or a token aligned with ε). An aligned token pair is represented as (l_i, r_j), where l_i is a token on the LHS at index i, and similarly r_j for the RHS. The exact features are

- **Deletions**

  - deleted(l_i tag)
  - deleted(l_i class)

The tag is the specific PTB part-of-speech tag (e.g., *NN*, *NNS*, *NNP*, etc.) and class is the broader word class (e.g., *noun*).

- **Insertions**
  - inserted(r_j tag)
  - inserted(r_j class)
- **Substitutions**
  - substituted(r_j tag)
  - substituted(r_j class)
  - substituted(l_i tag, r_j tag)

Morphological features:

  - inflection change(l_i, r_j)
  - same tag, different word(l_i, r_j)
  - different tag, different word(l_i, r_j)

Spelling features:

  - is in dictionary(l_i)

  - is a suggested correction(l_i, r_j)

  Counts of spelling corrections are weighted by the probability of r_j in an English Gigaword language model.

- **Rule-level features**
  - character Levenshtein distance(*LHS*, *RHS*)
  - token Levenshtein distance(*LHS*, *RHS*)
  - $\frac{\text{\# tokens}(\mathit{RHS})}{\text{\# tokens}(\mathit{LHS})}$
  - $\frac{\text{\# characters}(\mathit{RHS})}{\text{\# characters}(\mathit{LHS})}$

In previous MT approaches to GEC, Levenshtein distance has been a feature for tuning (Felice et al., 2014; Junczys-Dowmunt and Grundkiewicz, 2014; Junczys-Dowmunt and Grundkiewicz, 2016), and Junczys-Dowmunt and Grundkiewicz (2016) also used counts of deletions, insertions, and substitutions by word class. They additionally had sparse features, with counts of each lexicalized operation, e.g., substitute(*run*, *ran*), which we avoid by abstracting away from the lemmas and instead counting the operations by part of speech and indicating if the lemmas matched or differed for substitutions.

8.1.3 Metric

The decoder identifies the most probable derivation of an input sentence from the translation grammar. Derivations are scored by a combination of a language model score and weighted feature functions, and the weights are optimized to a specific metric during the tuning phase. In Section 7.7.2, we showed that MT metrics like BLEU are not sufficient for evaluating GEC, nor are they appropriate for tuning MT systems for GEC (Junczys-Dowmunt and Grundkiewicz, 2016). Fundamentally, MT metrics do not work for GEC because the output is usually very similar to the input, and therefore the input already has a high metric score. To address this issue, we tune to GLEU, which was specifically designed for evaluating GEC output. We chose GLEU instead of M^2 because the latter requires a token alignment between the input, output, and gold-standard references, and assumes only minimal, non-overlapping changes have been made. GLEU, on the other hand, measures n-gram overlap and therefore is better equipped to handle movement and changes to larger spans of text.

8.2 Experiments

For our experiments, we use the Joshua 6 toolkit (Post et al., 2015). Tokenization is done with Joshua and token-level alignment with fast-align (Dyer et al., 2013). All text is lowercased, and we use a simple algorithm to recase the output (Table 8.1). We extract a hierarchical phrase-based translation model with Thrax (Weese et al., 2011) and perform parameter tuning with pairwise ranked optimization in Joshua.

Our training data is from the Lang-8 corpus (Mizumoto et al., 2011), which contains 1 million parallel sentences (563k of which contain corrections),[3] and we tune to the JFLEG

[3] Going forward, we will refer to the size of training data by the number of corrected sentence pairs.

1. Generate POS tags of the cased input sentence
2. Label proper nouns in the input
3. Align the cased input tokens with the output
4. Capitalize the first alphanumeric character of the output sentence (if a letter).
5. For each pair of aligned tokens (l_i, r_j), capitalize r_j if l_i is labeled a proper noun or r_j is the token "i".

Table 8.1: A simple recasing algorithm, which relies on token alignments between the input and output.

tuning set (751 sentences) and evaluate on the JFLEG test set (747 sentences). We use an English Gigaword 5-gram language model.

We evaluate performance with two metrics, GLEU and M^2, which have similar rankings and match human judgments on the JFLEG corpus (Napoles et al., 2017c). The baseline is an unmodified MT pipeline trained on the Lang-8 corpus, optimized to BLEU with no specialized features, and we compare our performance to the leading neural machine translation system at the time, Yuan and Briscoe (2016), YB16. We additionally report metric scores for the human corrections, which we determine by evaluating each reference set against the other three and reporting the mean score. Our fully customized model with all modifications, Specialized Machine translation for Error Correction (SMEC$^{\text{+morph}}$), scores lower than YB16 according to GLEU but has the same M^2 score. Qualitative examination of the output reveals many incorrect or unnecessary number of tense changes, and automatic analysis reveals that it makes significantly more inflection changes than the hu-

System	GLEU	M^2	Sentences changed	LD
Baseline	54.9	36.0	39%	0.7
SMEC$^{+\text{morph}}$	57.9	52.3	**88%**	**2.8**
SMEC	58.3	52.2	85%	2.5
YB16	**58.4**	**52.3**	73%	1.9
Human	62.1	63.6	77%	3.1

Table 8.2: Results on the JFLEG test set. In addition to the GLEU and M^2 scores, we also report the percent of sentences changed from the input and the mean Levensthein distance (tokens).

mans or YB16 (detected with the same method described in Section 8.1.2), from which we can conclude that the morphological rules errors are applied too liberally. If we remove the generated morphological rules but keep the spelling rules (SMEC), performance improves by 0.4 GLEU points and decreases by 0.1 M^2 points—but, more importantly, the system makes more conservative morphological changes. Therefore, we consider SMEC, the model without artificial morphological rules, to be our best system.

There is a disparity in the GLEU and M^2 scores for the baseline: the baseline GLEU is about 5% lower than the other systems but the M^2 is 30% lower. This can be attributed to the lesser extent of changes made by the baseline system which results in low recall for M^2 but which is not penalized by GLEU. The human corrections have the highest metric scores, and make changes to 77% of the sentences, which is in between the number of sentences changed by YB16 and SMEC, however the human corrections have a higher mean edit distance, because the annotators made more extensive changes when a sentence needed to be corrected than any of the models.

The metrics only give us a high-level overview of the changes made in the output. With error-coded text, the performance by feature type can be examined with M^2, but this is not possible with GLEU or the JFLEG corpus, which is not coded. To investigate the types of changes made by a system on a more granular level, we apply the feature extraction method described in Section 8.1.2 to quantify the morphological and lexical transformations. While we developed this method for scoring translation rules, it can work on any aligned text. We calculate the number of each of these transformations made by to the input by each system and the human references, determining significant differences with a paired t-test ($p < 0.05$). Figure 8.1 contains the mean number of these transformations per sentence made by SMEC, YB16, and the human-corrected references, and Figure 8.2 shows the number of operations by part of speech. Even though the GLEU and M^2 scores of the two systems are nearly identical, they are significantly different in all of the transformations in Figure 8.1, with SMEC having a higher edit distance from the original, but YB16 making more insertions and substitutions. Overall, the human corrections have a significantly more inserted tokens than either system, while YB16 makes the most substitutions and fewer deletions than SMEC or the human corrections. The bottom plot displays the mean number of operations by part of speech (operations include deletion, insertion, and substitution). Both systems and the human corrections display similar rates of substitutions across different parts of speech, however the human references have significantly more preposition and verb operations and there are significant differences between the determiner and noun operations made by YB16 compared to SMEC and the references. This information can be further analyzed by part of speech and edit operation, and the same

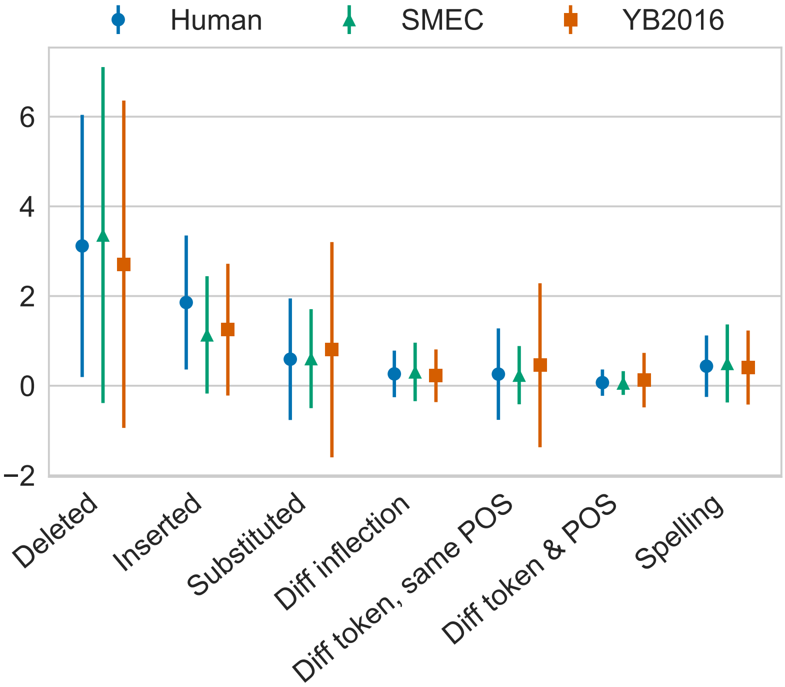


Figure 8.1: Mean tokens per sentence displaying certain changes from the input sentence.

information is available for other word classes.

8.3 Model Analysis

We wish to understand how each component of our model contributes to its performance, and therefore train a series of variations of the model, each time removing a single customization, specifically: the optimization metric (tuning to BLEU instead of GLEU; SMEC $^{-\text{GLEU}}$), the features (only using the standard MT features; SMEC$^{-\text{feats}}$), and eliminating artificial rules (SMEC $^{-\text{sp}}$). The impact of training data size will be investigated separately in Section 8.3.1. We computed the automatic metric scores of each model variation and performed the automatic edit analysis described in Section 8.1.2. In Table 8.3, we report

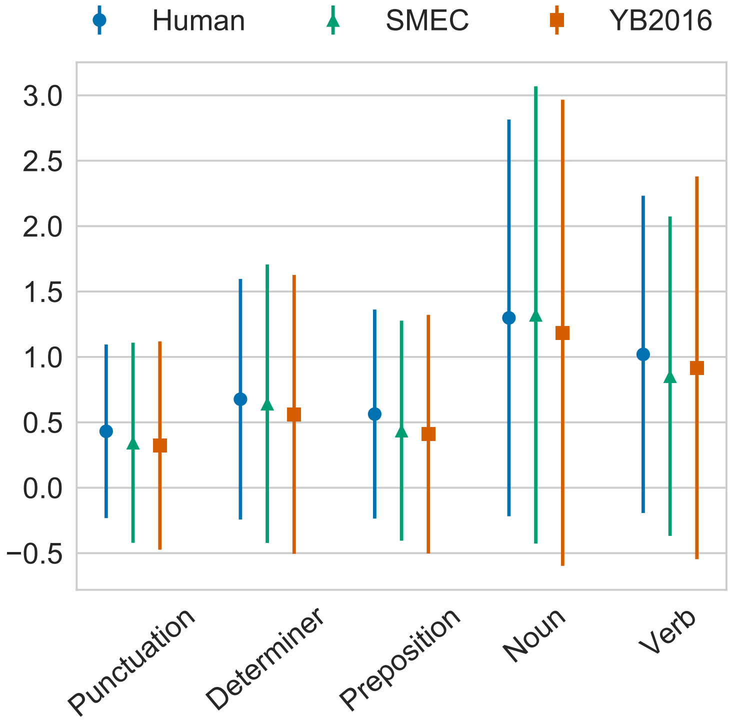


Figure 8.2: Mean number of operations (deletions, insertions, and substitutions) per sentence by part of speech.

the net metric increase or decrease compared to the full model, and the percent increase or decrease for each of the features. Changing the metric from GLEU to BLEU significantly decreases the amount of change made by the model, SMEC$^{-\text{GLEU}}$, with a 60% lower edit distance than SMEC, and at least 50% fewer of almost all transformations. The GLEU score of this system is nearly 1 point lower, however there is almost no change in the M^2 score, indicating that the changes made were appropriate, even though they were fewer in number. Tuning to BLEU causes fewer changes because the input sentence already has a high BLEU score due to the high overlap between the input and reference sentences. GLEU encourages more changes by penalizing text that should have been changed in the output.

Removing the custom features (SMEC$^{-\text{feats}}$) makes less of a difference in the GLEU score, however there are significantly more determiners added and more tokens are substituted with words that have different lemmas and parts of speech. This suggests that the specialized features encouraged morphologically-aware substitutions, reducing changes that did not have semantic or functional overlap with the original content. Removing the artificially generated spelling rules (SMEC$^{-\text{sp}}$) had the greatest impact on performance, with a nearly 10% decrease in GLEU score and 20% decrease in M^2. Without spelling rules, significantly fewer tokens were inserted in the corrections across all word classes. We also see a significantly greater number of substitutions made with words that had neither the same part of speech or lemma as the original word, which could be attributed to sparsity in the presence of spelling errors which is addressed with the artificial grammar.

Table 8.4 contains examples of sentences from the test set that illustrate some of these observations. These ungrammatical sentences range from one that can easily be corrected by an English speaker using *minimal* edits; to a sentence that requires more significant changes and inference but has an obvious meaning; to a sentence that is garbled and does not have an immediately obvious correction, even to a native English speaker. We show each of the original sentences next to one of the human-corrected references, and the corrections generated by SMEC and SMEC variations without GLEU and without spelling rules. The reference correction contains more extensive changes than the automatic systems and makes spelling corrections not found by the decoder (*engy* → *energy*) or inferences in the instance of the garbled third sentence, changing *lrenikg* → *Ranking*. SMEC makes many spelling corrections and makes more insertions, substitutions, and deletions than the two SMEC variations. However, the artificial rules also cause some bad corrections, found in the third example changing *studens* → *stud-ens*, while the intended word, *students*, is obvious to a human reader.[4] When optimizing to BLEU instead of the custom metric (SMEC $-$GLEU), there are fewer changes and therefore the output is less fluent. In the first example, SMEC$^{-\text{GLEU}}$ applies only one spelling change even though the rest of the sentence has many small errors that were all corrected in SMEC, such as missing determiner and extra auxiliary. The same pattern is visible in the other two examples. Finally, without the artificial rules, SMEC $-$sp fixes only a fraction of the spelling mistakes—however it is the only system that correctly changes *studens* → *students*. Independent from these modifications, the capitalization issues present in the input were all remedied by our recasing algorithm, which improves the metric score.

