COMPUTATIONAL GRAMMATICAL ERROR CORRECTION: BRIDGING THE GAP FROM ACADEMIA TO INDUSTRY

SIMON FLACHS

Ph.D. Thesis
February 2021

SUPERVISORS:
Anders Søgaard
Ophélie Lacroix

ASSESSMENT COMMITTEE:
Isabelle Augenstein, University of Copenhagen
Joel Tetrault, Dataminr
Zornitsa Kozareva, Google

AFFILIATION:

- Department of Computer Science
  Faculty of Science
  University of Copenhagen
- Siteimprove

Simon Flachs: *Computational Grammatical Error Correction: Bridging the Gap from Academia to Industry*, February 2021

ABSTRACT

Grammatical Error Correction (GEC) is the research field concerned with computational methods for correcting grammatical errors in text. With the vast amounts of content currently being produced online, these methods hold the promise of improving human communication by enabling clear and error-free prose.

While GEC is a thoroughly studied field in academia, industrial adoption has been limited. Three specific obstacles are particularly holding back wide-spread industrial adoption: current academic GEC systems 1) depend on a lot of expensive data for training the systems; 2) are mostly evaluated on text written by English language learners, leaving the systems' performance beyond this domain unclear; and 3) are mainly developed for the English language.

This thesis presents research into tackling these obstacles, in order to bridge the gap between academic research and industrial use. In the first part of the thesis, we investigate two avenues for building low-resource GEC systems. Firstly, we show that leveraging artificially generated training data improves systems' ability to detect subject-verb-agreement errors, particularly improving robustness to challenging linguistic phenomena. Secondly, we show that language models trained by self-supervision can be used for creating viable GEC systems that do not rely on annotated training data. In the second part of the thesis, we look into GEC systems' ability to generalize beyond the English language learner domain – we release a new GEC benchmark, CWEB, consisting of website text annotated for correctness, and show that current GEC systems do not generalize well to this domain. In the final part, we focus on GEC for non-English languages and investigate strategies for leveraging available sources of noisy data. We show that GEC systems pre-trained on noisy data can be fine-tuned effectively on only small amounts of expert-annotated data, which opens up for creating inexpensive GEC systems in new languages.

ABSTRACT IN DANISH – RESUMÉ PÅ DANSK

Grammatical Error Correction (GEC) er forskningsfeltet, der beskæftiger sig med algoritmiske metoder til at rette grammatiske fejl i tekst. Med de store mængder indhold, der i øjeblikket produceres online, har disse metoder potentiale til at forbedre menneskelig kommunikation ved at gøre den skrevne tekst klar og fejlfri.

Mens GEC er et grundigt studeret felt i den akademiske verden, har industriel anvendelse været begrænset. Især tre specifikke forhindringer holder udbredt industriel anvendelse tilbage: nuværende akademiske GEC systemer 1) afhænger af store mængder dyr data til træning af systemerne; 2) evalueres for det meste på tekst skrevet af studerende med engelsk som fremmedsprog, hvilket gør det uklart, hvordan systemerne klarer sig uden for dette domæne; og 3) er hovedsageligt udviklet til det engelske sprog.

Denne afhandling præsenterer forskning i at overkomme disse forhindringer for at bygge bro mellem akademisk forskning og industriel anvendelse. I den første del af afhandlingen undersøger vi to mulige veje til at bygge lav-ressource GEC-systemer. Først viser vi, at brugen af kunstigt genereret træningsdata forbedrer systemers evne til at opdage uoverensstemmelse i subjekt og verbums bøjning. Det forbedrer især robustheden over for udfordrende sproglige fænomener. For det andet viser vi, at sprogmodeller trænet med selv-supervision, kan bruges til at skabe GEC-systemer af rimelig kvalitet, der ikke afhænger af annoteret træningsdata. I anden del af afhandlingen ser vi på GEC-systemers evne til at generalisere ud over det engelske sprogindlærings domæne – vi lancerer et nyt GEC-benchmark, CWEB, der består af hjemmesidetekst annoteret for korrekthed, og viser, at nuværende GEC-systemer ikke generaliserer godt til dette domæne. I den sidste del fokuserer vi på GEC for ikke-engelske sprog og undersøger strategier til brug af tilgængelige kilder til støjfyldt data. Vi viser, at GEC-systemer, der er pre-trænede på støjfyldt data, kan fintunes effektivt på kun små mængder ekspert-annoteret data. Dette muliggør udviklingen af billige GEC-systemer på nye sprog.

PUBLICATIONS

This is an article-based thesis, where chapters 3 to 6 each represent a peer-reviewed article. The articles as they appear in the thesis are identical to the original publications, barring minor changes to the formatting. The following articles are included in the thesis:

Flachs, Simon, Ophélie Lacroix, Marek Rei, Helen Yannakoudakis, and Anders Søgaard (2019). "A Simple and Robust Approach to Detecting Subject-Verb Agreement Errors". In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 2418-2427

Flachs, Simon, Ophélie Lacroix, and Anders Søgaard (2019). "Noisy Channel for Low Resource Grammatical Error Correction". In: *Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pp. 191-196

Flachs, Simon, Ophélie Lacroix, Helen Yannakoudakis, Marek Rei, and Anders Søgaard (2020). "Grammatical Error Correction in Low Error Density Domains: A New Benchmark and Analyses". In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 8467-8478

Flachs, Simon, Felix Stahlberg, and Shankar Kumar (2021). "Data Strategies for Low-Resource Grammatical Error Correction". In: *Proceedings of the Sixteenth Workshop on Innovative Use of NLP for Building Educational Applications*

I was also involved in the following publication which is not included in this thesis:

Flachs, Simon, Marcel Bollmann, and Anders Søgaard (2019). "Historical Text Normalization with Delayed Rewards". In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 1614-1619

ACKNOWLEDGMENTS

I want to say thank you to the many people that in one way or another enabled me to complete this PhD. First, I'd like to thank my PhD supervisors Anders Søgaard and Ophélie Lacroix. Ophélie, thank you for all your support and your limitless patience. Anders, thank you for your guidance and never-failing optimism. I also want to thank my colleagues at Siteimprove. Special thanks go to Claus Lenander Jensen for his vision and for taking a chance on this project. Thanks also to Søren Jacobsen, Jannick Johnsen, and Nicolai Christensen for the company and inspiring discussions. Thank you to all the wonderful people at CoAStaL at the University of Copenhagen – truly the friendliest research group in Natural Language Processing. I am also grateful to Helen Yannakoudakis and Marek Rei for the productive collaborations and exchange of ideas. Thanks also to Shankar Kumar and Felix Stahlberg, working with you has been one of the high points of my PhD. I am very fortunate to have had the support of family and friends on the way. I am particularly grateful to my father, Allan, and my mother, Anette. Lastly, I would like to express my gratitude to Siteimprove and the Innovation fund of Denmark for funding this PhD project.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACRONYMS

AESW	Automated Evaluation of Scientific Writing Shared Task 2016.
BEA-2019	Building Educational Applications 2019 Shared Task.
BERT	Bidirectional Encoder Representations from Transformers.
BNC	British National Corpus.
CoNLL	Conference on Computational Natural Language Learning.
CWEB	Corrected Websites corpus.
ELLs	English language learners.
ESL	English as a Second Language.

FCE	The Cambridge Learner Corpus of First Certificate in English.
GEC	Grammatical Error Correction.
GED	Grammatical Error Detection.
GMEG	Grammarly Multi-domain Evaluation for GEC Data set.
JFLEG	JHU FLuency-Extended GUG corpus.
ML	Machine Learning.
NLP	Natural Language Processing.
NUCLE	The NUS Corpus of Learner English.
PoS	Part-Of-Speech.
PTB	Penn Treebank.
RNN	Recurrent Neural Network.
SOTA	State-of-the-Art.
SVA	subject–verb agreement.
UD	Universal Dependencies.
W&I	Write & Improve corpus.

Part I

BACKGROUND

1 INTRODUCTION

1.1 MOTIVATION

Since its invention, the written medium has been a decisive factor for humanity's societal progress. The ability to persist ideas in writing and share them across time and space has dramatically increased the speed and reach of knowledge sharing and communication. The written medium's importance holds increasingly true in the current digital age, where the communication speed and range enabled by the internet is a cornerstone of modern society (Bazerman, 2013).

A typical hindrance to effective written communication is the presence of grammatical errors, which writers of all skill levels are prone to make. Grammatical errors are a source of distraction and confusion for the reader and can result in the message not being conveyed properly, or in the worst case cause misinformation by changing the meaning of the message (Tomiyana, 1980). It could also have commercial implications by decreasing the perceived trustworthiness of the source of the text (Appelman and Schmierbach, 2018). When adding up the vast amount of written communication done each day, these errors present a large impediment to the overall effectiveness of the world's communication.

The quality of written work is commonly improved by getting a human language expert to proofread it. This process is, however, expensive, time-consuming, and the quality can vary. Therefore the ability to use computational methods to automatically detect and correct grammatical errors in text carries a lot of potential for augmenting human communication.

In recent decades the task of computationally correcting grammatical errors in text has been approached with Machine Learning (ML) methods coming from the field of Natural Language Processing (NLP). These NLP methods hold the promise of enabling high quality automatic error checking, which can be scaled up to processing large amounts of text. This would open up opportunities for many industrial applications of automatic error correction.

In this industrial PhD project, we investigate computational grammatical error correction from a commercial point of view, aiming to tackle roadblocks on the way to industrial adoption. The following section expands on computational methods for correcting grammatical errors.

1.2 NATURAL LANGUAGE PROCESSING FOR GRAMMATICAL ERROR CORRECTION

In the NLP field, computational methods for correcting grammatical errors are studied under the term Grammatical Error Correction (GEC). GEC can be cast as the problem of transforming grammatically incorrect text into an error-free version. In the last decade, the NLP field has progressed immensely, mostly due to advances in the subfield of ML called Deep Learning. This has, in turn, led to rapid progress in GEC, enabling higher quality checks for grammatical correctness. GEC methods now go beyond brittle rule-based approaches, and are able to correct errors that require a deeper syntactic and sometimes semantic understanding of the sentence.