8.3.1 Impact of Training Data

Lang-8 is the largest publicly available parallel corpus for GEC, with 1 million tokens and approximately 563k corrected sentence pairs, however this corpus may contain noise due to automatic alignment and the annotators, who were users of the online Lang-8 service and may not necessarily have provided accurate or complete corrections. Two other corpora, FCE and NUCLE, contain annotations by trained English instructors and absolute alignments between sentences, however each is approximately 20-times smaller than Lang-8.

[4] To filter out bad spelling candidates, we could further extend the generated rules by applying vector-space models as we have done for sentence compression (Section 4.3). This is an area of future work.

		SMEC		
Quality		**−GLEU**	**−feats**	**−sp**
GLEU		−0.7	−0.2	−3.9
M^2		−0.1	−0.5	−9.5
Edit dist		−60%		−11%
Deleted		−51%	−7%	−4%
Inserted		−46%		−24%
Substituted		−37%		+9%
Diff inflection		−53%		
Diff token		−18%	+9%	
Diff token+POS		−29%	+24%	+31%
Spelling		−35%	+8%	
Determiner		−51%		−7%
	del	−47%	−5%	
	ins	−70%	+39%	−30%
	sub	−56%		
Preposition		−52%		
	del	−51%		
	ins	−45%	−10%	−22%
	sub	−42%		
Noun		−40%	−6%	−10%
	del	−45%	−9%	−12%
	ins	−38%	−7%	−13%
	sub	−30%		
Verb		−57%	−5%	−11%
	del	−61%	−6%	−7%
	ins	−53%	−13%	−40%
	sub	−50%		
Punctuation		−52%		
	del	−47%	−5%	
	ins	−84%		−30%
	sub			

Table 8.3: Modifications of SMEC, reporting the mean occurrence of each transformation per sentence, when there is a significant difference ($p < 0.05$ by a paired t-test).

Orig	Unforturntly , almost older people can not use internet , in spite of benefit of internet .
Human	**Unfortunately** , **most** older people can not use **the** internet , in spite of **benefits** of **the** internet .
SMEC	**Unfortunately** , **most** older people can not use **the** internet , in spite of **the benefits** of **the** internet .
SMEC^{-GLEU}	**Unfortunately** , almost older people can not use internet , in spite of benefit of internet .
SMEC^{-sp}	Unforturntly , □ older people can not use **the** internet , in spite of **the benefits** of **the** internet .
Orig	becuse if i see some one did somthing to may safe me time and engy and it wok ’s i will do it .
Human	**Because** if **I** see **that someone** did **something that** may **save** me time and **energy** and it **works I** will **also** do it .
SMEC	**Because** if **I** see □ one did **something** □ may **save** me time and **edgy** and □ **work** □ **, I** will do it .
SMEC^{-GLEU}	**Because** if **I** see some one did **something** to may **save** me time and **edgy** and it wok ’s **I** will do it .
SMEC^{-sp}	**Because** if **I** see □ one **somthings** □ may **save** me time and engy □ **work** □ **I** will do it .
Orig	lrenikg the studens the ideas have many advantegis :
Human	**Ranking** the **students ’** □ ideas **has** many **advantages** .
SMEC	**Linking** the **stud-ens** □ ideas have many **advantages** :
SMEC^{-GLEU}	**Linking** the **stud-ens** the ideas have many **advantages** :
SMEC^{-sp}	**Lrenikg** □ **students** □ ideas have □ advantegis :

Table 8.4: Example corrections made by a human annotator, SMEC, and two variations: trained on BLEU instead of GLEU (SMEC$^{-\text{GLEU}}$) and without artificial spelling rules (SMEC$^{-\text{sp}}$). Inserted or changed text is in **bold** and deleted text is indicated with □.

We wish to isolate the effect of size and source of training data has on system performance, and therefore randomly sample the Lang-8 corpus to create smaller training sets that are the same size as FCE or NUCLE (21.5k corrected sentences) and FCE and NUCLE (43k corrected sentences). From Lang-8 and FCE/NUCLE, we extract grammars learned over 21.5k or 43k corrected sentences and train models following the same procedure described above. We hypothesized that including artificial rules may help address problems of sparsity in the training data, and therefore we also train additional models with and without spelling rules to determine how artificial data affects performance as the amount of training data increases. Figure 8.3 shows the relative GLEU scores of systems with different training data sizes and sources, before and after adding artificial spelling rules.

For Lang-8, more data increases performance, however there is no clear relationship between size and performance on the FCE+NUCLE data. Training on a small set of 21.5k Lang-8 sentence pairs results in much worse performance than training on the same size FCE or NUCLE data, however when using twice the training data, Lang-8 is better than FCE/NUCLE, suggesting that more data negates the presence of noise and the sentential rewrites present in Lang-8 are better for training a GEC system. These results suggest that, with even more data, performance would continue to improve, and SMEC may outperform YB16 if trained over the larger CLC.

For all models except for that trained on 21.5k Lang-8 corrected sentences, adding artificial spelling rules improves performance by about 4 GLEU points (adding spelling rules to FCE training data only causes a 2-point GLEU improvement). The amount of

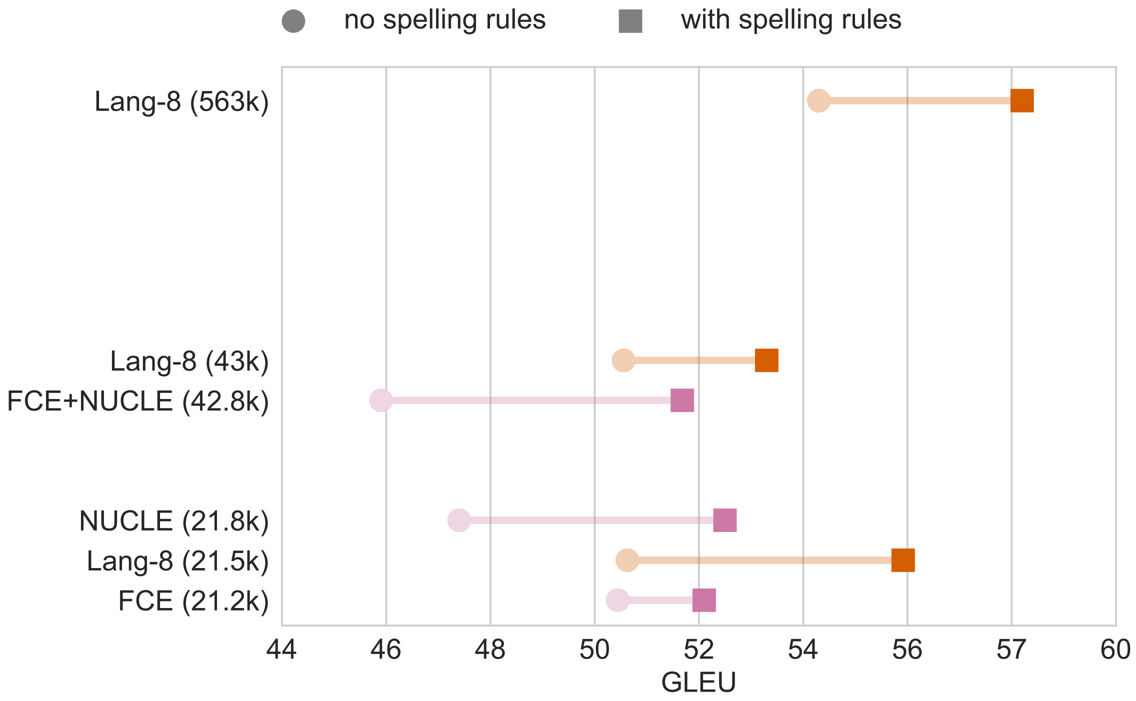


Figure 8.3: GLEU scores of SMEC with different training sizes, with and without artificial rules.

performance does not change related to the size of the training data, however the consistent improvement supports our hypothesis that artificial rules are useful to address problems of data sparsity.

8.4 Conclusion

This chapter extended our unified framework to GEC, a task fundamentally different from sentence compression and simplification because the source text is ill formed by definition. We presented a systematic investigation into how to customize the components of the T2T framework for GEC. We have found that extending the translation grammar with artificially generated rules for spelling correction can increase the M^2 score by as much as 20%. The amount of training data also has a substantial impact on performance, increasing GLEU and M^2 scores by approximately 10%. Tuning to a specialized GEC metric and using custom features both help performance but yield less considerable gains. The performance of our model, SMEC, is on par with the current state-of-the-art GEC system, which is neural MT trained on twice the available training data—and our analysis suggests that the performance of SMEC would continue to improve if trained on that amount of data. This research has demonstrated the flexibility and power of our T2T framework to a broad range of tasks.

Part IV

Conclusion

This thesis has examined different T2T tasks and provided a basis for applying paraphrasing to transforming text subject to the constraints of each task. We have identified shortcomings of previous evaluation methodologies and the corpora used for evaluating the tasks, and created new datasets and new metrics for evaluating them. We have also presented a unified framework for T2T that harnesses advances in statistical machine translation to monolingual translation tasks. With this framework, we have achieved state-of-the-art performance without relying on large-scale parallel corpora annotated for each specific task, which frequently do not exist and are costly to create. The T2T tasks in this thesis range in complexity, starting with a simple objective (shortening text) to a more complicated one (increasing readability), and we end with a task that takes ill-formed text as input with multiple objectives (correcting for spelling, grammar, and fluency). Our approaches are summarized in Table 8.5.

For each of these tasks, we had to make task-specific modifications, for which a small-scale, qualitative analysis of the data was necessitated. For sentence compression, we introduced deletion rules for certain classes of words: adjectives, adverbs, and determiners.

The rules we introduced for sentence simplification were fewer: just deletion of adjectives. For grammatical error correction, we could not use the general paraphrase corpus since it did not include examples of ill-formed text, and therefore we created a set of rules that harnessed existing technology in spell-checking and morphological analysis. For each of these tasks, we have demonstrated how to achieve state of the art performance by combining a small set of hand-crafted rules with the limited resources existing for that task. This approach can be extended for other T2T tasks such as style transfer.

Contributions

In Part I we examined the task of automatic sentence compression. While earlier approaches to the task treated it as primarily a deletion task, we describe how paraphrasing can be applied to the task and examined the ramifications of paraphrasing to existing evaluation methodologies, both automatic and manual. We quantified the impact of compression rate on perceived quality and made specific recommendations to accurately evaluate the candidate compressions (Chapter 3). We additionally explored the capability of paraphrasing to sentence compression, first by greedily selecting paraphrases to shorten a text, and next by modifying the statistical machine translation pipeline (Chapter 4).

Part II considered the task of sentence simplification. We analyzed a large-scale, naturally occurring parallel corpus of simplified text, quantified the difference between simplified and unsimplified text, and identified shortcomings of using a corpus of automatically aligned sentences for the task (Chapter 5).

We developed a novel metric for the task that accounted for fluency, meaning preservation, and simplification. We applied this metric with a large-scale paraphrase database in a modified machine-translation approach to generate simplifications that surpass those generated in earlier approaches in terms of meaning-preservation and readability, and required only a small amount of task-specific annotated data (Chapter 6).

Finally, Part III visited the task of grammatical error correction and began with a thorough examination of the approaches for evaluating GEC output and identified a mismatch between the stated aim of the task and the evaluation methodologies (Chapter 7). We proposed a new definition of the task supported by empirical evidence, defined new metrics, and created new corpora for GEC as fluency correction. Chapter 8 described how to apply methods from statistical machine translation to GEC and identified how to overcome a limited amount of annotated parallel corpora using artificial rules.

Future Directions

Advances in T2T have used neural sequence-to-sequence models for these tasks with great promise, subject to some problems with adequacy. There are examples of these advances in each of the tasks: sentence compression (Rush et al., 2015), text simplification (Nisioi et al., 2017), and grammatical error correction (Schmaltz et al., 2017), for example. One issue that we have identified with deep-learning approaches is the introduction of meaning-changing operations (Sakaguchi et al., 2017a). Koehn and Knowles (2017) described six issues for neural machine translation (NMT), many of which are relevant for monolingual

T2T: particularly that NMT systems may sacrifice adequacy for fluency and does not function well in low-resource settings or for longer sentences. As of this writing, a similar analysis yet needs to be done for monolingual T2T to fully understand the relative benefits of NMT-based approaches for a lower-resource tasks.

That being said, many of the findings in this thesis can be applied to future work: Koehn and Knowles showed that data-hungry neural approaches are less effective than SMT-based models when there is not sufficient training data, and our unified framework is demonstrably better in these settings than other approaches that require more annotated data—including NMT-based approach for GEC (Yuan and Briscoe, 2016). More generally, our findings into evaluation methodologies are universally applicable: automatic metrics (and, by extension, loss functions) should be validated against human judgments; the output of a model needs to be manually examined to ensure that the evaluation is not picking up on a related side effect, like unchanged text in the output; and the scientific validity of any result depends on careful scrutiny of the above.

The work in this thesis has advanced a deeper understanding of unbiased approaches for evaluating monolingual sentence rewriting and proposed automatic metrics as well as textual resources to advance the respective tasks. These findings can be applied to a multitude of generation tasks, not limited to the three explored herein. Unbiased evaluation is crucial for system development as current models approach human levels of quality. This thesis provides a framework for achieving state-of-the-art quality using existing resources for new tasks, minimizing the amount of annotated data necessary for training, tuning, and testing;

and we have additionally demonstrated how to ensure reliable, interpretable evaluation of system outputs with automatic and manual methodologies.

Task	**Grammar**	**Grammar augmentation**	**Tuning data**	**Features**	**Objective**
Sentence compression	Monolingual paraphrases (42M)	Deletion rules (adjectives, adverbs, and determiners)	Pairs of multiple reference translations (1k)	Word count, difference in word count, and difference in average word length	PRÉCIS (n-gram overlap with “verbosity penalty”)
Text simplification	Monolingual paraphrases (42M)	Identity rules (closed class words) and deletion rules (adjectives)	Manually validated aligned Simple and English Wikipedia sentences (1.4k)	Word length (syllables), difference in word count, count of Basic English words	GLiB (n-gram overlap penalizing overlap with source and candidates with a higher grade level than the source)
Grammatical error correction	Grammar extracted from parallel GEC corpus (560k sentences)	Spelling suggestions and morphological changes to nouns and verbs	Count of operations by part of speech, count of spelling errors, count of inflection changes	Sentences by English language learners (750)	GLEU (n-gram overlap, penalizing false negatives)

Table 8.5: Customizations to our unified T2T framework for each task in this thesis.

Part V

Appendices

Appendix A

Description of Manual Evaluations

This appendix describes the experimental setup for collecting judgments and human-annotated data, including the guidelines, participant profile, and quality control mechanisms.

A.1 Crowdsourcing

A bottleneck to developing robust models is the evaluation of output. Ideally, a trained human familiar with the task would manually inspect output and consult with other trained experts to eliminate bias to make the final assessment. Evaluation of this time is costly, both in terms of the expense of hiring and training judges and in terms of the time it takes for a small number of judges to evaluate a large set of sentences. One alternative seen in the literature is to manually evaluate a small number of instances, often 100, however such a small sample cannot reliably represent the overall quality of the entire output. Another option is to use automatic techniques in place of expert manual annotation, however this thesis expounds on many of the fallacies and improper applications of automatic evaluation. For all of the experiments conducted as a part of this thesis, we crowdsource evaluation of system outputs.

We recruit annotators from Amazon Mechanical Turk `https://mturk.com` for data collection. The use of crowdsourcing as a technique for data collection, annotation, and evaluation has become accepted in the NLP community. However concerns about the quality of crowdsourced annotations persist. We have paid special attention to quality in the design of our experiments, specifically:

1. Eliminating Bias

1. Redundancy: Collect multiple judgments for each item.

2. Randomization: Randomize the order in which items are displayed.

2. Quality Control

3. Speed: Check the amount of time taken for each task.

4. Embedded test questions: In each task, include questions for which there is a known answer and check performance against this.

5. Manual examination: Spot check the output.

6. Credentials:

Strength	**Absolute value**
Very Strong	$0.80-1.0$
Strong	$0.60-0.79$
Moderate	$0.40-0.59$
Weak	$0.20-0.39$
Negligible/Very Weak	$0.00-0.19$

Table A.1: Correlation strengths (Evans, 1996).

(a) Location: Limit participants of a task to IP addresses in specific countries (the U.S. and Canada, for our experiments).

(b) Previous Behavior: Only allow workers with a high previous acceptance rate and that have completed a certain number of previous tasks.

(c) Qualifying Task: Workers must demonstrate satisfactory completion of a preliminary task.

A.2 Data Evaluation

One of the most commonly used methods for calculating "goodness" of data is correlation, which ranges from -1 to 1. The correlation coefficient can be described by the strength of association. Table A.1 describes the vocabulary for describing correlation strengths.

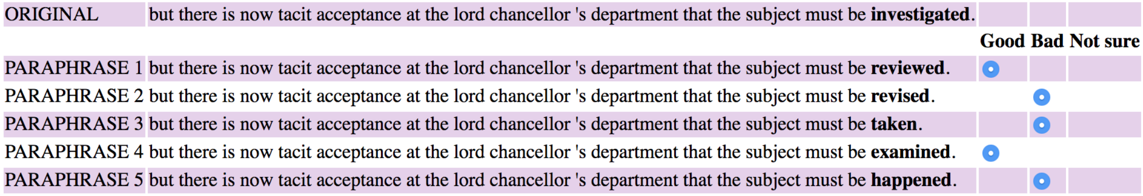
A.3 Evaluating Paraphrase Scores

The paraphrase candidates were ranked by human judges in an experiment conducted on Amazon's Mechanical Turk. Each judge was shown the original sentence and that same sentence with each paraphrase candidate substituted in. They were asked to score each paraphrase on two Likert-like scales based on the extent to which it preserved the meaning and affected the grammaticality of the sentence (1 to 5, with 5 being perfect). We collected 5 judgments per paraphrase context, and the participants were restricted to those in the United States with a HIT approval rate of at least 95%. Controls were additionally embedded into the judgments and participants were disqualified if they answered the controls incorrectly at a rate of 20% more than other participants. A positive control was a paraphrase identical to the original phrase, and the correct answer was to rate this version of a sentence with a 5 for meaning and grammar. The negative control was a randomly selected phrase and the correct answer was to rate this version with a 1 for meaning.

A.3.1 Judge Paraphrase Quality

Please read the groups of sentences below. Each group contains an original sentence along with ten different versions of it, which are created by replacing a phrase with automatically generated paraphrases. Your job is to say whether the automatically generated paraphrases are GOOD or BAD for the sentence. You should ignore punctuation in this HIT. If the new paraphrase does not change the meaning of the sentence and the sentence is still grammatical, then the paraphrase is good. If the paraphrase is the same as the original phrase, it is also good. A new paraphrase that makes the sentence mean something different or not sound right is bad.

Here is an example of how we graded a set of paraphrases:



ORIGINAL	but there is now tacit acceptance at the lord chancellor 's department that the subject must be **investigated**.			
		Good	**Bad**	**Not sure**
PARAPHRASE 1	but there is now tacit acceptance at the lord chancellor 's department that the subject must be **reviewed**.	●		
PARAPHRASE 2	but there is now tacit acceptance at the lord chancellor 's department that the subject must be **revised**.		●	
PARAPHRASE 3	but there is now tacit acceptance at the lord chancellor 's department that the subject must be **taken**.		●	
PARAPHRASE 4	but there is now tacit acceptance at the lord chancellor 's department that the subject must be **examined**.	●		
PARAPHRASE 5	but there is now tacit acceptance at the lord chancellor 's department that the subject must be **happened**.		●	

While the first four paraphrases are all grammatical, only examined and reviewed are similar in meaning to the original phrase, and therefore they are the only "good" paraphrases in this example.

A.4 Evaluating Sentence Compressions

Manual evaluation used Amazon's Mechanical Turk with three-way redundancy and positive and negative controls to filter bad workers. Positive controls were the gold-standard human-written compressions, and negative controls were the original sentences with 50% of the words randomly deleted.

A.4.1 Judge the Quality of Similar Sentences

Please read the groups of sentences below. Each group contains an original sentence along with five different versions of it, which are created automatically by a computer. Your job is to grade the quality of the automatically generated sentences with two 5-point scales: meaning and grammar.

APPENDIX A. DESCRIPTION OF MANUAL EVALUATIONS

Meaning

- *Perfect (5):* All of the meaning of the original sentence is retained, and nothing is added.
- *Minor differences (4):* The meaning of the original sentence is retained, although some minor information may be deleted or added without too great a difference in meaning.
- *Moderate differences (3):* Some of the meaning of the original sentence is retained, although a non-trivial amount of information was deleted or added.
- *Substantially different (2):* Substantial amount of the meaning is different.
- *Completely different (1):* The new sentence doesn't mean anything close to the original sentence.

Grammar

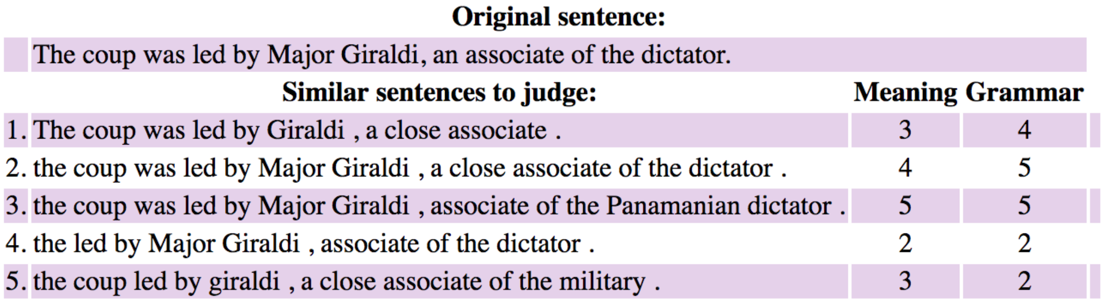
- *Perfect (5):* The new sentence is perfectly grammatical.
- *OK but awkward (4):* The sentence is grammatical, but might sound slightly awkward.
- *One error (3):* The sentence has an error (such as an agreement error between subject and verb, a plural noun with a singular determiner, or the wrong verb form).

- *Many errors (2):* The sentence has multiple errors or omits words that would be required to make it grammatical.

- *Ungrammatical (1):* The sentence is totally ungrammatical.

In this HIT, you should ignore capitalization. Some of the original sentences may not be 100% grammatical, but you should not let this affect your judgment of the new sentences. If sentences are repeated just give them the same scores.

Here is an example of how we graded a set of sentences:



Original sentence:

The coup was led by Major Giraldi, an associate of the dictator.

	Similar sentences to judge:	**Meaning**	**Grammar**
1.	The coup was led by Giraldi , a close associate .	3	4
2.	the coup was led by Major Giraldi , a close associate of the dictator .	4	5
3.	the coup was led by Major Giraldi , associate of the Panamanian dictator .	5	5
4.	the led by Major Giraldi , associate of the dictator .	2	2
5.	the coup led by giraldi , a close associate of the military .	3	2

We gave sentence #1 a score of 4 for grammar because it sounds like there is something missing from the end of the sentence (a close associate of whom?). Sentence #4 received a score of 2 for meaning because it doesn't mention the coup. Sentences #4 and #5 are incomplete sentences and so both got a grammar score of 2.

A.5 Compare the Simplicity of Sentences

We ran two HITs that asked judges to compare the simplicity (or readability) of different sentences. In the first HIT, judges were asked to read two versions of a sentence and choose the more simple version.

The second HIT collected rankings of five sentences by simplicity. We presented judges with five versions of an input sentence, including the original sentence, the manually simplified "gold-standard" sentence, and three automatic simplifications. Each ranking was completed by 3 separate judges, and for controls we made sure that the input sentence had a lower ranking than the gold-standard simplification

A.5.1 Which Sentence Is More Simple?

In this HIT, you will read pairs of sentences. Your task is to decide which of the two sentences is more *simple* or easier to read and understand. The options are:

- Sentence 1 is more simple (easier to read) than Sentence 2.
- Both sentences are the same difficulty.
- Sentence 1 is more difficult than Sentence 2.

In this HIT, the two sentences should be describing the same thing and one is a simplified version of the other. However, some sentence pairs may describe different things. If you think that this is the case, check the box next to *Neither sentence is a simplification of the other*. For example, #2 below is a simplified version of #1 but #3 is not a simplified version of either sentence.

1. The War of 1812 was a military conflict fought between the forces of the United States of America and those of the British Empire, stemming from impressment of

American merchant sailors into the Royal Navy and British support of American Indian tribes against American expansion.

2. The United States and the British Empire fought each other in the War of 1812 for a variety of reasons.

3. During the War of 1812, the British blockaded the Atlantic coast of the U.S. and mounted large-scale raids in the later stages of the war.

4. The United States and the British Empire fought each other in the War of 1812. One cause of the war was when the British began kidnapping American sailors and forcing them to serve in the British navy. Another cause was British support of American Indian tribes against American expansion.

It is possible that one of the sentences in the pair is actually more than one sentence (or a paragraph). In this case, you should consider whether that paragraph is easier or more difficult to read than the other sentence in the pair. For example, #4 is a simplification of #1, even though #4 is longer. Length is not a reliable way to decide which sentence is more simple.

A.5.2 Rate Simple Sentences

Please read the sentences and vote on the one that you think is the easiest to read and understand in each. Each sentence is an automatically generated simplification of the original sentence.

You should consider the following factors when evaluating the sentences:

- Does this sentence make more sense than the others?
- Is the grammar good?
- Is it easy to understand what the sentence means?

You should rank every sentence in a group, and ties are OK.

A.6 Collecting Grammatical Corrections

We ran three different experiments collecting corrections of sentences written by English language learners. The first experiment asked participants to make only *minimal edits* to correct the sentences. In the second and third experiments, participants were instructed to edit the sentences so that they were fluent to a native speaker of English. For all experiments, we collected multiple annotations and had participants complete a preliminary qualifying test to become eligible.

A.6.1 Correct Sentences with Minimal Edits

Please read the following instructions carefully. Please correct grammatical mistakes in the following sentences. In many cases, there are lots of ways to correct a sentence; please prefer corrections that require the smallest number of changes to the original sentence, even if the resulting sentence is slightly awkward or non-native sounding. Not all sentences require corrections.

The sentences are taken from student essays. For context, the previous sentence in the essay is also presented, greyed out. *Examples* are found in Table A.2.

Instructions: Please correct only the **BOLD, UNDERLINED** sentence. The gray sentence is given as a context. NOTE: Too much rewriting might be rejected. Please keep your edits as minimal as possible.

A.6.2 Correct Sentences with Fluency Edits

Please correct the following sentences to make them sound more natural to a native speaker of English. You should also fix grammatical mistakes, but focus on changing the sentences to remove awkward phrases and to follow standard written usage conventions. Not all sentences require corrections.

The sentences are taken from student essays. For context, the previous sentence in the essay is also presented, greyed out. *Examples* are found in Table A.3.

Instructions: Please correct only the **BOLD, UNDERLINED** sentence. The gray sentence is given as a context.

A.6.3 Correct Sentences with Unspecified Edits

Instructions

Please correct the following sentence to make it sound natural and fluent to a native speaker of English. The sentence is written by a second language learner of English. You should fix grammatical mistakes, awkward phrases, etc. following standard written usage con-

Original Sentence	**Good Edits**	**Bad Edits**
John is fund for cats.	John is fond of cats. *Fixed spelling and preposition errors.*	John is crazy about cats. *Changes the meaning for no good reason.*
If there is willing wish, Sanskrit can be made common studied again. *Fixed only the obvious grammatical errors; still a little strangely-worded, but grammatical (which is all we want).*	If there is a willing wish, Sanskrit can be made commonly studied again. *Too much rewriting; more than just grammatical issues were corrected.*	If there is a will to do so, Sanskrit can once again be commonly studied.
At center is Ganges Plains, this has the most fertile earth.	At the center is the Ganges Plains; this has the most fertile earth. *Only corrected grammar.*	At the center is the Ganges Plains, which has the most fertile soil in the world. *Wording choice change.*

Table A.2: Example minimal-edit corrections for the task described in Appendix A.6.1.

Original Sentence	**Good Rewrites**	**Bad Rewrites**
John is fund for cats.	John is fond of cats. *Fixed spelling and preposition errors.*	John is crazy about cats. *Changes the meaning for no good reason.*
If there is willing wish, Sanskrit can be made common studied again.	If there is a will to do so, Sanskrit can once again be commonly studied. *Rewritten to sound more natural to a native speaker.*	If there is a willing wish, Sanskrit can be made commonly studied again. *Fixed only the obvious grammatical errors; the sentence is grammatical but still a bit strangely-worded.*
This will, if not already, caused problems as there are very limited spaces for us.	If it hasn't already, this will cause problems, since there is limited space for us. *Many changes were made to produce a more natural-sounding sentence.*	This will, if not already, cause problems, as there are very limited spaces for us. *Sentence is grammatical but poorly written.*

Table A.3: Example fluency corrections for the task described in Appendix A.6.2.

ventions, but your edits must be conservative. Please keep the original sentence (words, phrases, and structure) as much as possible. The ultimate goal of this task is to make the given sentence sound natural to native speakers of English without making unnecessary changes. Please do not split the original sentence into two or more. Edits are not required when the sentence is already grammatical and sounds natural. When the sentence is too erroneous to correct (e.g. "The memory is very.", "For example, in children."), please check the box below the sentence.