Current GEC methods use supervised ML algorithms, where a ML model uses large amounts of human-annotated examples to learn a function that maps from inputs to outputs. The model then uses the learned function to perform inference on unseen data. The human-annotated examples consist of pairs of sentences before and after being corrected for grammatical errors – optimally, these corrections are carried out by professional proofreaders. As this data is expensive to acquire, only small amounts of annotated data are available for training models, which poses a challenge for the data-hungry supervised methods. A large part of the recent progress in GEC has stemmed from learning to take advantage of auxiliary data sources of lower quality than expert-annotated data.

GEC in academia has mostly been focused on developing systems for English language learners (ELLs). Here, the systems are trained and evaluated on error-annotated text written by ELLs. While this is an important task, this narrow focus has limited the field; it remains unclear how current systems perform across different domains.

Another aspect of GEC, which remains largely unexplored, is the development of GEC systems in languages other than English. This is mostly due to the fact that, until recently, annotated data in other languages has simply not been available to train and evaluate GEC systems. Furthermore, GEC approaches have historically been heavily dependent on expert-level linguistic knowledge of the language. It is only with recent data-driven methods that we can hope for approaches that generalize across languages.

1.3 FROM ACADEMIA TO INDUSTRY

While GEC is a well-studied field in academia, industrial adoption has been limited. Approaches taken in industry have generally been very dependent on expert-level linguistic knowledge of the language. Therefore, the industrial landscape has been dominated by a few large organizations, with the economic resources to employ a set of

computational linguists for each of the languages they desire GEC systems in (Smith, 2016). The costs of this approach have especially put a limit on the quality of GEC on smaller languages.

A lot of untapped potential lies in transferring the quality of current state-of-the-art academic algorithms to industry and broadening them to other languages: Word-processing tools could offer better grammar checking, as well as chat tools, email providers, etc. Furthermore, educational applications could enable quick feedback of grammatical correctness to language learners. Grammar checking algorithms could also be used as a post- or pre-processing step for other NLP tools, such as using GEC to normalize erratic speech patterns captured by automatic speech recognition tools.

This thesis is based on an industrial PhD project funded by the company Siteimprove. The aim of the project has been to investigate automatic methods of correcting text from a commercial point of view, in particular for Siteimprove's use cases. Siteimprove offers a web-based platform that assists companies in improving their digital presence. This currently involves automatically spell-checking text on websites, but a lot of commercial potential lies in more advanced checks for sentence correctness. Siteimprove is therefore interested in adding capabilities for correcting grammatical errors to their suite of products.

Given the industrial focus of this PhD project, the project's aim has been two-fold: 1) identify the boundaries of current GEC approaches in regard to industrial usage; and 2) actively push the research horizon of the field towards industrial usability.

When considering the industrial use cases of GEC, three considerations are particularly important. Firstly, data acquisition costs should be manageable. Secondly, GEC models should perform well across many domains. And finally, the methods should generalize to other languages than English. This has led us to investigate three core research questions:

How can data-scarcity in GEC be dealt with?

State-of-the-art GEC systems in academic research are heavily dependent on annotated GEC data for training the models. With few exceptions, this data is only available for research use and not for commercial use, and acquiring new data is expensive.

How do GEC systems perform outside the ELL domain?

In academia, GEC systems are usually trained and evaluated on data from essays written by ELLs. However, it is important that models generalize outside this domain to different types of text written by more advanced writers, which might contain errors that are more subtle and more difficult to detect and correct.

How can GEC be broadened to non-English languages?

Although little work has been done on GEC for non-English languages, a lot of commercial potential lies there. It is yet uncertain to which extent we can apply methods from English GEC to other languages. Data-scarcity, in particular, is a large issue, since when moving beyond English, little to no high quality GEC data is available.

1.4 CONTRIBUTIONS OF THE THESIS

This thesis presents research into the challenges of bringing GEC from academia to industry. The contributions presented in the following chapters are summarized below.

- We show that leveraging artificially generated training data results in a substantial positive impact on the performance of systems for detecting subject-verb agreement errors. Including artificial errors with human-annotated training data makes the system more robust to other errors in the sentence and challenging linguistic phenomena (Chapter 3 in Part ii).
- We show that by leveraging strong language models trained by self-supervision for GEC, reasonable performance can be achieved while allowing the system to be fully unsupervised (Chapter 4 in Part ii).
- We contribute a new GEC benchmark of website text annotated for correctness and show that state-of-the-art GEC systems do not generalize well to this new domain (Chapter 5 in Part iii).
- We show a set of best practices for data generation and utilization for GEC on non-English languages (Chapter 6 in Part iv).

1.5 THESIS OVERVIEW

The thesis is divided into four parts. The following Chapter 2 in the first part provides an introduction to the background of the GEC task.

Part ii of this thesis is focused on answering the first research question, by investigating approaches to deal with data-scarcity in GEC. We follow two particularly promising lines of investigation for ameliorating data-scarcity: artificial error generation and self-supervision.

- In Chapter 3, we focus on using artificially generated errors to train systems. While artificial errors have been shown to benefit GEC systems, not much analysis has been done on how they affect the capabilities of systems. We focus on a specific linguistic phenomenon, subject-verb-agreement, and limit the task to error detection, a precursor to error correction. We show that large

amounts of realistic errors can be easily generated. Combining this data with available high-quality training data leads to more robust detection of subject-verb agreement errors – performance is improved on both in-domain and out-of-domain benchmarks, as well as in the context of other errors, long-distance dependencies, and other challenging linguistic phenomena.

- In Chapter 4, we present our research on unsupervised GEC by leveraging language models trained with self-supervision, which was a contribution to the low-resource track of the BEA 2019 shared task on GEC (Bryant et al., 2019). Our approach to GEC builds on the theory of the noisy channel by combining a channel model and language model. We generate confusion sets from the Wikipedia edit history and use the frequencies of edits to estimate the channel model. We experiment with using two state-of-the-art language models and show that basing GEC systems on pure language modeling is a viable approach to unsupervised GEC.

In Part iii of the thesis, we present our work on GEC beyond the ELL domain to answer the second research question.

- In Chapter 5, our work on GEC for low-error density domains is presented, which introduces a new GEC benchmark, CWEB, consisting of corrected websites. This domain differs from existing benchmarks by containing far fewer errors, which we show poses a challenge to state-of-the-art systems. We suggest that a reason behind this is that GEC systems cannot rely on a strong internal language model in low error-density domains.

Part iv presents our work on GEC for non-English languages to answer our third research question.

- In Chapter 6, we look into determining a set of best-practice strategies for data generation and usage for non-English languages. We show that for morphologically rich languages, artificial error generation methods can benefit from including morphology-based confusion sets. We also demonstrate that Wikipedia revisions and the website Lang8 are very useful sources of data for pre-training systems, despite their inherent noise. We further show that not much expert-annotated data is needed when leveraging these noisy data sources.

Finally, in Part v, we summarize, discuss and conclude on the contributions made in this thesis.

2 BACKGROUND

The work presented in this thesis is a contribution to the field of Grammatical Error Correction (GEC). The GEC task has seen immense progress throughout the last decades. The progress has particularly accelerated in recent years due to increasing attention being placed on GEC, as well as on the NLP field in general. In the following, an overview of the background of the GEC field is provided, setting the starting point for the contributions of this thesis.

2.1 TASK DEFINITION

Grammatical Error Correction can be defined as an automatic sequence-to-sequence task – that is, a GEC system should automatically transform a sequence of words of potentially incorrect language to a sequence of words in correct language while retaining the meaning of the sentence. An example of an error-correction pair is:

(1) ~~Its~~ always a pleasure to ~~here~~ from you.
It's always a pleasure to **hear** from you.

In practice, the GEC task is not restricted to correcting grammatical errors, but covers other aspects of sentence correctness, such as lexical and orthographic errors. The quality of GEC systems is measured by the degree of agreement between a system's predictions and a test corpus. This test corpus is a dataset of text which has been annotated for correctness by human experts. Thereby, the types of errors to correct are defined by the specific corpus used to benchmark a system.

2.2 APPROACHES

Research on GEC can be traced back to the 1980s (Smith, Kiefer, and Gingrich, 1984), with different paradigms in the approaches taken being dominant throughout time. The general trend is that methods leveraging more specialized human linguistic knowledge have been gradually replaced by more general data-dependent methods. In the following, we give an overview of the different approaches that have been prevalent throughout time.

RULES First approaches to GEC were based on hand-crafted rules (e.g., Park, Palmer, and Washburn (1997) and Schneider and McCoy (1998)). However, the complexity of language, with the many

exceptions to the rules, made it unfeasible for this approach to yield high-quality performance across many languages.

CLASSIFIERS Given progress in data-driven methods, the field started moving away from expert-knowledge based systems towards hybrid methods, where computational linguists trained classifiers to detect and correct specific error types (e.g., De Felice and Pulman (2007), Rozovskaya et al. (2013), Rozovskaya and Roth (2011), and Tetreault, Foster, and Chodorow (2010)). However, the classification approach was only able to correct a subset of error types. Furthermore, separate classifiers were typically trained for different error classes, which meant the system could not account for the complex combinations of interacting errors that are often present.

LANGUAGE MODELS Another approach taken was to leverage language models (e.g., Gamon et al. (2008) and Turner and Charniak (2007)). Here, sequences of words assigned a low probability by the language model were assumed to be erroneous and could be corrected by being replaced with a sequence of words of higher probability. However, a weakness of this approach is its inability to distinguish between rare but correct sequences of words and erroneous ones.

STATISTICAL MACHINE TRANSLATION In the last decade, the field has moved towards modeling the problem as a machine translation problem, where incorrect sentences are translated to corrected sentences. Statistical machine translation approaches gave a large improvement over previous methods as they were better able to draw knowledge from corpora of error-correction examples (e.g., Felice et al. (2014) and Junczys-Dowmunt and Grundkiewicz (2016)).

NEURAL MACHINE TRANSLATION More recently, neural machine translation methods have surpassed the statistical ones by better being able to extract knowledge from large amounts of training data. Several iterations of neural sequence-to-sequence models have yielded greater performance in GEC: Xie et al. (2016) employed Long Short-Term Memory networks (Hochreiter and Schmidhuber, 1997), which was followed by Chollampatt and Ng (2018b) using convolutional neural networks (Fukushima, 1980) and more recently Kiyono et al. (2019) used the transformer architecture (Vaswani et al., 2017).

EDIT BASED MODELS Most recently, a new paradigm of edit-based models has yielded state-of-the-art results (e.g., Awasthi et al. (2019), Omelianchuk et al. (2020), and Stahlberg and Kumar (2020)). This approach leverages that GEC is, in fact, different from machine translation – in GEC, most of the time, a majority of the input sequence should not be changed, as opposed to machine translation where there

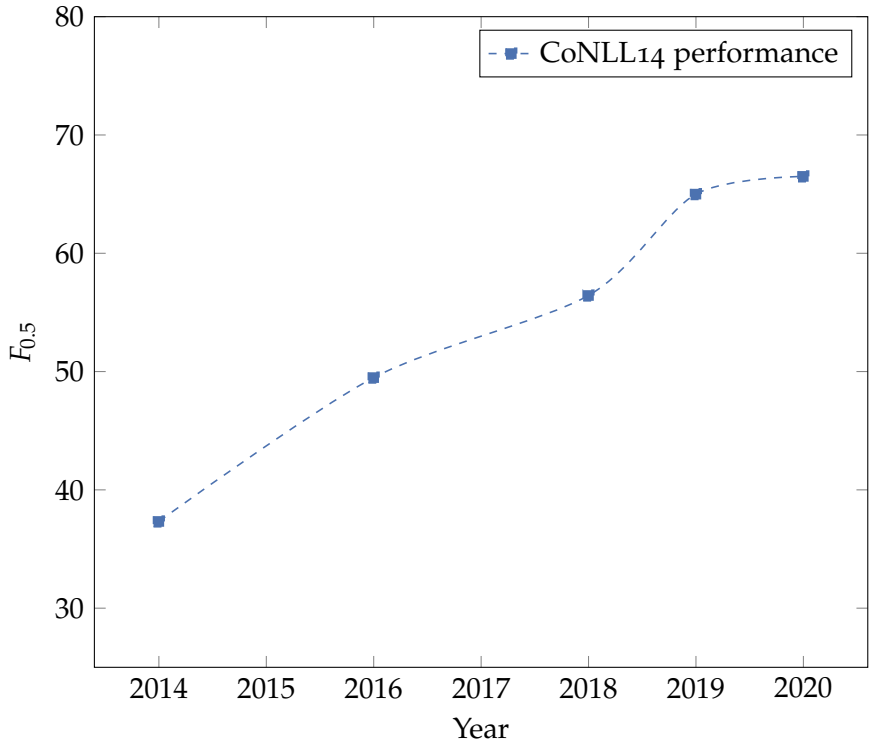


Figure 2.1: Progress in GEC performance ($F_{0.5}$) on CoNLL14 since 2014.

usually is not much overlap between the input and output sequence. The edit-based approach simply aims to only output the edits.

2.3 PROGRESS

The progress in the quality of GEC systems has been remarkable. A common benchmark for measuring the quality of GEC systems is the test set of the CoNLL-2014 shared task on GEC (CoNLL14) (Ng et al., 2014), where the performance is measured by the $F_{0.5}$ score. Below is an overview of state-of-the-art approaches throughout the years since the benchmark was introduced. The systems' scores on CoNLL14 are plotted in Figure 2.1, where a clear upward-going trend is evident, while the paradigms have moved from statistical machine translation to neural machine translation to edit based models.

- **2014** The winner of the CoNLL-2014 shared task on GEC was Felice et al. (2014), who used a hybrid system consisting of a pipeline of several approaches: rules derived from corpora, language modeling, statistical machine translation, and finally, a rule-based filtering of the suggestions.
- **2016** Junczys-Dowmunt and Grundkiewicz (2016) explored exploiting the full potential of statistical machine translation, by tuning the model towards the *MaxMatch* GEC metric (Dahlmeier

and Ng, 2012), and using feature engineering to create task-specific input features.

- **2018** Chollampatt and Ng (2018b) used a convolutional sequence-to-sequence approach combined with re-ranking output hypotheses with a quality estimation model.
- **2019** Kiyono et al. (2019) used a transformer sequence-to-sequence model pre-trained with large amounts of artificial data.
- **2020** Omelianchuk et al. (2020) used an edit-based model, employing a pre-trained transformer model to tag sequences with a set of custom grammatical transformations.

2.4 DATA

The current state-of-the-art methods are supervised ML algorithms trained using parallel corpora, consisting of pairs of sentences before and after being corrected. Especially the current deep learning-based methods are very data-hungry, requiring large amounts of parallel data. Optimally systems are trained on expert annotated *Gold* data. As this resource is scarce, a lot of work has been done leveraging data sources of lower quality, such as artificially generated errors, edits mined from Wikipedia revisions, and crowdsourced data. The data sources are described in more detail in the following.

GOLD DATA *Gold* data is created by human experts, such as linguists or professional proofreaders. For English, a range of high-quality *Gold* corpora is available, totaling more than 100,000 training examples (Bryant et al., 2019; Dahlmeier, Ng, and Wu, 2013b; Yannakoudakis, Briscoe, and Medlock, 2011). For other languages, however, little to no high-quality data is available. But even for English, these amounts are still relatively small for training deep learning systems. Furthermore, most of these datasets are generated from texts written by ELLs, which limits the data's applicability to other domains. Additionally, this data is generally only available for research use, preventing it from being used to train commercial systems.

ARTIFICIAL ERRORS The grammatical structure of a sentence can easily be destroyed by injecting noise into the fluent sentence – thereby, artificial training examples can be created cheaply in large amounts. It has been shown that pre-training GEC systems on artificial data is an essential component of state-of-the-art systems (Kiyono et al., 2019). Several methods for generating artificial errors have given good results, including round-trip translations (Lichtarge et al., 2019), rule-based noising (Grundkiewicz, Junczys-Dowmunt, and Heafield, 2019) and back-translation (Kiyono et al., 2019).

WIKIPEDIA REVISIONS Wikipedia keeps a record of all versions of its articles throughout history. By taking sentences altered in adjacent versions of Wikipedia articles, it is possible to extract examples to train GEC systems. These examples are useful since a lot of revisions are correcting grammatical errors in the articles. However, this data also contains a lot of noise since many revisions are unrelated to the GEC task, e.g., containing information change or acts of vandalism. Despite this noise, Wikipedia revisions have shown to be very useful for training GEC systems (Grundkiewicz and Junczys-Dowmunt, 2014; Lichtarge et al., 2019).

CROWDSOURCED DATA *Lang8*[1] is a web platform where language learners post essays, which native speakers correct. By scraping this website, a GEC dataset of crowdsourced corrections can be generated (Mizumoto et al., 2011). In general, the data is of good quality, although some characteristics of the crowdsourced nature make it less clean than *Gold* data. Furthermore, the data is produced by language learners and therefore represents a narrow domain. Substantial amounts of data are available for larger languages, such as English and Japanese, but the quantity drops drastically for smaller languages.

2.5 EVALUATION

GEC algorithms are measured by their performance on an unseen test set – inference is performed, and the predicted outputs are compared with human annotations. The end goal is to agree with the humans; however, perfect agreement is unattainable since what constitutes an error is subjective, and often there are many different valid ways to correct a mistake (Bryant and Ng, 2015). The following describes the various scoring measures and benchmark datasets commonly used to score systems.

MEASURES As measuring the performance of GEC algorithms is an unsolved problem, several scoring measures have been proposed, e.g., GLEU (Mutton et al., 2007), I-measure (Felice and Briscoe, 2015), MaxMatch (Dahlmeier and Ng, 2012), and ERRANT (Bryant, Felice, and Briscoe, 2017). In this thesis, the presented systems are evaluated based on the $F_{0.5}$ score calculated by either MaxMatch or ERRANT. The $F_{0.5}$ score puts two times more weight on precision than recall, reflecting the importance of offering the end-user precise suggestions.

BENCHMARKS In the last decade, several shared tasks have helped set the direction of GEC research and have pushed the field forward significantly, e.g, HOO-2012 (Dale, Anisimoff, and Narroway, 2012), CoNLL-2013 (Ng et al., 2013), CoNLL-2014 (Ng et al., 2014), AESW

1 https://lang-8.com/

(Daudaravicius et al., 2016), and BEA-2019 (Bryant et al., 2019). Particularly, the test sets of the CoNLL-2014 shared task, and the BEA-2019 shared task are commonly used to benchmark systems. CoNLL-2014 focused on GEC for essays written by ELLs. The more recent BEA-2019 shared task focused on both essays written by ELLs and native speakers. In addition to the test sets from shared tasks, several other benchmarks are available, e.g., FCE (Yannakoudakis, Briscoe, and Medlock, 2011), JFLEG (Napoles, Sakaguchi, and Tetreault, 2017), and GMEG (Napoles, Nădejde, and Tetreault, 2019).

2.6 CURRENT TRENDS

As described in Section 2.3, English GEC has seen immense progress in the last decade. However, many areas are still under active exploration, and several current research trends are challenging some of the weaknesses in current approaches. In the following, we highlight recent work related to the three core research questions of this thesis.

LOW-RESOURCE SYSTEMS Current state-of-the-art approaches to GEC are based on data-hungry neural models, which require large numbers of high-quality training examples. Several lines of investigation into low-resource approaches are currently showing promise: A lot of benefit is seen from leveraging lower quality data sources described in Section 2.4, such as Wikipedia edits and artificial data (Grundkiewicz, Junczys-Dowmunt, and Heafield, 2019; Kiyono et al., 2019; Lichtarge et al., 2019). However, there remains a need for a more in-depth analysis of how training on artificial errors impacts the grammatical understanding of systems – our work in Chapter 3 delves into this aspect. Recently, language model-based approaches have been revisited for low-resource GEC (Bryant and Briscoe, 2018). Our work in Chapter 4 shows that unsupervised GEC systems of reasonable quality can be created by leveraging state-of-the-art neural language models – other concurrent work have similarly achieved strong results using this approach (Alikaniotis and Raheja, 2019; Stahlberg, Bryant, and Byrne, 2019).