Examples are found in Table A.4.

Please keep in mind the instructions.

- Correct the following sentence to make it sound natural and fluent to a native speaker of English.
- Your edits must be conservative. Please keep the original sentence (words, phrases, and structure) as much as possible.
- Please do not split the sentence into two or more.
- Edits are not required when the sentence is already grammatical and sounds natural.

A.7 Ranking Sentences by Grammaticality

In this HIT, we had participants rank the grammaticality of five versions of the same sentence. The versions were randomly selected from the input sentence, a human-corrected reference, and the 13 system outputs from the CoNLL-2014 shared task participants.

	Sentence	**Notes**
Original	John is fund for cats.	
Good edits	John is fond of cats.	Fixed spelling and preposition errors.
Bad edits	John is crazy about cats.	Added emphasis and changes the meaning for no reason.
		Notes
Original	From this scope, social media has shorten our distance.	
Good edits	From this perspective, social media has shortened the distance between us.	The edits are grammatical and fluent. Also, the original words and phrases are kept as much as possible.
Bad edits	From this scope, social media has shortened our distance.	The edits are grammatical but the sentence doesn't sound natural to a native speaker.
		Notes
Original	It is obvious to see that internet saves people's time and also connect people globally.	
Good edits	It is obvious to see that the internet saves people's time and also connects people globally.	Missing article (*the*) and subject-verb agreement (*verbs*) are corrected.
Bad edits	It's clear the Internet's greatest appeal lies partially in its time-saving ability but is mostly attributable to the ease with which it brings users closer together.	The edits are grammatical and fluent but the original words and phrases are not preserved at all.

Table A.4: Example fluency corrections for the task described in Appendix A.6.3.

A.7.1 Grammaticality Ranking Challenge

You will be first presented with two sentences from a passage: context and original. The second one, original, may contain grammatical errors. The next group of sentences are possible grammatical corrections to that original sentence.

Please rank that group of sentences by their grammaticality. The rank is between 1 (the best) and 5 (the worst), and ties are allowed. Specifically, you should assign the best sentence(s) as Rank 1, the next best sentence(s) as Rank 2, and so on. For reference, a previous sentence is given as context. Also, the buttons (show/hide mark-up) will show/hide all the edits from the original sentence via highlighting or strike-throughs.

Instructions:

Please select the rank between 1 (the best) and 5 (the worst). Ties are allowed. Sometimes you are asked to compare less than 5 sentences with a filler "###", and you don't have to rank the fillers. For reference, a previous sentence is given as context. Also, the buttons (show/hide mark-up) will show/hide all the edits from the original sentence by highlighting or crossing through.

Best ← Rank 1 — Rank 2 — Rank 3 — Rank 4 — Rank 5 → Worst

Appendix B

Language packs for T2T

We present a simple, prepackaged solution for generating paraphrases of English sentences. We use the Paraphrase Database (PPDB) for monolingual sentence rewriting and provide machine translation *language packs:* prepackaged, tuned models that can be downloaded and used to generate paraphrases on a standard Unix environment. The language packs can be treated as a black box or customized to specific tasks. In this demonstration, we will explain how to use the included interactive web-based tool to generate sentential paraphrases.[1]

[1] Originally appeared as a 2017 NAACL System Demonstrated, "Sentential Paraphrasing as Black-Box Machine Translation" (Napoles, Callison-Burch, and Post, 2016).

B.1 Introduction

Monolingual sentence rewriting encompasses a variety of tasks for which the goal is to generate an output sentence with similar meaning to an input sentence, in the same language. The generated sentences can be called *sentential paraphrases*. Some tasks that generate sentential paraphrases include sentence simplification, compression, grammatical error correction, or expanding multiple reference sets for machine translation. For researchers not focused on these tasks, it can be difficult to develop a one-off system due to resource requirements.

To address this need, we are releasing a black box for generating sentential paraphrases: machine translation language packs. The language packs consist of prepackaged models for the Joshua 6 decoder (Post et al., 2015) and a monolingual "translation" grammar derived from the Paraphrase Database (PPDB) 2.0 (Pavlick et al., 2015). The PPDB provides tremendous coverage over English text, containing more than 200 million paraphrases extracted from 100 million sentences (Ganitkevitch and Callison-Burch, 2014). For the first time, any researcher with Java 7 and Unix (there are no other dependencies) can generate sentential paraphrases without developing their own system. Additionally, the language packs include a web tool for interactively paraphrasing sentences and adjusting the parameters.

The language packs contain everything needed to generate sentential paraphrases in English:

- a monolingual synchronous grammar,
- a language model,
- a ready-to-use configuration file,
- the Joshua 6 runtime, so that no compilation is necessary,
- a shell script to invoke the Joshua decoder, and
- a web tool for interactive decoding and parameter configuration.

The system is invoked by a single command, either on a batch of sentences or as an interactive server.

Users can choose which size grammar to include in the language pack, corresponding to the PPDB pack sizes (S through XXXL).

In the following sections, we will describe the translation model and grammar, provide examples of output, and explain how the configuration can be adjusted for specific needs.

B.2 Language Pack Description

Several different size language packs are available for download.[2] The components of the language packs are described below.

[2] `http://joshua-decoder.com/language-packs/paraphrase/`

APPENDIX B. LANGUAGE PACKS FOR T2T

1. Grammar

Our approach to sentential paraphrasing is analogous to machine translation. As a translation grammar, we use PPDB 2.0, which contains 170-million lexical, phrasal, and syntactic paraphrases (Pavlick et al., 2015). Each language pack contains a PPDB grammar that has been packed into a binary form for faster computation (Ganitkevitch et al., 2012), and users can select which size grammar to use. The rules present in each grammar are determined by the PPDB 2.0 score, which indicates the paraphrase quality (as given by a supervised regression model) and correlates strongly with human judgments of paraphrase appropriateness (Pavlick et al., 2015). Grammars of different sizes are created by changing the paraphrase score thresholds; larger grammars therefore contain a wider diversity of paraphrases, but with lower confidences.

2. Features

Each paraphrase in PPDB 2.0 contains 44 features, described in Ganitkevitch and Callison-Burch (2014) and Pavlick et al. (2015). For each paraphrase pair, we call the input the *original* and the new phrase the *candidate*. Features can reflect just the candidate phrase or a relationship between the original and candidate phrases. Each of these features is assigned a weight, which guides the decoder's choice of paraphrases to apply to generate the final candidate sentence. All feature values are pre-calculated in PPDB 2.0.

3. Decoding

The language packs include a compiled Joshua runtime for decoding, a script to invoke it, and configuration files for different tuned models. There is also a web-based tool for interactively querying a server version of the decoder for paraphrases. We include a 5-gram Gigaword v.5 language model for decoding. One or more language-model scores are used to rank translation candidates during decoding. The decoder outputs the *n*-best candidate paraphrases, ranked by model score.

B.3 Models

Each language pack has three pre-configured models to use either out of the box or as a starting point for further customization. There are tuned models for (1) sentence compression, (2) text simplification, and (3) a general-purpose model with hand-tuned weights. These models are distinguished only by the different weight vectors, and are selected by point the Joshua invocation script to the corresponding configuration file.

B.3.1 Tuned Models

We include two models that were tuned for (1) sentence compression and (2) simplification. The compression model is based on the work of Ganitkevitch et al. (2011), and uses the same features, tuning data, and objective function, PRÉCIS. The simplification model is described in Xu et al. (2016), and is optimized to the SARI metric. The system was tuned using the parallel data described therein as well as the Newsela corpus (Xu et al., 2015). There is no specialized grammar for these models; instead, the parameters were tuned to choose appropriate paraphrases from the PPDB.

Sample output generated with these models is shown in Table B.1.

B.3.2 Hand-Derived Weights

To configure the general-purpose model, which generates paraphrases for no specific task, we examined the output of 100 sentences randomly selected from each of three different domains: newswire (WSJ 0–1 (Marcus et al., 1993)), “simple” English (the Britannica Elementary corpus (Barzilay and Elhadad, 2003)), and general text (the WaCky corpus (Baroni et al., 2009)). We systematically varied the weights of the Gigaword LM and the PPDB 2.0 score features and selected values that yielded the best output as judged by the authors. The parameters selected for the generic language packs are $\textit{weight}_{\texttt{lm}} = 10$ and $\textit{weight}_{\texttt{ppdb2}} = 15$, with all other weights are set to zero. Example output is shown in Table B.1.

B.4 User Customization

The language packs include configuration files with pre-determined weights that can be used on their own or as a jumping-off point for custom configurations. There are weights for each of the 44 PPDB 2.0 features as well as for the language model(s) used by the

	Compression
Orig:	rice admits mistakes have been made by american administration in rebuilding iraq
Gen:	rice admits mistakes were made by american administration in rebuilding iraq
Orig:	partisanship is regarded as a crime , and pluralism is rejected , and no one in the shura council would seek to compete with the ruler or distort his image .
Gen:	partisanship is regarded as a crime □ and pluralism is rejected □ and none in the shura council would seek to compete with the ruler or distort his image .
	Simplification
Orig:	fives is a british sport believed to derive from the same origins as many racquet sports .
Gen:	fives is a british sport thought to come from the same source as many racquet sports .
Orig:	in the soviet years , the bolsheviks demolished two of rostov 's principal landmarks — st alexander nevsky cathedral (1908) and st george cathedral in nakhichevan (1783-1807) .
Gen:	in the soviet years , the bolsheviks destroyed two of rostov 's key landmarks — st alexander nevsky church (1908) and st george church in naxçivan (1783-1807) .
	Generic
Orig:	because the spaniards had better weapons , cortes and his army took over tenochtitlan by 1521 .
Gen:	as the spaniards had better weapons , cortes and his men took over tenochtitlan by 1521 .
Orig:	it was eventually abandoned due to resistance from the population .
Gen:	it was later abandoned due to opposition from the population .

Table B.1: Sample output from the three models. Underlines designate changed spans, and □ indicates deletions.

decoder. We encourage researchers to explore modifications to the model to suit their specific tasks, and we have clearly identified five aspects of the language packs that can be modified:

1. Alternate language models. The decoder can accept multiple LMs, and the packs include LMs estimated over newswire text and "simple" English. Other user-provided LMs can be used for tasks targeting different domains of text.

2. Rank output with a custom metric. The n-best candidate sentences are chosen by their score according to a given metric (LM score for the generic model, and PRÉCIS and SARI for the tuned models), however other metrics can be used instead.

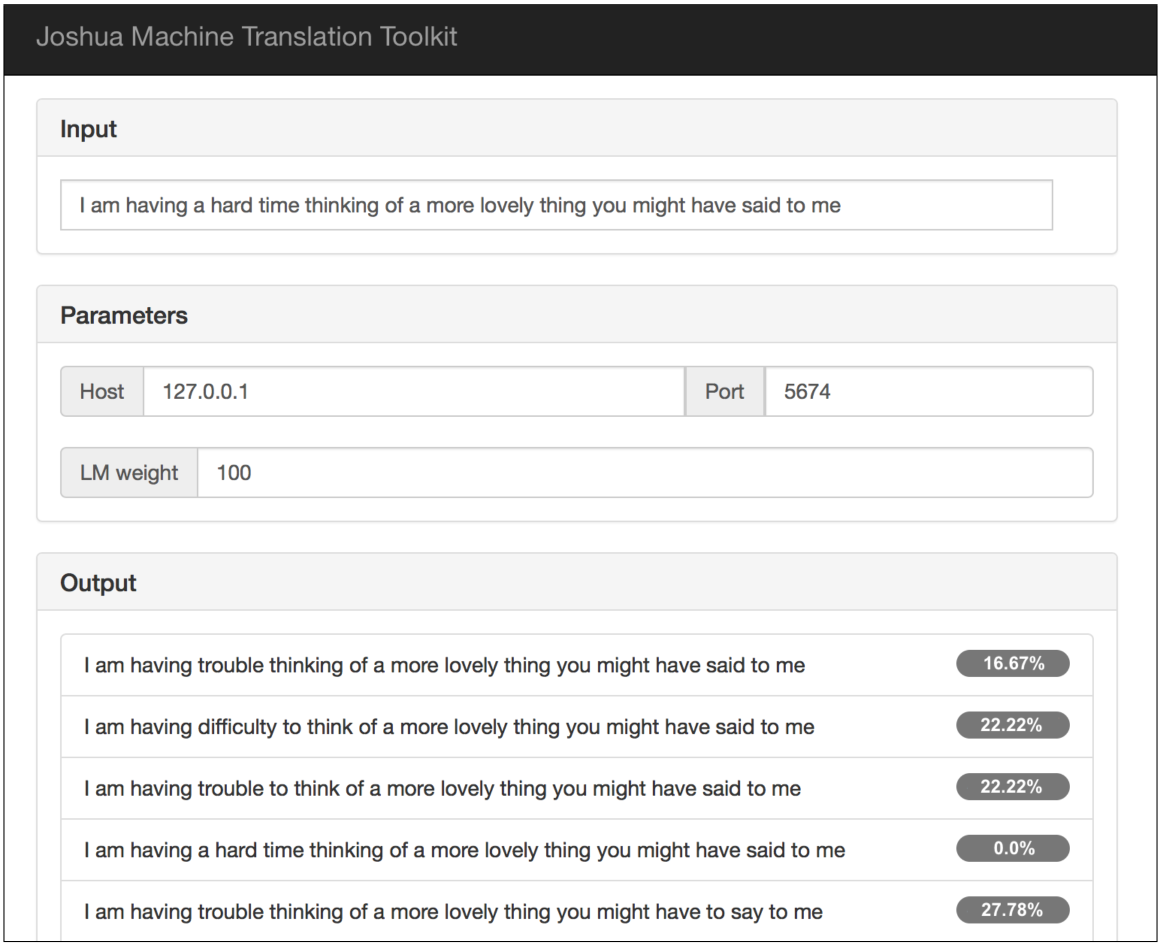
3. Manually adjust parameters. The weights of the features discussed in Section B.3 can be adjusted, as well as other PPDB feature weights. The web tool (Figure B.1) allows users to select the weights for all of the features and see the top-5 candidates generated with those weights. Some of the more interpretable features to target include the following:

- Language model perplexity of the candidate
- Length difference between the phrase pair
- Paraphrase score
- Formality of the candidate
- Complexity of the candidate
- Entailment relations between the original phrase and the candidate paraphrase

4. Optimize parameters with parallel data. For tailoring machine translation to a specific task, the weights given to each feature can be optimized to a given metric over a tuning set of parallel data. This metric is commonly BLEU in machine translation, but it can be a custom metric for a specific task, such as PRÉCIS for compression (Ganitkevitch et al., 2011) or SARI for simplification (Xu et al., 2016). The user needs to provide a parallel dataset for tuning, ideally with about 2,000 thousand sentences. The pipeline scripts in the Joshua decoder have options for optimization, with the user specifying the language pack grammar and parallel tuning data. The configuration file included in the language pack can be used as a template for tuning.