NON-ELL DOMAINS GEC systems are commonly evaluated based on their performance on benchmarks from the ELL domain. This domain, however, only represents part of the full spectrum of GEC applications. In the last few years, some research has gone into broadening GEC into other domains. The BEA-2019 shared-task (Bryant et al., 2019) evaluated the contributions in part on a set of essays written by native speakers, and showed that the approaches, in general, transferred well across essays written by beginner-level learners to native speakers. Napoles, Nădejde, and Tetreault (2019) released a new benchmark, GMEG, which consists of corrected text gathered from

Wikipedia revisions and online comments, and showed that current systems also generalize well across these domains. Despite these contributions showing good generalizability in particular domains, a lot of future work still lies in assessing GEC in an open-domain setting. Our work in Chapter 5 highlights one domain where current GEC systems do not generalize well.

NON-ENGLISH LANGUAGES Research on GEC in languages other than English is still in its infancy, mostly due to a lack of *Gold* corpora in other languages. But in recent years, several GEC corpora have been released for languages such as German (Boyd, 2018), Spanish (Davidson et al., 2020), Czech (Náplava and Straka, 2019), and Russian (Rozovskaya and Roth, 2019), which has enabled investigations into GEC in these languages. Some lines of investigation into non-English languages have shown the usefulness of techniques from English GEC, such as using artificial data (Grundkiewicz and Junczys-Dowmunt, 2019; Náplava and Straka, 2019) and Wikipedia revisions (Boyd, 2018). But there remains a need for a more thorough investigation of best practices for data generation and utilization across languages, which is the aim of our work in Chapter 6.

Part II

LOW-RESOURCE APPROACHES

3

A SIMPLE AND ROBUST APPROACH TO DETECTING SUBJECT–VERB AGREEMENT ERRORS

ABSTRACT

While rule-based detection of subject–verb agreement (SVA) errors is sensitive to syntactic parsing errors and irregularities and exceptions to the main rules, neural sequential labelers have a tendency to overfit their training data. We observe that rule-based *error generation* is less sensitive to syntactic parsing errors and irregularities than error detection and explore a simple, yet efficient approach to getting the best of both worlds: We train neural sequential labelers on the combination of large volumes of silver standard data, obtained through rule-based error generation, and gold standard data. We show that our simple protocol leads to more robust detection of SVA errors on both in-domain and out-of-domain data, as well as in the context of other errors and long-distance dependencies; and across four standard benchmarks, the induced model on average achieves a new state of the art.

3.1 INTRODUCTION

GRAMMATICAL ERROR DETECTION. Grammatical Error Detection (GED, Leacock et al., 2010) is the task of detecting grammatical errors in text. It is used in various real-world applications, such as writing assistance tools, self-assessment frameworks and language tutoring systems, facilitating incremental and/or exploratory editing of one's writing. Accurate error detection systems also have potential applications for language generation and machine translation systems, guiding automatically generated output towards grammatically correct sequences.

The problem of detecting subject–verb agreement (SVA) errors is an important subtask of GED. In this work, we focus on detecting subject–verb agreement errors in the English as a Second Language (ESL) domain. Most SVA errors occur at the third-person present tense when determining whether the subject describes a singular or a plural concept. The following examples demonstrate subject–verb agreement errors (bold):

(1) a. *They all **knows** where the conference is.
b. *The Hotel **are** very close to Town Hall.

The task can be formulated as a sequence labeling problem, with the goal of labeling subject–verb pairs as being in agreement or not.

APPROACHES. Sequence labeling problems in NLP, including GED and the subtask of identifying SVA errors, have, in recent years, been handled with Recurrent Neural Networks (RNNs) trained on large amounts of data (Rei et al., 2017; Rei and Yannakoudakis, 2016). However, most publicly available datasets for GED are relatively small, making it difficult to learn a general grammar representation and potentially leading to over-fitting. Previous work has also shown that neural language models with a similar architecture have difficulty learning subject–verb agreement patterns in the presence of agreement attractors (Linzen, Dupoux, and Goldberg, 2016).

Rule-based approaches (Andersen et al., 2013) are still considered a strong alternative to end-to-end neural networks, with many industry solutions still relying on rules defined over syntactic trees. The rule-based approach has the advantage of not requiring manual annotation, while also allowing easy access to adding and removing individual rules. On the other hand, language is continuously evolving, and there are exceptions to most grammar rules we know. Additionally, rule-based matching typically relies on syntactic pre-processing, which is error-prone, leading to compounding errors that hurt the downstream GED performance.

OUR CONTRIBUTIONS. In this work, we compare the performance of rule-based approaches and end-to-end neural models for the detection of SVA errors. We show that rule-based systems are vulnerable to errors in the underlying syntactic parsers, while also failing to capture irregularities and exceptions. In contrast, end-to-end neural architectures are limited by the available labeled examples and sensitive to the variance in these datasets. We then make the following observation: while rule-based error *detection* is severely affected by errors and irregularities in syntactic parsing, rule-based error *generation* is more robust. SVA errors can be generated without identifying subject dependency relations in advance, and changing the number of a verb almost always leads to an error. This generated data can be used as a silver standard for optimizing neural sequence labeling models. We demonstrate that a system trained on a combination of available labeled data and large volumes of silver standard data outperforms both neural and rule-based baselines by a margin on three out of four standard benchmarks, and on average achieves a new state-of-the-art on detecting SVA errors.

3.2 RELATED WORK

NEURAL APPROACHES. Recent neural approaches to GED include Rei and Yannakoudakis (2016) who argue that bidirectional (bi-) LSTMs, in particular, are superior to other RNNs when evaluated on standard ESL benchmarks for GED and give state-of-the-art results. Rei and Yannakoudakis (2017) show even better performance using a multi-task learning architecture for training bi-LSTMs that additionally predicts linguistic properties of words, such as their part of speech (PoS).

Recent studies (Gulordava et al., 2018; Kuncoro et al., 2018; Linzen, Dupoux, and Goldberg, 2016) have specifically analyzed the performance of LSTMs in learning syntax-sensitive dependencies such as SVA.

RULE-BASED APPROACHES. Cai et al. (2009) use a combination of dependency parsing and sentence simplification, as well as special handling of *wh*-elements, to detect SVA errors. Once the subject–verb relation is identified, after parsing the simplified input sentence, a PoS tagger is used to check agreement. This is similar in spirit to the rule-based baseline system used in our experiments below. Wang and Zhao (2015) use a similar approach, distinguishing between four different sentence types and using slightly different rules for each type. Their rules are, again, defined over the outputs of a dependency parser and a PoS tagger. Sun et al. (2007) use labeled data to derive rules based on dependency tree patterns.

AUTOMATIC ERROR GENERATION. Because of the scarcity of annotated datasets in GED, research has been carried out on creating artificial errors, where errors are injected into otherwise correct text using deterministic rules or probabilistic approaches using linguistic information (Felice and Yuan, 2014; Kasewa, Stenetorp, and Riedel, 2018). Studies focusing on detecting specific error types such as determiners and prepositions (Rozovskaya and Roth, 2011) or noun number (Brockett, Dolan, and Gamon, 2006) are mainly developed within the framework of automatic error generation. Recent work, expanding the detection (Rei et al., 2017) and the correction (Xie et al., 2018) tasks to all types of errors, improves the performance of neural models by training on additional artificial error data generated via machine translation methods.

MISCELLANEOUS. Recent work has also led to good performance in correcting grammatical errors (Bryant and Briscoe, 2018; Chollampatt and Ng, 2018a; Yannakoudakis et al., 2017). However, in this paper, we are interested in the task of grammatical error *detection* and we therefore compare our work to current state-of-the-art approaches

to detecting errors and do not report the performance of correction systems.

3.3 SUBJECT–VERB AGREEMENT DETECTION

Following recent work on GED (Rei and Yannakoudakis, 2016), we define SVA error detection as a sequence labeling task, where each token is simply labeled as correct or incorrect. For a given SVA error, only the verb is labeled as incorrect. Error types other than SVA are ignored, i.e., we do not correct the errors in the text and we do not attempt to predict them as incorrect.

In this paper, we only study SVA in English. We note that even for English, there is some controversy about what constitutes an SVA error. Manaster-Ramer (1987), cites this example, which has been used by some as an argument for English exhibiting cross-serial dependencies:

(2) The man and the women dance and sing, respectively.

We also note that subject–verb agreement can be more or less pervasive across languages, depending on how rich the morphology is, whether the given language exhibits *pro-drop*, and how far apart subjects and verbs are likely to occur.

3.4 SYSTEMS

3.4.1 *Rule-based system*

Typically, building a GED rule-based system is time-consuming and requires specific knowledge to deal with the multiple exceptions and irregularities of languages. Difficult cases (such as long distance subject–verb relations) are often ignored in order to ensure high precision, at the expense of the recall of the system. However, our rule-based system is not limited to the detection of simple cases of SVA errors. It relies on PoS tags and dependency relations to identify all types of SVA errors. Specifically, our rule-based system operates as follows: (i) it identifies the candidate verbs based on PoS tags;[1] (ii) for a given verb, it uses the dependency relations to find its subject;[2] (iii) the PoS tag of the verb and its subject are used to check whether they agree in number and person. We use predicted Penn Treebank PoS tags and dependency relations provided by the Stanford Log-linear PoS Tagger (Toutanova et al., 2003) and the Stanford Neural Network Dependency Parser (Chen and Manning, 2014) respectively.

1 Present tense verbs + "was" and "were".

2 The subject can be direct – attached with a `nsubj` relation – or indirect, such as when the syntactic subject is a relative pronoun, e.g., *who*, or an expletive, e.g., *there*.

3.4.2 *Neural system*

We use the state-of-the-art neural sequence labeling architecture for error detection (Rei and Yannakoudakis, 2016). The model receives a sequence of tokens $(w_1, ..., w_T)$ as input and outputs a sequence of labels $(l_1, ..., l_T)$, i.e., one for each token, indicating whether a token is grammatically correct (in agreement) or not, in the given context. All tokens are first mapped to distributed word representations, pre-trained using word2vec (Mikolov et al., 2013) on the Google News corpus. Following Lample et al. (2016), character-based representations are also built for every word using a bi-LSTM (Hochreiter and Schmidhuber, 1997) and then concatenated onto the word embedding.

The combined embeddings are then given as input to a word-level bi-LSTM, creating representations that are conditioned on the context from both sides of the target word. These representations are then passed through an additional feedforward layer, in order to combine the extracted features and map them to a more suitable space. A softmax output layer returns the probability distribution over the two possible labels (*correct* or *incorrect*) for each word. We also include the language modeling objective proposed by Rei (2017), which encourages the model to learn better representations via multi-tasking and predicting surrounding words in the sentence. Dropout (Srivastava et al., 2014) with probability 0.5 is applied to word representations and to the output from the word-level bi-LSTM. The model is optimised using categorical cross-entropy with AdaDelta (Zeiler, 2012).

3.5 DATA

3.5.1 *Data preprocessing*

As the public datasets either have their own taxonomy or they are not annotated with error types at all, we apply the error type extraction tool of Bryant, Felice, and Briscoe (2017) to automatically get error types mapped to the same taxonomy for all datasets. The tool automatically annotates parallel original and corrected sentences with error type information. When evaluated by human raters, the predicted error types were rated as "good" or "acceptable" in at least 95% of the cases. We use their publicly available tool[3] to automatically get error types for all public datasets mapped to the same taxonomy of 25 error types in total. We then set SVA errors as our target class.

3.5.2 *Test data*

We compare the rule-based and neural approaches for the task of SVA error detection on four benchmarks in the ESL domain.

3 https://github.com/chrisjbryant/errant

- **FCE**. The Cambridge Learner Corpus of First Certificate in English (FCE) exam scripts consists of texts produced by ESL learners taking the FCE exam, which assesses English at the upper-intermediate proficiency level (Yannakoudakis, Briscoe, and Medlock, 2011). We use the publicly available test set.

- **AESW**. The dataset from the Automated Evaluation of Scientific Writing Shared Task 2016 (AESW) is a collection of text extracts from published journal articles (mostly in physics and mathematics) along with their (sentence-aligned) corrected counterparts (Daudaravicius et al., 2016). We test on the combined train, development and test set.[4]

- **JFLEG**. The JHU Fluency-Extended GUG corpus (JFLEG) represents a cross-section of ungrammatical data, consisting of sentences written by ESL learners with different proficiency levels and L1s (Napoles, Sakaguchi, and Tetreault, 2017). We evaluate our models on the public test set.

- **CoNLL14**. The test dataset from the CoNLL 2014 shared task consists of (mostly argumentative) essays written by advanced undergraduate students from the National University of Singapore, and are annotated for grammatical errors by two native speakers of English (Ng et al., 2014).

3.5.3 *Training data*

ESL WRITINGS. We use the following ESL datasets as training data:

- **Lang8** is a parallel corpus of sentences with errors and their corrected versions created by scraping the Lang-8 website[5], which is an open platform where language learners can write texts and native speakers of that language can provide feedback via error correction (Mizumoto et al., 2011). It contains 1,047,393 sentences.

- **NUCLE** comprises around 1,400 essays written by students from the National University of Singapore. It is annotated for error tags and corrections by professional English instructors (Dahlmeier, Ng, and Wu, 2013a). It contains 57,151 sentences.

- **FCE train set**. We use the publicly available FCE training set, containing 25,748 sentences. A subset of 5,000 sentences was separated and used for development experiments.

4 Sentences containing special placeholders for mathematical equations, dates, etc. are filtered out.

5 http://lang-8.com/

		Rules	Bert-LM	LSTM$_{ESL}$	LSTM$_{ESL+art}$
FCE	P	43.75	66.67	71.88	**72.41**
	R	40.23	**52.87**	26.44	48.84
	$F_{0.5}$	43.00	63.36	53.49	**66.04**
AESW	P	14.82	18.36	**27.75**	19.05
	R	**49.75**	39.61	10.33	40.66
	$F_{0.5}$	17.24	20.57	20.75	**21.31**
CoNLL14	P	27.93	50.00	**54.84**	49.32
	R	31.96	35.24	17.53	**37.11**
	$F_{0.5}$	28.65	46.13	38.46	**46.27**
JFLEG	P	37.50	60.00	**73.91**	64.71
	R	**48.21**	32.14	30.91	39.29
	$F_{0.5}$	39.24	51.14	**57.82**	57.29
	$F_{0.5}$ avg.	32.03	45.30	42.63	**47.73**

Table 3.1: Performance of our systems (rule-based and LSTMs) and baselines. BERT-LM is the language model baseline.

ARTIFICIAL ERRORS. We generate artificial subject–verb agreement errors from large amounts of data. Specifically, we use the British National Corpus (BNC, BNC-Consortium et al., 2007), a collection of British English sentences that includes samples from different media such as newspapers, journals, letters or essays. Subject–verb agreement in English merely consists of inflecting 3rd person singular verbs in the present tense (and *be* in the past), which makes any text in English fairly easy to corrupt with SVA errors. We assume that the BNC data is written in correct British English. Using predicted PoS tags provided by the Stanford Log-linear PoS Tagger, we identify verbs in present tense, as well as *was* and *were* for the past tense, and flip them to their respective opposite version using the list of inflected English words (annotated with morphological features) from the Unimorph project (Kirov et al., 2016). The final artificial training set includes the sentences with injected errors (265,742 sentences), their original counterpart, and sentences where SVA errors could not be injected due to not containing candidate verbs that could be flipped (241,295 sentences).

3.6 EXPERIMENTS

THE MODELS. We compare our neural model trained on both artificially generated errors and ESL data (LSTM$_{ESL+art}$) to three baselines: a neural model trained only on ESL data (LSTM$_{ESL}$) (i.e., reflecting

the performance of current state-of-the-art approaches for GED), a language model based method (BERT-LM) and our rule-based system. In order to measure the real performance of a language model (LM) on the detection of SVA errors, we choose to use the BERT system (Devlin et al., 2019) to assign probabilities to different versions of the test sentences. Specifically, we use the pre-trained uncased BERT-Base model. We duplicate the sentences each time a corruptible verb occurs (flipping its number). The LM assigns a probability to both possible versions of the verbs. We select the version which has the highest probability, if this probability is at least 0.1[6] higher than the probability of the verb in the original sentence.

HYPER-PARAMETERS. We tune the model hyper-parameters on the FCE development set, according to the $F_{0.5}$ score. Training is stopped when $F_{0.5}$ on the FCE development set does not improve over 7 epochs. Word representations have size 300, while character representations have size 100. The word-level LSTM hidden layers have size 300 for each direction, and the character-level LSTM hidden layers have size 100 for each direction.

EVALUATION. Existing approaches are typically optimised for high precision at the cost of recall, as a system's utility depends strongly on the ratio of true to false positives, which has been found to be more important in terms of learning effect. A high number of false positives would mean that the system often flags correct language as incorrect, and may therefore end up doing more harm than good (Nagata and Nakatani, 2010). Because of this, $F_{0.5}$ is preferred to F_1 in the GED domain as it puts more weight on precision than recall. For each experiment, we report the token-level precision (P), the recall (R), and the $F_{0.5}$ scores.

3.7 RESULTS

The main results are summarized in Table 3.1. Looking at the performance of the $\text{LSTM}_{ESL+art}$ system, we see that on 3 out of 4 benchmarks, our neural model trained on artificially generated errors outperforms the LSTM_{ESL} system with respect to $F_{0.5}$. On average, over the four benchmarks, its $F_{0.5}$ score is 2.43 points higher than the best performing baseline. Both neural models obtain higher $F_{0.5}$ scores than the rule-based baseline, on average and across the board, i.e., +10.6 for LSTM_{ESL} and +15.7 for $\text{LSTM}_{ESL+Art}$. The BERT-LM outperforms the LSTM_{ESL} (mostly due to its higher recall, i.e., +18.66) but still does

6 We tune the threshold on the test dataset from the CoNLL 2013 shared task on Grammatical Error Correction of ESL learner essays.

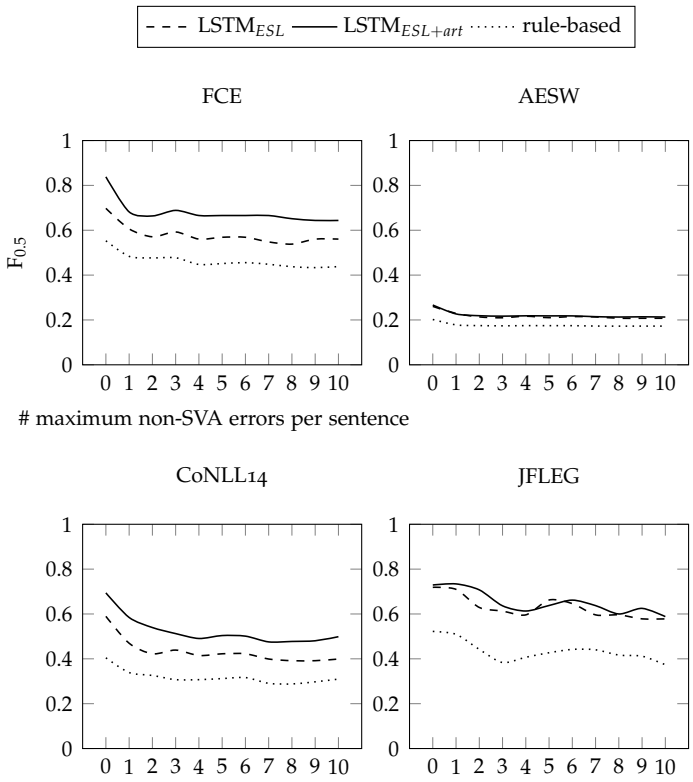


Figure 3.1: Performance ($F_{0.5}$ scores) of the systems with respect to the noise in test data (i.e., the number of additional non-SVA errors in sentences).

not reach the $F_{0.5}$ score of the LSTM$_{ESL+Art}$ system which gets higher precision and recall overall (+2.62 and +1.51 respectively).