B.5 Interactive Tool

Finally, we include a web tool that lets users interact with the decoder and choose custom weights (Figure B.1). Once users have downloaded the toolkit, an included script lets them run the decoder as a server, and through the web interface they can type individual sentences and adjust model parameters. The interface includes an input text box (one sentence at a time), and slider bars to change the weights of any of the features used for decoding. Since this model has not been manually evaluated, we favor precision over recall and maintain a relatively conservative level of paraphrasing. The user is shown the top 10 outputs, as ranked by the sentence score. For each output sentence, we report the Translation Edit Rate (TER), which is the number of changes needed to transform the output sentence into the input (Snover et al., 2006).



Input

I am having a hard time thinking of a more lovely thing you might have said to me

Parameters

Host 127.0.0.1

Port 5674

LM weight 100

Output

I am having trouble thinking of a more lovely thing you might have said to me	16.67%
I am having difficulty to think of a more lovely thing you might have said to me	22.22%
I am having trouble to think of a more lovely thing you might have said to me	22.22%
I am having a hard time thinking of a more lovely thing you might have said to me	0.0%
I am having trouble thinking of a more lovely thing you might have to say to me	27.78%

Figure B.1: A screen shot of the web tool. The number to the right of each output sentence is the TER.

This tool can be used to demonstrate and test a model or to hand-tune the model in order to determine the parameters for a configuration file to paraphrase a large batch of sentences. Detailed instructions for using the tool and shell scripts, as well as a detailed description of the configuration file, are available at the language pack homepage: `http://joshua-decoder.com/language-packs/paraphrase/`.

B.6 Related Work

Previous work has applied machine translation techniques to monolingual sentence rewriting tasks. The most closely related works used a monolingual paraphrase grammar for sentence compression (Ganitkevitch et al., 2011) and sentence simplification (Xu et al., 2016), both of which developed custom metrics and task-specific features. Various other MT approaches have been used for generating sentence simplifications, however none of these used a general-purpose paraphrase grammar (e.g., Narayan and Gardent, 2014; Wubben et al., 2012). Another application of sentential paraphrases is to expand multiple reference sets for machine translation (Madnani and Dorr, 2010).

PPDB has been used for many tasks, including recognizing textual entailment, question generation, and measuring semantic similarity.

These language packs were inspired by the foreign language packs released with Joshua 6 (Post et al., 2015)

.

B.7 Conclusion

We have presented a black box for generating sentential paraphrases: PPDB language packs. The language packs include everything necessary for generation so that they can be downloaded and invoked with a single command. This toolkit can be used for a variety of tasks: as a helpful tool for writing (what is another way to express a sentence?); generating additional training or tuning data, such as multiple-references for machine translation or other text-to-text rewriting tasks; or for changing the style or tone of a text. We hope their ease-of-use will facilitate future work on text-to-text rewriting tasks.

References

Yasuhiro Akiba, Kenji Imamura, and Eiichiro Sumita (2001). “Using multiple edit distances to automatically rank machine translation output”. In *Proceedings of the Machine Translation Summit VIII*. Santiago de Compostela, Spain, pp. 15–20.

Joshua Albrecht and Rebecca Hwa (2007). “A Re-examination of Machine Learning Approaches for Sentence-Level MT Evaluation”. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*. Prague, Czech Republic: Association for Computational Linguistics, pp. 880–887.

Sandra M. Aluísio, Lucia Specia, Thiago A.S. Pardo, Erick G. Maziero, and Renata P.M. Fortes (2008). “Towards Brazilian Portuguese Automatic Text Simplification Systems”. In *Proceedings of the Eighth ACM Symposium on Document Engineering*. DocEng ’08. Sao Paulo, Brazil: ACM, pp. 240–248.

Voice Of America (2009). *Word Book, 2009 Edition*. Accessed: February 1, 2010. URL: `www.voaspecialenglish.com`.

Eleftherios Avramidis, Maja Popović, David Vilar, and Aljoscha Burchardt (2011). “Evaluate with Confidence Estimation: Machine ranking of translation outputs using grammatical features”. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*. Edinburgh, Scotland, pp. 65–70.

Nguyen Bach, Qin Gao, Stephan Vogel, and Alex Waibel (2011). “TriS: A Statistical Sentence Simplifier with Log-linear Models and Margin-based Discriminative Training”. In *Proceedings of 5th International Joint Conference on Natural Language Processing*. Chiang Mai, Thailand: Asian Federation of Natural Language Processing, pp. 474–482.

Srinivas Bangalore, Owen Rambow, and Steve Whittaker (2000). “Evaluation metrics for generation”. In *Proceedings of the First International Conference on Natural Language Generation—Volume 14*. Association for Computational Linguistics, pp. 1–8.

Michele Banko, Vibhu O. Mittal, and Michael J. Witbrock (2000). “Headline Generation Based on Statistical Translation”. In *Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics*. Hong Kong: Association for Computational Linguistics, pp. 318–325.

Colin Bannard and Chris Callison-Burch (2005). “Paraphrasing with Bilingual Parallel Corpora”. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL’05)*. Ann Arbor, Michigan: Association for Computational Linguistics, pp. 597–604.

Marco Baroni, Silvia Bernardini, Adriano Ferraresi, and Eros Zanchetta (2009). “The WaCky wide web: a collection of very large linguistically processed web-crawled corpora”. In *Language resources and evaluation* 43.3, pp. 209–226.

REFERENCES

Regina Barzilay and Noemie Elhadad (2003). “Sentence Alignment for Monolingual Comparable Corpora”. In *Proceedings of the 2003 Conference on Empirical Methods in Natural Language Processing*. Ed. by Michael Collins and Mark Steedman, pp. 25–32.

Regina Barzilay and Lillian Lee (2003). “Learning to Paraphrase: An Unsupervised Approach using Multiple-Sequence Alignment”. In *Proceedings of the Human Language Technology Conference of the North American chapter of the Association for Computational Linguistics (HLT/NAACL-2003)*. Edmonton, Alberta, pp. 16–23.

Anja Belz and Adam Kilgarriff (2006). “Shared-task evaluations in HLT: Lessons for NLG”. In *Proceedings of the Fourth International Natural Language Generation Conference*. Sydney, Australia: Association for Computational Linguistics, pp. 133–135.

Naftali Bendavid and Janet Hook (2011). “Last-Minute Deal Averts Shutdown”. In *The Wall Street Journal*. Accessed: April 9, 2011. URL: `http://www.wsj.com/articles/SB10001424052748704503104576250541381308346`.

Adam L. Berger, Vincent J. Della Pietra, and Stephen A. Della Pietra (1996). “A maximum entropy approach to natural language processing”. In *Computational Linguistics* 22.1, pp. 39–71.

Fadi Biadsy, Julia Hirschberg, and Elena Filatova (2008). “An unsupervised approach to biography production using Wikipedia”. In *Proceedings of ACL-08: HLT*. Columbus, Ohio: Association for Computational Linguistics, pp. 807–815.

Steven Bird and Edward Loper (2004). “NLTK: The Natural Language Toolkit”. In *The Companion Volume to the Proceedings of 42nd Annual Meeting of the Association for*

Computational Linguistics. Barcelona, Spain: Association for Computational Linguistics, pp. 214–217.

Steven Bird, Ewan Klein, and Edward Loper (2009). *Natural language processing with Python: Analyzing text with the Natural Language Toolkit*. Sebastopol, CA: O'Reilly Media, Inc.

Ondřej Bojar, Christian Buck, Christian Federmann, Barry Haddow, Philipp Koehn, Johannes Leveling, Christof Monz, Pavel Pecina, Matt Post, Herve Saint-Amand, Radu Soricut, Lucia Specia, and Aleš Tamchyna (2014). "Findings of the 2014 Workshop on Statistical Machine Translation". In *Proceedings of the Ninth Workshop on Statistical Machine Translation*. Baltimore, Maryland, USA: Association for Computational Linguistics, pp. 12–58.

Ondřej Bojar, Rajen Chatterjee, Christian Federmann, Barry Haddow, Matthias Huck, Chris Hokamp, Philipp Koehn, Varvara Logacheva, Christof Monz, Matteo Negri, Matt Post, Carolina Scarton, Lucia Specia, and Marco Turchi (2015). "Findings of the 2015 Workshop on Statistical Machine Translation". In *Proceedings of the Tenth Workshop on Statistical Machine Translation*. Lisbon, Portugal: Association for Computational Linguistics, pp. 1–46.

Ondřej Bojar, Rajen Chatterjee, Christian Federmann, Yvette Graham, Barry Haddow, Matthias Huck, Antonio Jimeno Yepes, Philipp Koehn, Varvara Logacheva, Christof Monz, Matteo Negri, Aurelie Neveol, Mariana Neves, Martin Popel, Matt Post, Raphael Rubino, Carolina Scarton, Lucia Specia, Marco Turchi, Karin Verspoor, and Marcos

Zampieri (2016). “Findings of the 2016 Conference on Machine Translation”. In *Proceedings of the First Conference on Machine Translation*. Berlin, Germany: Association for Computational Linguistics, pp. 131–198.

Ted Briscoe (2006). *An introduction to tag sequence grammars and the RASP system parser*. Tech. rep. University of Cambridge Computer Library.

Christopher Bryant and Hwee Tou Ng (2015). “How Far are We from Fully Automatic High Quality Grammatical Error Correction?” In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Beijing, China: Association for Computational Linguistics, pp. 697–707.

Christopher Bryant, Mariano Felice, and Ted Briscoe (2017). “Automatic Annotation and Evaluation of Error Types for Grammatical Error Correction”. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Vancouver, Canada: Association for Computational Linguistics, pp. 793–805.

Chris Callison-Burch and Raymond S. Flournoy (2001). “A Program for Automatically Selecting the Best Output from Multiple Machine Translation Engines”. In *Proceedings of the Machine Translation Summit VIII*. Vol. 318.

Chris Callison-Burch, Miles Osborne, and Philipp Koehn (2006). “Re-evaluating the role of BLEU in machine translation research”. In *11th Conference of the European Chapter of the Association for Computational Linguistics*. Trento, Italy: Association for Computational Linguistics, pp. 249–256.

REFERENCES

Yvonne Canning, John Tait, Jackie Archibald, and Ros Crawley (2000). “Cohesive generation of syntactically simplified newspaper text”. In *Proceedings of the Third International Workshop on Text, Speech and Dialogue (TSD)*, pp. 145–150.

John Carroll, Guido Minnen, Darren Pearce, Yvonne Canning, Siobhan Devlin, and John Tait (1999). “Simplifying text for language-impaired readers”. In *Proceedings of the 14th Conference of the 9th European Conference for Computational Linguistics (EACL)*. Bergen, Norway: Association for Computational Linguistics.

Jieun Chae and Ani Nenkova (2009). “Predicting the fluency of text with shallow structural features”. In *Proceedings of the 12th Conference of the European Chapter of the ACL (EACL 2009)*. Athens, Greece: Association for Computational Linguistics, pp. 139–147.

Tsz Ping Chan, Chris Callison-Burch, and Benjamin Van Durme (2011). “Reranking Bilingually Extracted Paraphrases Using Monolingual Distributional Similarity”. In *Proceedings of the GEMS 2011 Workshop on GEometrical Models of Natural Language Semantics*. Edinburgh, UK: Association for Computational Linguistics, pp. 33–42.

Eugene Charniak and Mark Johnson (2005). “Coarse-to-fine n-best parsing and MaxEnt discriminative reranking”. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*. Ann Arbor, Michigan: Association for Computational Linguistics, pp. 173–180.

REFERENCES

Martin Chodorow and Claudia Leacock (2000). "An Unsupervised Method for Detecting Grammatical Errors". In *1st Meeting of the North American Chapter of the Association for Computational Linguistics*, pp. 140–147.

Martin Chodorow, Markus Dickinson, Ross Israel, and Joel Tetreault (2012). "Problems in Evaluating Grammatical Error Detection Systems". In *Proceedings of COLING 2012*. Mumbai, India, pp. 611–628.

Shamil Chollampatt, Duc Tam Hoang, and Hwee Tou Ng (2016). "Adapting Grammatical Error Correction Based on the Native Language of Writers with Neural Network Joint Models". In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. Austin, Texas: Association for Computational Linguistics, pp. 1901–1911.

Noam Chomsky (1957). *Syntactic Structures*. The Hague: Mouton Publishers.

James Clarke and Mirella Lapata (2006). "Models for sentence compression: A comparison across domains, training requirements and evaluation measures". In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, pp. 377–384.

James Clarke and Mirella Lapata (2007). "Modelling compression with discourse constraints". In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pp. 1–11.

REFERENCES

James Clarke and Mirella Lapata (2008). “Global Inference for Sentence Compression: An Integer Linear Programming Approach”. In *Journal of Artificial Intelligence Research* 31, pp. 399–429.

Trevor Cohn and Mirella Lapata (2007). “Large Margin Synchronous Generation and its Application to Sentence Compression”. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*. Prague, Czech Republic, pp. 73–82.

Trevor Cohn and Mirella Lapata (2008). “Sentence Compression Beyond Word Deletion”. In *Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008)*. Manchester, UK: Coling 2008 Organizing Committee, pp. 137–144.

Trevor Cohn and Mirella Lapata (2009). “Sentence Compression as Tree Transduction”. In *Journal of Artificial Intelligence Research (JAIR)* 34, pp. 637–674.

Chloe Coleman (2017). “Finding beauty in function”. In *The Washington Post*. Accessed: June 14, 2017. URL: `https://www.washingtonpost.com/news/in-sight/wp/2017/06/14/finding-beauty-in-function-the-work-of-charles-sheeler/`.

Simon Corston-Oliver (2001). “Text compaction for display on very small screens”. In *Proceedings of the NAACL Workshop on Automatic Summarization*.

Will Coster and David Kauchak (2011a). “Learning to Simplify Sentences Using Wikipedia”. In *Proceedings of the Workshop on Monolingual Text-To-Text Generation*. Portland, Oregon: Association for Computational Linguistics, pp. 1–9.

William Coster and David Kauchak (2011b). “Simple English Wikipedia: A new text simplification task”. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*. Portland, Oregon, pp. 665–669.