Furthermore, we observe a trend that the two LSTM systems trade off precision and recall, with the LSTM$_{ESL}$ system yielding the highest precision across most datasets, but also yielding significantly lower recall than LSTM$_{ESL+Art}$. It is also evident that the performance varies over domains: all models struggle with AESW. This is likely due to the complexity of the scientific writing genre where, for example, sentences contain parentheses interposed between a verb and its subject. We also note errors are far less frequent in this genre, leading to moderate recall and very low precision. For the rest of the datasets, system performance is generally better.

3.8 ANALYSIS

We analyze the effect of adding artificial errors to the training data. In particular, we focus on the robustness of our models by looking at how sensitive they are to grammatical errors in the surrounding context; and by looking at how good the models are at predicting agreement relative to the distance between the subject and verb. This set of experiments is similar in spirit to Linzen, Dupoux, and Goldberg

(2016). We also analyze our rule-based baseline: so far, we know our rule-based baseline was sensitive to parser errors and irregularities. We inspect the quality of the underlying parser by evaluating it on data that resembles the data used in our experiments, to see whether errors seem to result more from parser errors or irregularities. Finally, we also look at the sensitivity of our systems to other linguistic phenomena such as relative clauses or conjunctions.

3.8.1 *Sensitivity to other errors in the surrounding context*

In ESL writings, multiple errors can occur in the same sentence. This means more variable contexts, which can lead to degradation in the performance of both syntactic parsers / rule-based systems and GED models.

TESTING ON NOISY CONTEXTS We first evaluate how our systems are impacted by additional non-SVA errors in the surrounding context of SVA errors in our test data. For each of the test datasets, we create multiple versions, allowing for n non-SVA errors per sentence (we correct the extra non-SVA errors). This way we can create datasets with different levels of complexity with respect to the grammatical errors within them.

In Figure 3.1, the $F_{0.5}$ scores of the models are shown for different numbers of grammatical errors per sentence. It is evident that all of the models are negatively affected by the presence of other errors in the same sentence. Using more data for training – i.e., our artificial training data which does not include context errors – generally boosts performance on data with and without grammatical errors in the context. In other words, training with additional artificially generated errors seems, overall, to be making our model more robust. We also note that our rule-based baseline is affected by errors to roughly the same extent as our baseline neural model is. One might have thought the rule-based baseline would suffer more, because of it being sensitive to errors in the underlying syntactic parser. We return to this issue below.

TRAINING ON NON-NOISY CONTEXTS In order to assess the benefit of training on non-erroneous contexts, we create a new dataset from our ESL training data (see §3.5.3). Based on the annotations in the data, we apply the corrections of error types other than SVA, thereby only leaving SVA errors in the data. We experiment with how adding this 'clean' dataset to the training set of our existing systems affects performance. The resulting $F_{0.5}$ scores are listed in Table 3.2. Using 'clean' sentences in addition to our original ESL data for training always positively affects performance. In this regard, as experimented in (Rei and Yannakoudakis, 2016), training on more data in the same

domain is a valid solution for improving the performance of LSTM models. However, when also adding artificially generated data to the training set, we reach higher scores only on 2 out of the 4 benchmarks. It greatly improves the average recall (+11.03), without hurting the precision on FCE and CoNLL14 but affects negatively the precision on AESW and JFLEG.

	FCE	**AESW**	**CoNLL14**	**JFLEG**
System	$F_{0.5}$	$F_{0.5}$	$F_{0.5}$	$F_{0.5}$
$LSTM_{ESL}$	53.49	20.75	38.46	57.82
$LSTM_{ESL+art}$	66.04	21.31	46.27	57.29
$LSTM_{ESL+cor}$	65.08	**27.16**	46.26	**59.52**
$LSTM_{ESL+art+cor}$	**67.16**	21.12	**52.28**	54.64

Table 3.2: Performance ($F_{0.5}$ scores) of the LSTM models when trained using an additional set of 'clean' sentences (*cor*) where non-SVA errors have been corrected.

3.8.2 *Sensitivity to long-distance dependencies*

Next, we want to study how well our models perform when the subjects and verbs are far apart, i.e., when the agreement relation is defined over a long-distance dependency. In order to see how our systems are affected by the distance between the subject and verb, we split the test sets based on different subject–verb distances.

Note, however, that our benchmarks are not annotated with PoS tags and dependency relations. If we binned our test data based on predicted dependencies, the inductive bias of our syntactic parser and the errors it made would bias our evaluation. Instead, we perform our analyses on section 22 and 23 of the Penn Treebank (PTB) dataset (Marcus, Santorini, and Marcinkiewicz, 1993). The PTB however is not annotated with grammatical errors. We therefore corrupt the sentences by injecting SVA errors, in the same way we corrupted the BNC (§3.5.3) to create additional training data.

For each sentence in the PTB, we identify a subject–verb pair, and group the sentences by the subject–verb distance. We then run our models on two versions of each sentence: an unaltered version and a corrupted one, where we have generated an SVA error by corrupting the verb, using the method described earlier (§3.5.3). This way we can compute the performance of our models as $F_{0.5}$ scores over this dataset. The results are displayed in Figure 3.2. We can see that the LSTM trained with artificial data performs significantly better on long-distance subject–verb pairs than the LSTM trained only on ESL data. This suggests that training on artificially generated errors also makes our models more robust to this potential source of error.

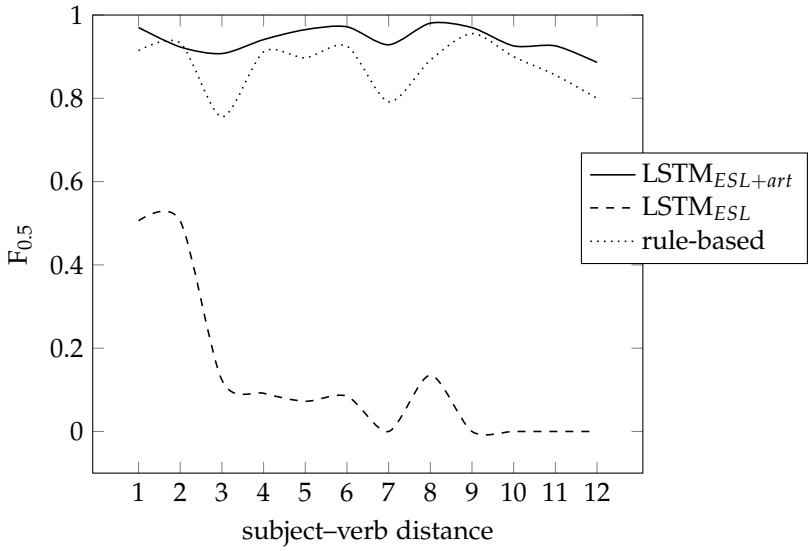


Figure 3.2: $F_{0.5}$ scores of the systems on the PTB as a function of subject–verb distance.

Note that, in general, there is a substantial gap between the performance of the two LSTM models. This is because one is trained on artificial data – similar to the data we use in our analysis. However, the conclusions are based on the relative differences in performance over long-distance dependencies, and these differences should still be comparable across the two models.

3.8.3 *Sources of error for our rule-based baseline*

There are two obvious potential sources of error for our rule-based baseline: sensitivity to errors in the underlying syntactic parsers, and sensitivity to the irregularities of language, e.g., when collective nouns or named entities are subjects, subject–verb agreement cannot always be determined by the PoS tags. We show that the main source of error seems to be irregularities by showing that the underlying syntactic parsers perform relatively well, even in the ESL domain.

Table 3.3 lists the parsing and tagging performance of our underlying syntactic parsers across three domains: learner data (ESL) and web data (EWT) from the Universal Dependencies (UD) project (Nivre et al., 2017), as well as the newswire data it was trained on (PTB). We only evaluate subject–verb relations, since these are the only ones of interest in this paper. We see that while there is a noticeable out-of-domain drop going from newswire to learner language or web data, the parser is still able to detect subject–verb relations with high precision and recall. This suggests that the vulnerability of our rule-based baseline is primarily a result of linguistic irregularities and exceptions to the implemented rules.

	UD-ESL	UD-EWT	PTB 23
Subject–verb precision	88.47	88.86	91.31
Subject–verb recall	89.37	85.11	89.84
PoS tags accuracy	96.36	93.20	97.79

Table 3.3: The Stanford PoS Tagger and Dependency Parser's performance on different treebanks. Subject–verb precision/recall relates to subject–verb relations. PoS tag accuracy is only for PoS tags of the subjects and verbs.

3.8.4 *Sensitivity to other linguistic phenomena*

Finally, manually reviewing the errors made by the rule-based system, we identified frequent linguistic sources of errors, including relative clauses, conjunctions, ambiguous PoS tags, and collective nouns. We therefore analyze how the LSTMs and the rule-based system are globally sensitive to these potential sources of error. Since our benchmarks are not annotated with PoS and dependency relations, we again use the corrupted PTB sentences (see §3.8.2).