Koby Crammer, Ofer Dekel, Joseph Keshet, Shai Shalev-Shwartz, and Yoram Singer (2006). “Online Passive-Aggressive Algorithms”. In *Journal of Machine Learning Research (JMLR)*.

Walter Daelemans, Anja Höthker, and Erik Tjong Kim Sang (2004). “Automatic Sentence Simplification for Subtitling in Dutch and English”. In *Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC’04)*. Lisbon, Portugal: European Language Resources Association (ELRA).

Daniel Dahlmeier and Hwee Tou Ng (2012). “Better Evaluation for Grammatical Error Correction”. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Montréal, Canada: Association for Computational Linguistics, pp. 568–572.

Daniel Dahlmeier, Hwee Tou Ng, and Siew Mei Wu (2013). “Building a Large Annotated Corpus of Learner English: The NUS Corpus of Learner English”. In *Proceedings of the Eighth Workshop on Innovative Use of NLP for Building Educational Applications*. Atlanta, Georgia: Association for Computational Linguistics, pp. 22–31.

Robert Dale and Adam Kilgarriff (2011). “Helping Our Own: The HOO 2011 Pilot Shared Task”. In *Proceedings of the Generation Challenges Session at the 13th European*

Workshop on Natural Language Generation. Nancy, France: Association for Computational Linguistics, pp. 242–249.

Robert Dale, Ilya Anisimoff, and George Narroway (2012). “HOO 2012: A Report on the Preposition and Determiner Error Correction Shared Task”. In *Proceedings of the Seventh Workshop on Building Educational Applications Using NLP*. Montréal, Canada: Association for Computational Linguistics, pp. 54–62.

Vidas Daudaravicius, Rafael E. Banchs, Elena Volodina, and Courtney Napoles (2016). “A Report on the Automatic Evaluation of Scientific Writing Shared Task”. In *Proceedings of the 11th Workshop on Innovative Use of NLP for Building Educational Applications*. San Diego, CA: Association for Computational Linguistics, pp. 53–62.

Hal Daumé III and Daniel Marcu (2002). “A noisy-channel model for document compression”. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*. Association for Computational Linguistics, pp. 449–456.

Rachele De Felice and Stephen G. Pulman (2008). “A Classifier-Based Approach to Preposition and Determiner Error Correction in L2 English”. In *Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008)*. Manchester, UK: Coling 2008 Organizing Committee, pp. 169–176.

Marie-Catherine De Marneffe, Timothy Dozat, Natalia Silveira, Katri Haverinen, Filip Ginter, Joakim Nivre, and Christopher D Manning (2014). “Universal Stanford dependencies: A cross-linguistic typology.” In *Conference on Language Resources and Evaluation (LREC)*. Vol. 14, pp. 4585–92.

Ludovic Denoyer and Patrick Gallinari (2007). “The Wikipedia XML Corpus”. In *Comparative Evaluation of XML Information Retrieval Systems*, pp. 12–19.

Siobhan Devlin and Gary Unthank (2006). “Helping Aphasic People Process Online Information”. In *Proceedings of the 8th International ACM SIGACCESS Conference on Computers and Accessibility*. Assets ’06. Portland, Oregon, USA: ACM, pp. 225–226.

George Doddington (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*. HLT ’02. San Diego, California: Morgan Kaufmann Publishers Inc., pp. 138–145.

Bonnie Dorr, David Zajic, and Richard Schwartz (2003). “Hedge trimmer: A parse-and-trim approach to headline generation”. In *Proceedings of the HLT-NAACL Workshop on Text Summarization Workshop*.

Mark Dredze, Koby Crammer, and Fernando Pereira (2008). “Confidence-weighted Linear Classification”. In *Proceedings of the 25th International Conference on Machine Learning*. ICML ’08. Helsinki, Finland: ACM, pp. 264–271.

Chris Dyer, Victor Chahuneau, and Noah A. Smith (2013). “A Simple, Fast, and Effective Reparameterization of IBM Model 2”. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Atlanta, Georgia: Association for Computational Linguistics, pp. 644–648.

REFERENCES

Jens Eeg-Olofsson and Ola Knutsson (2003). “Automatic grammar checking for second language learners—the use of prepositions”. In *Proceedings of NoDaLida 2003*.

Desmond Elliott and Frank Keller (2013). “Image Description using Visual Dependency Representations”. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*. Seattle, Washington, USA: Association for Computational Linguistics, pp. 1292–1302.

James D. Evans (1996). *Straightforward Statistics for the Behavioral Sciences*. Pacific Grove, California: Brooks/Cole Publishing Company.

Mariano Felice and Ted Briscoe (2015). “Towards a standard evaluation method for grammatical error detection and correction”. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics*. Denver, Colorado: Association for Computational Linguistics, pp. 578–587.

Mariano Felice, Zheng Yuan, Øistein E. Andersen, Helen Yannakoudakis, and Ekaterina Kochmar (2014). “Grammatical error correction using hybrid systems and type filtering”. In *Proceedings of the Eighteenth Conference on Computational Natural Language Learning: Shared Task*. Baltimore, Maryland: Association for Computational Linguistics, pp. 15–24.

Lijun Feng, Noémie Elhadad, and Matt Huenerfauth (2009). “Cognitively Motivated Features for Readability Assessment”. In *Proceedings of the 12th Conference of the European Chapter of the ACL (EACL 2009)*. Athens, Greece: Association for Computational Linguistics, pp. 229–237.

REFERENCES

Katja Filippova and Michael Strube (2008). “Dependency Tree Based Sentence Compression”. In *Proceedings of the Fifth International Natural Language Generation Conference*. Salt Fork, Ohio, USA: Association for Computational Linguistics, pp. 25–32.

Katja Filippova, Enrique Alfonseca, Carlos A. Colmenares, Lukasz Kaiser, and Oriol Vinyals (2015). “Sentence Compression by Deletion with LSTMs”. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. Lisbon, Portugal: Association for Computational Linguistics, pp. 360–368.

Andrew M. Finch, Yasuhiro Akiba, and Eiichiro Sumita (2004). “How Does Automatic Machine Translation Evaluation Correlate with Human Scoring as the Number of Reference Translations Increases?” In *Proceedings of the Fourth International Conference on Language Resources and Evaluation, LREC*. Lisbon, Portugal.

Dimitrios Galanis and Ion Androutsopoulos (2010). “An extractive supervised two-stage method for sentence compression”. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*. Los Angeles, California: Association for Computational Linguistics, pp. 885–893.

Michel Galley and Kathleen McKeown (2007). “Lexicalized Markov Grammars for Sentence Compression”. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Proceedings of the Main Conference*. Rochester, New York: Association for Computational Linguistics, pp. 180–187.

Michael Gamon, Jianfeng Gao, Chris Brockett, Alex Klementiev, William B. Dolan, Dmitriy Belenko, and Lucy Vanderwende (2008). “Using Contextual Speller Techniques and Language Modeling for ESL Error Correction”. In *Proceedings of IJCNLP*. Hyderabad, India.

Juri Ganitkevitch and Chris Callison-Burch (2014). “The Multilingual Paraphrase Database”. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC-2014)*. Reykjavik, Iceland: European Language Resources Association (ELRA).

Juri Ganitkevitch, Chris Callison-Burch, Courtney Napoles, and Benjamin Van Durme (2011). “Learning Sentential Paraphrases from Bilingual Parallel Corpora for Text-to-Text Generation”. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*. Edinburgh, Scotland, UK.: Association for Computational Linguistics, pp. 1168–1179.

Juri Ganitkevitch, Yuan Cao, Jonathan Weese, Matt Post, and Chris Callison-Burch (2012). “Joshua 4.0: Packing, PRO, and Paraphrases”. In *Proceedings of the Seventh Workshop on Statistical Machine Translation*. Montréal, Canada: Association for Computational Linguistics, pp. 283–291.

Zeno Gantner and Lars Schmidt-Thieme (2009). “Automatic Content-Based Categorization of Wikipedia Articles”. In *Proceedings of the 2009 Workshop on The People’s Web Meets NLP: Collaboratively Constructed Semantic Resources (People’s Web)*. Suntec, Singapore: Association for Computational Linguistics, pp. 32–37.

Oren Glickman, Ido Dagan, Mikaela Keller, Samy Bengio, and Walter Daelemans (2006). “Investigating lexical substitution scoring for subtitle generation”. In *Proceedings of the Tenth Conference on Computational Natural Language Learning*. Association for Computational Linguistics, pp. 45–52.

Yvette Graham, Timothy Baldwin, Alistair Moffat, and Justin Zobel (2017). “Can machine translation systems be evaluated by the crowd alone”. In *Natural Language Engineering* 23.1, pp. 3–30.

Roman Grundkiewicz and Marcin Junczys-Dowmunt (2014). “The WikEd Error Corpus: A Corpus of Corrective Wikipedia Edits and its Application to Grammatical Error Correction”. In *Advances in Natural Language Processing – Lecture Notes in Computer Science*. Ed. by Adam Przepiórkowski and Maciej Ogrodniczuk. Vol. 8686. Springer, pp. 478–490.

Roman Grundkiewicz, Marcin Junczys-Dowmunt, and Edward Gillian (2015). “Human Evaluation of Grammatical Error Correction Systems”. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. Lisbon, Portugal: Association for Computational Linguistics, pp. 461–470.

Xianpei Han, Le Sun, and Jun Zhao (2011). “Collective Entity Linking in Web Text: A Graph-based Method”. In *Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval*. SIGIR ’11. Beijing, China: ACM, pp. 765–774.

REFERENCES

Donna Harman and Mark Liberman (1993). *TIPSTER Complete*. Linguistic Data Consortium. DVD. Philadelphia.

George E. Heidorn, Karen Jensen, Lance A. Miller, Roy J. Byrd, and Martin S Chodorow (1982). “The EPISTLE text-critiquing system”. In *IBM Systems Journal* 21.3, pp. 305–326.

Michael Heilman, Aoife Cahill, Nitin Madnani, Melissa Lopez, Matthew Mulholland, and Joel Tetreault (2014). “Predicting Grammaticality on an Ordinal Scale”. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Baltimore, Maryland: Association for Computational Linguistics, pp. 174–180.

Ralf Herbrich, Tom Minka, and Thore Graepel (2006). “TrueSkill$^{\text{TM}}$: A Bayesian Skill Rating System”. In *Proceedings of the Twentieth Annual Conference on Neural Information Processing Systems*. Vancouver, British Columbia, Canada: MIT Press, pp. 569–576.

Shudong Huang, David Graff, and George Doddington (2002). *Multiple-Translation Chinese Corpus*. Linguistic Data Consortium. Web download. Philadelphia.

Zhongqiang Huang and Mary Harper (2009). “Self-Training PCFG Grammars with Latent Annotations Across Languages”. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*. Singapore: Association for Computational Linguistics, pp. 832–841.

Hongyan Jing (2000). “Sentence reduction for automatic text summarization”. In *Proceedings of the Sixth Conference on Applied Natural Language Processing*. Association for Computational Linguistics, pp. 310–315.

Thorsten Joachims (1998a). *Making large scale SVM learning practical*. Tech. rep. SFB 475: Komplexitätsreduktion in Multivariaten Datenstrukturen. Universität Dortmund.

Thorsten Joachims (1998b). “Text Categorization with Support Vector Machines: Learning with Many Relevant Features”. In *European Conference on Machine Learning (ECML)*. Springer, pp. 137–142.

Marcin Junczys-Dowmunt and Roman Grundkiewicz (2014). “The AMU System in the CoNLL-2014 Shared Task: Grammatical Error Correction by Data-Intensive and Feature-Rich Statistical Machine Translation”. In *Proceedings of the Eighteenth Conference on Computational Natural Language Learning: Shared Task*. Baltimore, Maryland: Association for Computational Linguistics, pp. 25–33.

Marcin Junczys-Dowmunt and Roman Grundkiewicz (2016). “Phrase-based Machine Translation is State-of-the-Art for Automatic Grammatical Error Correction”. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. Austin, Texas: Association for Computational Linguistics, pp. 1546–1556.

J. Peter Kincaid, Robert P. Fishburne Jr., Richard L. Rogers, and Brad S. Chissom (1975). *Derivation of new readability formulas (automated readability index, fog count and flesch reading ease formula) for Navy enlisted personnel*. Tech. rep. DTIC Document.

Tibor Kiss and Jan Strunk (2006). “Unsupervised Multilingual Sentence Boundary Detection”. In *Computational Linguistics* 32.4, pp. 485–525.

Kevin Knight and Daniel Marcu (2000). “Statistics-based summarization – Step one: Sentence compression”. In *Proceedings of the Seventeenth National Conference on Artificial Intelligence (AAAI)*, pp. 703–710.

Kevin Knight and Daniel Marcu (2002). “Summarization beyond sentence extraction: A probabilistic approach to sentence compression”. In *Artificial Intelligence* 139, pp. 91–107.

Philipp Koehn (2005). “Europarl: A parallel corpus for statistical machine translation”. In *Proceedings of the Machine Translation Summit*. Vol. 5.

Philipp Koehn (2012). “Simulating Human Judgment in Machine Translation Evaluation Campaigns”. In *Proceedings of the 9th International Workshop on Spoken Language Translation*, pp. 179–184.

Philipp Koehn and Rebecca Knowles (2017). “Six Challenges for Neural Machine Translation”. In *Proceedings of the First Workshop on Neural Machine Translation*. Vancouver: Association for Computational Linguistics, pp. 28–39.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondrej Bojar, Alexandra Constantin, and Evan Herbst (2007). “Moses: Open Source Toolkit for Statistical Machine Translation”. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceed-*

ings of the Demo and Poster Sessions. Prague, Czech Republic: Association for Computational Linguistics, pp. 177–180.

Julian Kupiec, Jan Pedersen, and Francine Chen (1995). “A trainable document summarizer”. In *Proceedings of the 18th annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, pp. 68–73.

Alon Lavie and Abhaya Agarwal (2007). “METEOR: An Automatic Metric for MT Evaluation with High Levels of Correlation with Human Judgments”. In *Proceedings of the Second Workshop on Statistical Machine Translation*. Prague, Czech Republic: Association for Computational Linguistics, pp. 228–231.

Claudia Leacock, Martin Chodorow, Michael Gamon, and Joel Tetreault (2014). *Automated Grammatical Error Detection for Language Learners, Second Edition*. Synthesis Lectures on Human Language Technologies. Morgan & Claypool Publishers.

Zhifei Li, Chris Callison-Burch, Chris Dyer, Juri Ganitkevitch, Ann Irvine, Sanjeev Khudanpur, Lane Schwartz, Wren Thornton, Ziyuan Wang, Jonathan Weese, and Omar Zaidan (2010). “Joshua 2.0: A Toolkit for Parsing-Based Machine Translation with Syntax, Semirings, Discriminative Training and Other Goodies”. In *Proceedings of the Workshop on Statistical Machine Translation (WMT10)*. Uppsala, Sweden: Association for Computational Linguistics, pp. 133–137.

Rensis Likert (1932). “A technique for the measurement of attitudes”. In *Archives of Psychology* 22.140, pp. 1–55.

Chin-Yew Lin (2003). “Improving summarization performance by sentence compression: A pilot study”. In *Proceedings of the Sixth International Workshop on Information Retrieval with Asian Languages-Volume 11*. Association for Computational Linguistics, pp. 1–8.

Dekang Lin and Patrick Pantel (2001). “Discovery of Inference Rules from Text”. In *Natural Language Engineering* 7.3, pp. 343–360.

Dekang Lin, Kenneth Church, Heng Ji, Satoshi Sekine, David Yarowsky, Shane Bergsma, Kailash Patil, Emily Pitler, Rachel Lathbury, Vikram Rao, Kapil Dalwani, and Sushant Narsale (2010). “New Tools for Web-Scale N-grams”. In *Proceedings of the Seventh conference on International Language Resources and Evaluation (LREC’10)*. Valletta, Malta: European Languages Resources Association (ELRA).

Nina H. MacDonald, Lawrence T. Frase, Patricia S. Gingrich, and Stacey A. Keenan (1982). “The Writer’s Workbench: Computer aids for text analysis”. In *IEEE Transactions on Communications* 30.1.

Nitin Madnani and Bonnie Dorr (2010). “Generating Phrasal and Sentential Paraphrases: A Survey of Data-Driven Methods”. In *Computational Linguistics* 36.3, pp. 341–388.

Nitin Madnani, David Zajic, Bonnie Dorr, Necip Fazil Ayan, and Jimmy Lin (2007). “Multiple Alternative Sentence Compressions for Automatic Text Summarization”. In *Proceedings of the 2007 Document Understanding Conference*. Rochester, NY.

Inderjeet Mani, Gary Klein, David House, Lynette Hirschman, Therese Firmin, and Beth Sundheim (2002). “SUMMAC: A text summarization evaluation”. In *Natural Language Engineering* 8.01, pp. 43–68.

Mitchell P. Marcus, Mary Ann Marcinkiewicz, and Beatrice Santorini (1993). “Building a large annotated corpus of English: The Penn Treebank”. In *Computational Linguistics* 19.2, pp. 313–330.

Marie-Catherine de Marneffe and Christopher D. Manning (2008). *Stanford typed dependencies manual*. Tech. rep. Stanford University.

Erwin Marsi and Emiel Krahmer (2005). “Explorations in sentence fusion”. In *Proceedings of the European Workshop on Natural Language Generation*, pp. 8–10.

Erwin Marsi, Emiel Krahmer, Iris Hendrickx, and Walter Daelemans (2009). “Is Sentence Compression an NLG task?” In *Proceedings of the 12th European Workshop on Natural Language Generation (ENLG 2009)*. Athens, Greece: Association for Computational Linguistics, pp. 25–32.

Erwin Marsi, Emiel Krahmer, Iris Hendrickx, and Walter Daelemans (2010). “On the limits of sentence compression by deletion”. In *Empirical Methods in Natural Language Generation*, pp. 45–66.

André F. T. Martins and Noah A. Smith (2009). “Summarization with a Joint Model for Sentence Extraction and Compression”. In *Proceedings of the Workshop on Integer Linear Programming for Natural Language Processing*. Boulder, Colorado: Association for Computational Linguistics, pp. 1–9.

Ryan McDonald (2006). “Discriminative Sentence Compression With Soft Syntactic Evidence”. In *11th Conference of the European Chapter of the Association for Computational Linguistics (EACL-2006).*

Rada Mihalcea (2007). “Using Wikipedia for Automatic Word Sense Disambiguation”. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Proceedings of the Main Conference*. Rochester, New York: Association for Computational Linguistics, pp. 196–203.

Marcin Miłkowski (2010). “Developing an open-source, rule-based proofreading tool”. In *Software: Practice and Experience* 40.7, pp. 543–566.

Guido Minnen, John Carroll, and Darren Pearce (2001). “Applied morphological processing of English”. In *Natural Language Engineering* 7.03, pp. 207–223.

Tomoya Mizumoto, Mamoru Komachi, Masaaki Nagata, and Yuji Matsumoto (2011). “Mining Revision Log of Language Learning SNS for Automated Japanese Error Correction of Second Language Learners”. In *Proceedings of 5th International Joint Conference on Natural Language Processing*. Chiang Mai, Thailand: Asian Federation of Natural Language Processing, pp. 147–155.

Andrew H. Morris, George M. Kasper, and Dennis A. Adams (1992). “The Effects and Limitations of Automated Text Condensing on Reading Comprehension Performance”. In *Information Systems Research* 3.1, pp. 17–35.

Christof Müller and Iryna Gurevych (2008). “Using Wikipedia and Wiktionary in domain-specific information retrieval”. In *Working Notes of the Annual CLEF Meeting*. Springer, pp. 219–226.

Courtney Napoles (2012). “Computational Approaches to Shortening and Simplifying Text”. Master’s thesis. Johns Hopkins University.

Courtney Napoles and Chris Callison-Burch (2015). “Automatically Scoring Freshman Writing: A Preliminary Investigation”. In *Proceedings of the Tenth Workshop on Innovative Use of NLP for Building Educational Applications*. Denver, Colorado: Association for Computational Linguistics, pp. 254–263.

Courtney Napoles and Chris Callison-Burch (2017). “Systematically Adapting Machine Translation for Grammatical Error Correction”. In *Proceedings of the 12th Workshop on Innovative Use of NLP for Building Educational Applications*. Copenhagen, Denmark: Association for Computational Linguistics, pp. 345–356.

Courtney Napoles and Mark Dredze (2010). “Learning Simple Wikipedia: A Cogitation in Ascertaining Abecedarian Language”. In *Proceedings of the NAACL HLT 2010 Workshop on Computational Linguistics and Writing: Writing Processes and Authoring Aids*. Los Angeles, CA, USA: Association for Computational Linguistics, pp. 42–50.

Courtney Napoles, Benjamin Van Durme, and Chris Callison-Burch (2011a). “Evaluating sentence compression: Pitfalls and suggested remedies”. In *Proceedings of the Workshop on Monolingual Text-To-Text Generation*. Association for Computational Linguistics. Portland, Oregon, pp. 91–97.

Courtney Napoles, Chris Callison-Burch, Juri Ganitkevitch, and Benjamin Van Durme (2011b). “Paraphrastic Sentence Compression with a Character-based Metric: Tightening without Deletion”. In *Proceedings of the Workshop on Monolingual Text-To-Text Generation*. Portland, Oregon: Association for Computational Linguistics, pp. 84–90.

Courtney Napoles, Matthew Gormley, and Benjamin Van Durme (2012). “Annotated Gigaword”. In *Proceedings of the Joint Workshop on Automatic Knowledge Base Construction and Web-scale Knowledge Extraction (AKBC-WEKEX)*. Montréal, Canada: Association for Computational Linguistics, pp. 95–100.

Courtney Napoles, Keisuke Sakaguchi, Matt Post, and Joel Tetreault (2015). “Ground Truth for Grammatical Error Correction Metrics”. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*. Beijing, China: Association for Computational Linguistics, pp. 588–593.

Courtney Napoles, Keisuke Sakaguchi, Matt Post, and Joel R. Tetreault (2016a). “GLEU Without Tuning”. In *CoRR* abs/1605.02592. arXiv: `1605.02592`.

Courtney Napoles, Chris Callison-Burch, and Matt Post (2016b). “Sentential Paraphrasing as Black-Box Machine Translation”. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations*. San Diego, California: Association for Computational Linguistics, pp. 62–66.

Courtney Napoles, Aoife Cahill, and Nitin Madnani (2016c). “The Effect of Multiple Grammatical Errors on Processing Non-Native Writing”. In *Proceedings of the 11th*

Workshop on Innovative Use of NLP for Building Educational Applications. San Diego, CA: Association for Computational Linguistics, pp. 1–11.

Courtney Napoles, Keisuke Sakaguchi, and Joel Tetreault (2016d). “There’s No Comparison: Reference-less Evaluation Metrics in Grammatical Error Correction”. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. Austin, Texas: Association for Computational Linguistics, pp. 2109–2115.

Courtney Napoles, Aasish Pappu, and Joel Tetreault (2017a). “Automatically Identifying Good Conversations Online (Yes, They Do Exist!)” In *International AAAI Conference on Web and Social Media (ICWSM17)*. Montréal, Canada, pp. 628–631.

Courtney Napoles, Joel Tetreault, Aasish Pappu, Enrica Rosato, and Brian Provenzale (2017b). “Finding Good Conversations Online: The Yahoo News Annotated Comments Corpus”. In *Proceedings of the 11th Linguistic Annotation Workshop*. Valencia, Spain: Association for Computational Linguistics, pp. 13–23.

Courtney Napoles, Keisuke Sakaguchi, and Joel Tetreault (2017c). “JFLEG: A Fluency Corpus and Benchmark for Grammatical Error Correction”. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*. Valencia, Spain: Association for Computational Linguistics, pp. 229–234.

Shashi Narayan and Claire Gardent (2014). “Hybrid Simplification using Deep Semantics and Machine Translation”. In *Proceedings of the 52nd Annual Meeting of the Associ-*

ation for Computational Linguistics (Volume 1: Long Papers). Baltimore, Maryland: Association for Computational Linguistics, pp. 435–445.

Hwee Tou Ng, Siew Mei Wu, Yuanbin Wu, Christian Hadiwinoto, and Joel Tetreault (2013). "The CoNLL-2013 Shared Task on Grammatical Error Correction". In *Proceedings of the Seventeenth Conference on Computational Natural Language Learning: Shared Task*. Sofia, Bulgaria: Association for Computational Linguistics, pp. 1–12.

Hwee Tou Ng, Siew Mei Wu, Ted Briscoe, Christian Hadiwinoto, Raymond Hendy Susanto, and Christopher Bryant (2014). "The CoNLL-2014 Shared Task on Grammatical Error Correction". In *Proceedings of the Eighteenth Conference on Computational Natural Language Learning: Shared Task*. Baltimore, Maryland: Association for Computational Linguistics, pp. 1–14.

Diane Nicholls (2003). "The Cambridge Learner Corpus: Error coding and analysis for lexicography and ELT". In *Proceedings of the Corpus Linguistics 2003 conference*. Vol. 16, pp. 572–581.

Sergiu Nisioi, Sanja Štajner, Simone Paolo Ponzetto, and Liviu P. Dinu (2017). "Exploring Neural Text Simplification Models". In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Vancouver, Canada: Association for Computational Linguistics, pp. 85–91.

NIST Multimodal Information Group (2010). *NIST 2005 Open Machine Translation (OpenMT) Evaluation*. Linguistic Data Consortium. Web download. Philadelphia.

Tadashi Nomoto (2008). "A Generic Sentence Trimmer with CRFs". In, pp. 299–307.

REFERENCES

Tadashi Nomoto (2009). “A Comparison of Model Free versus Model Intensive Approaches to Sentence Compression”. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*. Singapore: Association for Computational Linguistics, pp. 391–399.

Charles K. Ogden (1930). *Basic English: A General Introduction with Rules and Grammar*. Paul Treber & Co., Ltd.

Carlos Otero (1972). “Acceptable Ungrammatical Sentences in Spanish”. In *Linguistic Inquiry* 3.2, pp. 233–242.

Paul Over and James Yen (2004). “An introduction to DUC 2004: Intrinsic evaluation of generic news text summarization systems”. In *Proceedings of DUC 2004 Document Understanding Workshop, Boston*.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu (2002). “BLEU: A Method for Automatic Evaluation of Machine Translation”. In *Proceedings of 40th Annual Meeting of the Association for Computational Linguistics*. Philadelphia, Pennsylvania, USA: Association for Computational Linguistics, pp. 311–318.

Y. Albert Park and Roger Levy (2011). “Automated Whole Sentence Grammar Correction Using a Noisy Channel Model”. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*. Portland, Oregon, USA: Association for Computational Linguistics, pp. 934–944.

Robert Parker, David Graff, Junbo Kong, Ke Chen, and Kazuaki Maeda (2011). *English Gigaword Fifth Edition*. Web download. Linguistic Data Consortium.

Ellie Pavlick and Chris Callison-Burch (2016). "Simple PPDB: A Paraphrase Database for Simplification". In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Berlin, Germany: Association for Computational Linguistics, pp. 143–148.

Ellie Pavlick, Pushpendre Rastogi, Juri Ganitkevitch, Benjamin Van Durme, and Chris Callison-Burch (2015). "PPDB 2.0: Better paraphrase ranking, fine-grained entailment relations, word embeddings, and style classification". In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*. Beijing, China: Association for Computational Linguistics, pp. 425–430.

Fernando Pereira, Naftali Tishby, and Lillian Lee (1993). "Distributional Clustering of English Words". In *31st Annual Meeting of the Association for Computational Linguistics*.

Sarah E Petersen and Mari Ostendorf (2007). "Text simplification for language learners: A corpus analysis". In *Proceedings of Workshop on Speech and Language Technology for Education*. Citeseer, pp. 69–72.

Slav Petrov and Dan Klein (2007). "Improved Inference for Unlexicalized Parsing". In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Proceedings of the Main Conference*. Rochester, New York: Association for Computational Linguistics, pp. 404–411.

Slav Petrov, Leon Barrett, Romain Thibaux, and Dan Klein (2006). "Learning accurate, compact, and interpretable tree annotation". In *Proceedings of the 21st International*

Conference on Computational Linguistics and the 44th aNnual Meeting of the Association for Computational Linguistics. Association for Computational Linguistics, pp. 433–440.

Simone Paolo Ponzetto and Michael Strube (2006). “Exploiting semantic role labeling, WordNet and Wikipedia for coreference resolution”. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the ACL*. New York, pp. 192–199.

Matt Post (2011). “Judging Grammaticality with Tree Substitution Grammar Derivations”. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*. Portland, Oregon, USA: Association for Computational Linguistics, pp. 217–222.

Matt Post, Yuan Cao, and Gaurav Kumar (2015). “Joshua 6: A phrase-based and hierarchical statistical machine translation system”. In *The Prague Bulletin of Mathematical Linguistics* 104.1, pp. 5–16.

Ehud Reiter and Anja Belz (2006). “GENEVAL: A proposal for shared-task evaluation in NLG”. In *Proceedings of the Fourth International Natural Language Generation Conference*. Association for Computational Linguistics. Sydney, Australia, pp. 136–138.