Many of the examples in which our rule-based baseline fails include *relative clauses* (when the verb is the root of a relative clause) and *conjunctions* (when the subject is a conjunction). A second major cause of failure is ambiguous verbs, i.e., verb forms that can also be nouns (*ambiguous PoS*, e.g., "need", "stop", "point", etc.), and subjects which are singular nouns describing groups of people or things (*collective nouns*, e.g., "team", "family", "staff", etc.). The following examples illustrate these cases (underlined):

(3) a. The church and the cathedral **are** very interesting [...] (*conjunction*)
b. If there is someone who **doesn't** agree with me, he or she [...] (*relative clause*)
c. It is said that the majority of the citizens **has** got a car [...] (*collective noun*)
d. [...] and police officer **walk** around the building as well. (*ambiguous PoS*)

We evaluate our models on the PTB data and report the error rate (the lower the better) on present tense verbs (Figure 3.3). Overall, results show that all models are negatively affected when they encounter complex syntactic structures and ambiguous cases. Figure 3.3 also confirms that the rule-based baseline is the most sensitive one to complex structures. Especially in comparison with the $\text{LSTM}_{ESL+art}$ model, the rule-based system achieves good scores on verbs which are not part of complex structures, but performs significantly worse on difficult cases. The LSTM_{ESL} model is the worst across almost all

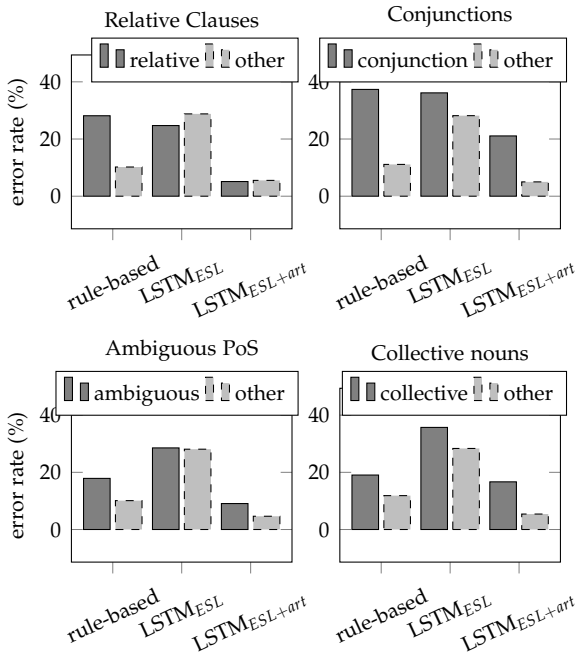


Figure 3.3: SVA error rates on the PTB data for complex syntactic structures and ambiguous cases.

cases, while the LSTM$_{ESL+art}$ shows significant improvements over the baselines, in particular for the difficult cases.

3.9 CONCLUSION

In this paper, we argue for artificial error generation as an effective approach to learning more robust neural models for subject–verb agreement detection. We demonstrate that error generation is much less sensitive to parsing errors and irregularities than rule-based systems for detecting subject–verb agreement. On the other hand, artificial error generation enables us to utilise much more training data, and therefore can develop more robust neural models for SVA error detection that do not overfit the available, manually annotated training data. Our simple approach to detecting subject–verb agreements achieves a new state of the art on three out of four available benchmarks, and, on average, is better than previous approaches on the task. We show that, in particular, models trained on large volumes of artificially generated errors become more robust to other errors in the surrounding context of SVA, long-distance dependencies, and other challenging linguistic phenomena.

ACKNOWLEDGEMENTS

This project was supported by Siteimprove and the Innovation Fund of Denmark through an industrial PhD grant. Marek Rei and Helen Yannakoudakis were supported by Cambridge Assessment, University of Cambridge.

4

NOISY CHANNEL FOR LOW RESOURCE GRAMMATICAL ERROR CORRECTION

ABSTRACT

This paper describes our contribution to the low-resource track of the BEA 2019 shared task on Grammatical Error Correction (GEC). Our approach to GEC builds on the theory of the noisy channel by combining a channel model and language model. We generate confusion sets from the Wikipedia edit history and use the frequencies of edits to estimate the channel model. Additionally, we use two pre-trained language models: 1) Google's BERT model, which we fine-tune for specific error types and 2) OpenAI's GPT-2 model, utilizing that it can operate with previous sentences as context. Furthermore, we search for the optimal combinations of corrections using beam search.

4.1 INTRODUCTION

GRAMMATICAL ERROR CORRECTION Grammatical Error Correction (GEC) is the task of automatically correcting grammatical errors in written text. The task is relevant for users producing text through text interfaces, both as assistance during the writing process and for proofreading already written work. In recent years, GEC has received increasing attention in the research community with several shared tasks on the topic, such as CoNLL 13-14 (Ng et al., 2014, 2013), HOO (Dale and Kilgarriff, 2011), and AESW (Daudaravicius et al., 2016), and most recently the BEA 2019 shared task on GEC (Bryant et al., 2019), which this work is a contribution to.

SUPERVISED GEC Current state-of-the-art approaches to GEC use a supervised machine translation setup (Ge, Wei, and Zhou, 2018; Grundkiewicz and Junczys-Dowmunt, 2018), that relies on large amounts of annotated learner data. This means that systems do not generalize well to non-learner domains and that these approaches do not work well for low-resource languages. As most existing datasets are not freely available for commercial use, the supervised approach also limits industrial uses.

UNSUPERVISED GEC In order to combat these problems, in recent years several approaches to GEC have used the concept of language modeling, which allows for training GEC systems without supervised data, and have so far given promising results. Bryant and Briscoe

(2018) uses a 5-gram language model while Makarenkov, Rokach, and Shapira (2019) uses a bidirectional LSTM-based language model. Kaili et al. (2018) fine-tunes LSTM-based language models for specific error types.

Using a language modeling approach means that we can create models that are trained unsupervised by only being based on high quality native text corpora. This means that our systems will only require a small amount of labeled data for tuning purposes. We can therefore build GEC systems for any language given enough native text.

THE NOISY CHANNEL The core idea that these language modeling approaches are using for GEC is that low probability sequences are more likely to contain grammatical errors than high probability sequences. However this formulation does not take into account the writer's likelihood of making particular errors. For example, "then" $\rightarrow$ "than" is much more common than "then" $\rightarrow$ "the" due to an underlying similarity in phonetics.

In order to take this into account we utilize the concept of the noisy channel model, which allows for modeling the users likelihood of making particular errors, instead of only relying on which sequences of words are more probable.

CONTRIBUTIONS In the following, we present our low-resource approach to GEC, which ranked as the 6th best performing system in the low-resource track of the BEA 2019 shared task. We utilize confusion sets and edit statistics gathered from the Wikipedia edit history, as well as unsupervised language models in a noisy channel setting.

Our contributions are 1) formalizing GEC in the noisy channel framework, 2) generating confusion sets from the Wikipedia edit history, 3) estimating a channel model based on frequencies of edits from the confusion sets, 4) combining existing pre-trained language models, with each their own strength, 5) specializing models for specific grammatical error types, and 6) using beam search to find the optimal combination of corrections.

4.2 THE NOISY CHANNEL

The intuition of the noisy channel model (Kemighan, Church, and Gale, 1990; Mays, J. Damerau, and Mercer, 1990) is that for any given word in a sentence, we have a true underlying word, that has been passed through a noisy communication channel, which potentially has modified the word into an erroneous surface form.

Our goal is to build a model of the channel. With this, given a confusion set, we can pass every candidate correction through this

noisy channel to see which one is most likely to have produced the surface word.

The noisy channel model can be formulated as a form of Bayesian inference. Given a potentially erroneous surface word, x, we want to find the hidden word, c^*, from all candidates $c \in C$, that generated x.

$$\hat{c} =_{c \in C} P(c|x)$$

Using Bayes' rule this can be restated as

$$\hat{c} =_{c \in C} P(x|c) * P(c)$$

where $P(x|c)$ is the likelihood of the noisy channel producing a particular x. This is referred to as the channel model. The prior probability of a hidden word, $P(c)$, is modeled by a language model (Jurafsky and Martin, 2009).

4.3 SYSTEM

Our system is a combination of several components: a PoS tagger, the channel model, two language models (BERT and GPT-2) and beam search. We first PoS tag the sentence. Then, the sentence is processed from left to right, and for every word x, we identify the set C of possible correction candidates, based on the PoS tag and our generated confusion sets. We then pick the $c \in C$ with the highest $P(c|x)$ estimated using our components in the following formula:

$$P(c|x) = P_{Channel} * P_{BERT} * P_{GPT-2}$$

We allow the system to consider multiple hypotheses by using beam search, which continuously keeps track of a beam of the most likely hypotheses.

In the following, we describe the different components that make up our GEC system in more detail.

4.3.1 *Channel model*

We estimate the channel model in two ways, depending if the written word is in our vocabulary (real-word error) or not (non-word error).

REAL-WORD ERRORS In order to estimate the channel model $P(x|c)$ for real-word errors, we first make a simplifying assumption that a human only makes a mistake for 1 in 20 words. This means that there is a 5% probability (denoted as α) of the surface word x being wrong. This probability can be distributed between the candidate corrections taken from the confusion set. For a given candidate word c_i we can calculate the channel probability using frequency counts of edits for

all candidates in C. We gather frequency counts from the Wikipedia edit history (§ 4.4.1).

$$P(x|c_i) = \alpha * \frac{|x \rightarrow c_i|}{\sum_{j=1}^{|C|} |x \rightarrow c_j|}$$

NON-WORD ERRORS For non-word errors we assume that any x not in our vocabulary and not a named entity[1] is an error. Assuming a list of candidate corrections, we use the inverse Levenshtein distance to distribute the error probability between the candidates. Hereby, candidates which are lexically closer to the original word are made more likely.

4.3.2 *Language models*

For language modeling we use a combination of two pre-trained models that have recently given good results: BERT (Devlin et al., 2019) and GPT-2 (Radford et al., 2019).

BERT BERT is a Transformer-based (Vaswani et al., 2017) language model pre-trained on a large text corpus. It estimates probabilities by jointly conditioning on both left and right context. We use the pre-trained BERT-Base Uncased model as a starting point for several models, which are each fine-tuned for specific error types on sentences extracted from a Wikipedia dump. We do three types of fine-tuning, using the default hyperparameters of BERT.

- PoS-based fine-tuning, where a word is removed and the model predicts its PoS tag. This is used to classify which word category should be at the position for verb form errors and noun number errors.
- Word-based fine-tuning, where a word is removed and the model predicts the word from a vocabulary of the most common 40.000 words from the Wikipedia dump. This is used to estimate probabilities for words in our confusion sets.
- Comma prediction, where we remove all commas and let the model predict where to insert commas. Any discrepancies between the produced and original sentence is used as comma edits, if the model is more than 95% certain.