Stephen D. Richardson and Lisa C. Braden-Harder (1988). “The Experience of Developing a Large-scale Natural Language Text Processing System: CRITIQUE”. In *Proceedings* *of the Second Conference on Applied Natural Language Processing*. ANLC '88. Austin, Texas: Association for Computational Linguistics, pp. 195–202.

Stefan Riezler, Tracy H. King, Richard Crouch, and Annie Zaenen (2003). "Statistical sentence condensation using ambiguity packing and stochastic disambiguation methods for lexical-functional grammar". In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1*. Association for Computational Linguistics, pp. 118–125.

Alla Rozovskaya and Dan Roth (2010). "Annotating ESL Errors: Challenges and Rewards". In *Proceedings of the NAACL HLT 2010 Fifth Workshop on Innovative Use of NLP for Building Educational Applications*. Los Angeles, California: Association for Computational Linguistics, pp. 28–36.

Alla Rozovskaya and Dan Roth (2016). "Grammatical Error Correction: Machine Translation and Classifiers". In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Berlin, Germany: Association for Computational Linguistics, pp. 2205–2215.

Alla Rozovskaya, Kai-Wei Chang, Mark Sammons, Dan Roth, and Nizar Habash (2014). "The Illinois-Columbia System in the CoNLL-2014 Shared Task". In *Proceedings of the Eighteenth Conference on Computational Natural Language Learning: Shared Task*. Baltimore, Maryland: Association for Computational Linguistics, pp. 34–42.

Alexander M. Rush, Sumit Chopra, and Jason Weston (2015). "A Neural Attention Model for Abstractive Sentence Summarization". In *Proceedings of the 2015 Conference on*

Empirical Methods in Natural Language Processing. Lisbon, Portugal: Association for Computational Linguistics, pp. 379–389.

Keisuke Sakaguchi, Matt Post, and Benjamin Van Durme (2014). "Efficient Elicitation of Annotations for Human Evaluation of Machine Translation". In *Proceedings of the Ninth Workshop on Statistical Machine Translation*. Baltimore, Maryland, USA: Association for Computational Linguistics, pp. 1–11.

Keisuke Sakaguchi, Courtney Napoles, Matt Post, and Joel Tetreault (2016). "Reassessing the Goals of Grammatical Error Correction: Fluency Instead of Grammaticality". In *Transactions of the Association for Computational Linguistics* 4, pp. 169–182.

Keisuke Sakaguchi, Courtney Napoles, and Joel Tetreault (2017a). "GEC into the future: Where are we going and how do we get there?" In *Proceedings of the 12th Workshop on Innovative Use of NLP for Building Educational Applications*. Copenhagen, Denmark: Association for Computational Linguistics, pp. 180–187.

Keisuke Sakaguchi, Matt Post, and Benjamin Van Durme (2017b). "Grammatical Error Correction with Neural Reinforcement Learning". In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*. Taipei, Taiwan: Asian Federation of Natural Language Processing, pp. 366–372.

Ralf Schenkel, Fabian Suchanek, and Gjergji Kasneci (2007). "YAWN: A semantically annotated Wikipedia XML corpus". In *Proceedings of GI-Fachtagung für Datenbanksysteme in Business, Technologie und Web (BTW2007)*. Aachen, Germany.

Allen Schmaltz, Yoon Kim, Alexander Rush, and Stuart Shieber (2017). “Adapting Sequence Models for Sentence Correction”. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. Copenhagen, Denmark: Association for Computational Linguistics, pp. 2807–2813.

Michael D. Shear (2010). “Senators Have Votes, if Time Permits, to Repeal ‘Don’t Ask, Don’t Tell’”. In *The New York Times*. Accessed: December 16, 2010. URL: `https://thecaucus.blogs.nytimes.com/2010/12/16/senators-have-votes-if-time-permits-to-repeal-dont-ask-dont-tell/`.

Advaith Siddharthan (2006). “Syntactic simplification and text cohesion”. In *Research on Language and Computation* 4.1, pp. 77–109.

Jason R. Smith, Chris Quirk, and Kristina Toutanova (2010). “Extracting Parallel Sentences from Comparable Corpora using Document Level Alignment”. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*. Los Angeles, California: Association for Computational Linguistics, pp. 403–411.

Matthew Snover, Bonnie Dorr, Richard Schwartz, Linnea Micciulla, and John Makhoul (2006). “A study of translation edit rate with targeted human annotation”. In *Proceedings of Association for Machine Translation in the Americas*, pp. 223–231.

Hong Sun and Ming Zhou (2012). “Joint Learning of a Dual SMT System for Paraphrase Generation”. In *Proceedings of the 50th Annual Meeting of the Association for Com-*

putational Linguistics (Volume 2: Short Papers). Jeju Island, Korea: Association for Computational Linguistics, pp. 38–42.

Raymond Hendy Susanto, Peter Phandi, and Hwee Tou Ng (2014). “System Combination for Grammatical Error Correction”. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Doha, Qatar: Association for Computational Linguistics, pp. 951–962.

Ben Swanson and Elif Yamangil (2012). “Correction Detection and Error Type Selection as an ESL Educational Aid”. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Montréal, Canada: Association for Computational Linguistics, pp. 357–361.

Joel Tetreault and Martin Chodorow (2008). “Native Judgments of Non-Native Usage: Experiments in Preposition Error Detection”. In *COLING 2008: Proceedings of the Workshop on Human Judgements in Computational Linguistics*. Manchester, UK: COLING 2008 Organizing Committee, pp. 24–32.

Joel Tetreault, Martin Chodorow, and Nitin Madnani (2014). “Bucking the trend: Improved evaluation and annotation practices for ESL error detection systems”. In *Language Resources and Evaluation* 48.1, pp. 5–31.

Simone Teufel and Marc Moens (1997). “Sentence extraction as a classification task”. In *Proccedings of the Workshop on Intelligent Scalable Text Summarization*. Association for Computational Linguistics.

REFERENCES

Kristina Toutanova, Dan Klein, Christopher D Manning, and Yoram Singer (2003). “Feature-rich part-of-speech tagging with a cyclic dependency network”. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1*. Association for Computational Linguistics, pp. 173–180.

Kristina Toutanova, Chris Brockett, M. Ke Tran, and Saleema Amershi (2016). “A Dataset and Evaluation Metrics for Abstractive Compression of Sentences and Short Paragraphs”. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. Austin, Texas: Association for Computational Linguistics, pp. 340–350.

Jenine Turner and Eugene Charniak (2005). “Supervised and Unsupervised Learning for Sentence Compression”. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL’05)*. Ann Arbor, Michigan: Association for Computational Linguistics, pp. 290–297.

Yuya Unno, Takashi Ninomiya, Yusuke Miyao, and Jun’ichi Tsujii (2006). “Trimming CFG parse trees for sentence compression using machine learning approaches”. In *Proceedings of the COLING/ACL on Main Conference Poster Sessions*. Association for Computational Linguistics, pp. 850–857.

Benjamin Van Durme and Ashwin Lall (2010). “Online Generation of Locality Sensitive Hash Signatures”. In *Proceedings of the ACL 2010 Conference Short Papers*. Uppsala, Sweden: Association for Computational Linguistics, pp. 231–235.

Vincent Vandeghinste and Yi Pan (2004). “Sentence Compression for Automated Subtitling: A Hybrid Approach”. In *Proceedings of the ACL Workshop on Text Summarization*.

Yannick Versley, Simone Paolo Ponzetto, Massimo Poesio, Vladimir Eidelman, Alan Jern, Jason Smith, Xiaofeng Yang, and Alessandro Moschitti (2008). “BART: A Modular Toolkit for Coreference Resolution”. In *Proceedings of the ACL-08: HLT Demo Session*. Columbus, Ohio: Association for Computational Linguistics, pp. 9–12.

Tong Wang, Ping Chen, John Rochford, and Jipeng Qiang (2016). “Text Simplification Using Neural Machine Translation.” In *Proceedings of the Conference on Artificial Intelligence (AAAI)*. American Association for Artificial Intelligence, pp. 4270–4271.

Yiming Wang, Longyue Wang, Xiaodong Zeng, Derek F. Wong, Lidia S. Chao, and Yi Lu (2014). “Factored Statistical Machine Translation for Grammatical Error Correction”. In *Proceedings of the Eighteenth Conference on Computational Natural Language Learning: Shared Task*. Baltimore, Maryland: Association for Computational Linguistics, pp. 83–90.

Jonathan Weese, Juri Ganitkevitch, Chris Callison-Burch, Matt Post, and Adam Lopez (2011). “Joshua 3.0: Syntax-based Machine Translation with the Thrax Grammar Extractor”. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*. Edinburgh, Scotland: Association for Computational Linguistics, pp. 478–484.

Wikipedia (2009a). “British Empire”. In *Wikipedia, The Free Encyclopedia*. Accessed: January 6, 2010. URL: https://simple.wikipedia.org/w/index.php?title=British_Empire&oldid=1913719.

Wikipedia (2009b). “Confucianism”. In *Wikipedia, The Free Encyclopedia*. Accessed: January 6, 2010]. URL: https://simple.wikipedia.org/w/index.php?title=Confucianism&oldid=1907391.

Wikipedia (2009c). “How to Write Simple English pages”. In *Simple English Wikipedia*. Accessed: March 1, 2009. URL: https://simple.wikipedia.org/w/index.php?title=Wikipedia:How_to_write_Simple_English_pages&oldid=1348780.

Wikipedia (2009d). “Stephen Hawking”. In *Wikipedia, The Free Encyclopedia*. Accessed: January 31, 2010. URL: https://en.wikipedia.org/w/index.php?title=Stephen_Hawking&oldid=334303159.

Wikipedia (2010a). “Stephen Hawking”. In *Simple English Wikipedia*. Accessed: January 31, 2010. URL: https://simple.wikipedia.org/w/index.php?title=Stephen_Hawking&oldid=1939617.

Wikipedia (2010b). “Umbrella term”. In *Wikipedia, The Free Encyclopedia*. Accessed: July 1, 2010. URL: https://en.wikipedia.org/w/index.php?title=Umbrella_term&oldid=368903880.

Kristian Woodsend and Mirella Lapata (2011). “Learning to simplify sentences with quasi-synchronous grammar and integer programming”. In *Proceedings of the Conference on*

Empirical Methods in Natural Language Processing. Association for Computational Linguistics, pp. 409–420.

Kristian Woodsend, Yansong Feng, and Mirella Lapata (2010). "Title Generation with Quasi-Synchronous Grammar". In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*. Cambridge, MA: Association for Computational Linguistics, pp. 513–523.

Sander Wubben, Antal van den Bosch, and Emiel Krahmer (2012). "Sentence Simplification by Monolingual Machine Translation". In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Jeju Island, Korea: Association for Computational Linguistics, pp. 1015–1024.

Wei Xu, Chris Callison-Burch, and Courtney Napoles (2015). "Problems in Current Text Simplification Research: New Data Can Help". In *Transactions of the Association for Computational Linguistics* 3, pp. 283–297.

Wei Xu, Courtney Napoles, Ellie Pavlick, Quanze Chen, and Chris Callison-Burch (2016). "Optimizing statistical machine translation for text simplification". In *Transactions of the Association for Computational Linguistics* 4, pp. 401–415.

Huichao Xue and Rebecca Hwa (2014). "Improved Correction Detection in Revised ESL Sentences". In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Baltimore, Maryland: Association for Computational Linguistics, pp. 599–604.

Elif Yamangil and Stuart M. Shieber (2010). “Bayesian Synchronous Tree-Substitution Grammar Induction and Its Application to Sentence Compression”. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*. Uppsala, Sweden: Association for Computational Linguistics, pp. 937–947. URL: http://aclweb.org/anthology/P10-1096.

Xiaofeng Yang and Jian Su (2007). “Coreference Resolution Using Semantic Relatedness Information from Automatically Discovered Patterns”. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*. Prague, Czech Republic: Association for Computational Linguistics, pp. 528–535.

Helen Yannakoudakis, Ted Briscoe, and Ben Medlock (2011). “A New Dataset and Method for Automatically Grading ESOL Texts”. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*. Portland, Oregon, USA: Association for Computational Linguistics, pp. 180–189.

Mark Yatskar, Bo Pang, Cristian Danescu-Niculescu-Mizil, and Lillian Lee (2010). “For the sake of simplicity: Unsupervised extraction of lexical simplifications from Wikipedia”. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*. Los Angeles, California: Association for Computational Linguistics, pp. 365–368.

Zheng Yuan and Ted Briscoe (2016). “Grammatical error correction using neural machine translation”. In *Proceedings of the 2016 Conference of the North American Chapter* *of the Association for Computational Linguistics: Human Language Technologies*. San Diego, California: Association for Computational Linguistics, pp. 380–386.

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

David M. Zajic, Bonnie Dorr, Jimmy Lin, and Richard Schwartz (2006). “Sentence compression as a component of a multi-document summarization system”. In *Proceedings of the 2006 Document Understanding Workshop*. Brooklyn, NY.

Raffaella Zanuttini and Laurence Horn (2014). *Micro-syntactic Variation in North American English*. Oxford University Press.

Ying Zhang, Almut Silja Hildebrand, and Stephan Vogel (2006). “Distributed Language Modeling for N-best List Re-ranking”. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*. Sydney, Australia: Association for Computational Linguistics, pp. 216–223.

Shiqi Zhao, Haifeng Wang, Ting Liu, and Sheng Li (2008). “Pivot Approach for Extracting Paraphrase Patterns from Bilingual Corpora”. In *Proceedings of ACL-08: HLT*. Columbus, Ohio: Association for Computational Linguistics, pp. 780–788.

Shiqi Zhao, Xiang Lan, Ting Liu, and Sheng Li (2009). “Application-driven Statistical Paraphrase Generation”. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language*

Processing of the AFNLP. Suntec, Singapore: Association for Computational Linguistics, pp. 834–842.

Zhemin Zhu, Delphine Bernhard, and Iryna Gurevych (2010). “A Monolingual Tree-based Translation Model for Sentence Simplification”. In *Proceedings of the 23rd International Conference on Computational Linguistics (COLING 2010)*. Beijing, China: COLING 2010 Organizing Committee, pp. 1353–1361.

Vita

Courtney Napoles received her PhD and MSE in computer science from Johns Hopkins University, where she was co-advised by Chris Callison-Burch and Benjamin Van Durme. She was affiliated with the Center for Language and Speech Processing and was supported by the National Science Foundation Graduate Research Fellowship. During her PhD, she interned at Educational Testing Service (ETS) and Yahoo Research. She holds a bachelor's degree in psychology from Princeton University with a certificate in linguistics. Before graduate school, she edited non-fiction books for a trade publisher. She is currently a research scientist with Grammarly.