GPT-2 GPT-2 is another Transformer-based language model trained on a dataset of 8 million web pages. GPT-2 only looks at the previous context to estimate probabilities. We take advantage of the fact that

1 as estimated by Spacy, https://spacy.io

GPT-2 is trained using previous sentences as context by including the previous sentence when estimating probabilities.

4.3.3 *Beam search*

Since our error correction models make a decision separately for every word, sometimes conflicting corrections can be made, e.g., "the cats is big." might be corrected to "the cat are big". Therefore we utilize beam search in order to efficiently explore combinations of corrections in order to find the optimal output sentence. We utilize a beam width of 3.

4.4 CONFUSION SETS

The first step in correcting a sentence is to identify the potentially erroneous tokens (or groups of tokens) and determine a set of possible corrections for each. We use several methods for deducing these confusion sets according to different error types.

4.4.1 *Wikipedia edit history*

We utilize the WikEd Error Corpus (Grundkiewicz and Junczys-Dowmunt, 2014) generated from Wikipedia revision histories to create confusion sets. We only retain edits of sentences where only a single word has been changed. We first end up with a list of confused token pairs which includes all types of edits, i.e., semantic or grammatical. We set up a set of rules to filter the edits not adapted to the task (e.g., the semantic replacements), and infrequent ones. We thus remove confusion pairs which define: (i) the replacement of a verb form (e.g., tense/subject–verb agreement errors); (ii) noun number errors; (iii) replacement of numbers or dates; (iv) synonyms and antonyms (using Wordnet[2] (Miller, 1995)); (v) replacement of pronouns with determiners; (vi) insertion/deletion of content words (e.g., nouns) and numbers; (vii) spelling errors.

We end up with a list of 348 edit pairs and their corresponding frequency counts in the WikedEd Error Corpus (ranging from 741 to 60,184 instances). The list includes, for instance, determiner replacements (e.g., "a"→"an") and frequently confused tokens (e.g. "to"→"too"). It covers most replacement error types but mostly closed-class words replacements such as R:DET or R:PREP.

2 https://wordnet.princeton.edu/

4.4.2 *Misspelled words*

Given a misspelled word (which we refer as non-word in the channel model) we use the Enchant library[3] to derive a set of suggestions for corrections. It mostly covers the R:SPELL error type but can also include other replacement types (such as content word replacements).

4.4.3 *Specialized models*

For fine-tuned models on specific error types, we define specific rules (mainly based on Part-of-Speech tags) to detect the corresponding tokens and their possible replacements. We use the Spacy[4] library to PoS-tag the sentences.

NOUN NUMBER MODEL We detect the nouns by their PoS-tags: NN (singular) and NNS (plural) and use a list of matching singular/plural nouns derived from Wiktionary[5] to suggest a correction. It covers the R:NOUN:NUM and R:NOUN:INFL error types.

VERB FORMS MODEL We detect all forms of verbs through their PoS-tags and derive a list of potential corrections (i.e., all possible inflections) using the list of English verb inflections from the Unimorph project (Kirov et al., 2016). Here, we mainly cover the R:VERB:FORM and R:VERB:SVA error types but also cases of R:VERB:INFL and R:VERB:TENSE error types.

4.5 DISCUSSION

4.5.1 *Results*

Results on the BEA 2019 shared task test dataset are listed per edit and error type in Table 4.1. It is evident, that out approach deals with a wide array of error types, but with varying quality. The model performs particularly well on spelling errors, subject–verb agreement errors and inserting missing commas. However, the model performs rather poorly on the replacement of adjectives, adverbs and conjunctions which are based on confusion sets derived from Wikipedia edits suggesting that more filtering would be necessary.

3 Wrapper for various spell checker engines.
4 https://spacy.io/
5 https://www.wiktionary.org/

Error type	#	P	R	$F_{0.5}$
M:PUNCT	422	80.10	38.15	65.66
R:ADJ	24	12.50	4.17	8.93
R:ADV	17	33.33	5.88	17.24
R:CONJ	5	2.22	20.00	2.70
R:DET	129	20.48	52.71	23.34
R:MORPH	128	46.15	18.75	35.71
R:NOUN	70	50.00	8.57	25.42
R:NOUN:INFL	19	42.86	31.58	40.00
R:NOUN:NUM	290	43.79	68.31	47.18
R:ORTH	349	10.20	1.43	4.59
R:OTHER	618	20.43	6.15	13.95
R:PART	15	38.89	46.67	40.23
R:PREP	292	39.49	58.56	42.24
R:PRON	50	34.15	56.00	37.04
R:SPELL	321	76.51	75.08	76.22
R:VERB	134	25.00	2.99	10.10
R:VERB:FORM	169	47.96	55.62	49.32
R:VERB:INFL	7	100.00	85.71	96.77
R:VERB:SVA	146	74.39	83.56	76.06
R:VERB:TENSE	160	42.50	10.62	26.56
U:PUNCT	118	34.90	88.14	39.69
All error types	4498	44.52	28.88	40.17

Table 4.1: Span-level correction results of our system. We do not show results for the error types we do not predict.

4.5.2 *Ablation analysis*

We do an ablation analysis of the different components of our model to see how each part contributes to the performance. The global results are shown in Table 4.2. Detailed results per error type are shown in Appendix A.1 for all models.

BEAM SEARCH removing the beam search results in a considerable drop in $F_{0.5}$ by 2.73. This shows that figuring out how to optimally combine multiple local edits is important.

GPT-2 removing GPT-2 results in the largest drop in $F_{0.5}$ score of 5.09. The drop is large for most error types but the ablation is especially damaging on the precision of verb form errors.

BERT dropping BERT results in a 1.11 drop in $F_{0.5}$ score. This indicates that GPT-2 is pulling most of the weight.

CHANNEL MODEL we ablate the channel model by dividing out probabilities by uniform distribution over the candidates instead of using the frequency counts of the confusion sets and reverse Levenshtein distance. It results in a drop in $F_{0.5}$ score by 0.44.

	P	**R**	**$F_{0.5}$**
Chan + BERT + GPT	40.29	29.19	37.44
Chan + BERT + beam	37.03	28.98	35.08
Chan + GPT + beam	42.31	29.89	39.06
BERT + GPT + beam	43.50	29.49	39.73
Chan + BERT + GPT + beam	44.52	28.88	40.17

Table 4.2: Span-level correction results of the ablated models.

4.6 CONCLUSIONS

In this work we have presented our system for the BEA 2019 shared task on Grammatical Error Correction, which ranked as the 6th best in the low resource track.

Our ablation analysis showed that each of the components of our system has a positive effect on the overall performance, meaning that the combination of all of our components leads to the best score.

Future work could explore using more advanced channel models, such as using phonetic features to determine the similarity of words. Furthermore our approach could also be adapted to handle insertions and deletions. Additionally, there are several parameters that could be tuned for better performance, including for example, α, the probability that the channel inserts an error, and the beam width.

ACKNOWLEDGEMENTS

This project was supported by Siteimprove and the Innovation Fund of Denmark through an industrial PhD grant.

Part III

DOMAIN GENERALIZATION

5

GRAMMATICAL ERROR CORRECTION IN LOW ERROR DENSITY DOMAINS: A NEW BENCHMARK AND ANALYSES

ABSTRACT

Evaluation of grammatical error correction (GEC) systems has primarily focused on essays written by non-native learners of English, which however is only part of the full spectrum of GEC applications. We aim to broaden the target domain of GEC and release CWEB, a new benchmark for GEC consisting of website text generated by English speakers of varying levels of proficiency. Website data is a common and important domain that contains far fewer grammatical errors than learner essays, which we show presents a challenge to state-of-the-art GEC systems. We demonstrate that a factor behind this is the inability of systems to rely on a strong internal language model in low error density domains. We hope this work shall facilitate the development of open-domain GEC models that generalize to different topics and genres.

5.1 INTRODUCTION

Grammatical error correction (GEC) is the task of automatically editing text to remove grammatical errors; for example: [*A link to registration can also be found ~~at~~* ***on*** *the same page.*]. GEC systems so far have primarily focused on correcting essays produced by English-as-a-second-language (ESL) learners, providing fast and inexpensive feedback to facilitate language learning. However, this is only one target domain in the full spectrum of GEC applications. GEC models can also help to improve written communication outside of the formal education setting. Today the largest medium of written communication is the internet, with approximately 380 new websites created every minute.[1] Ensuring grammatical correctness of websites helps facilitate clear communication and a professional commercial presentation. Therefore, it is important that GEC models perform well in the open-domain setting and generalize, not only to writing produced in the educational context, but also to language production "in the wild". Website data specifically represent a broad and diverse range of writing and constitute a major part of what people read and write on an everyday basis.

1 https://www.millforbusiness.com/how-many-websites-are-there/

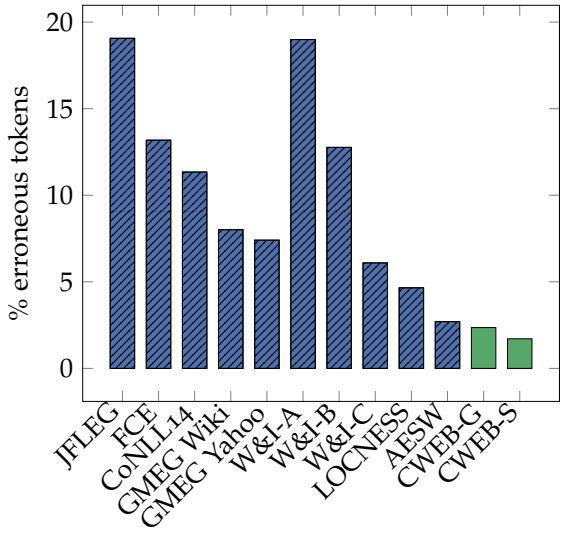


Figure 5.1: Percentage of erroneous tokens per domain. CWEB-G/S are our newly devised datasets.

This work highlights two major prevailing challenges of current approaches to GEC: *domain adaptation* and *low precision* in texts with low error density. Previous work has primarily targeted essay-style text with high error density (see Figure 5.1); however, this lack of diversity means that it is not clear how systems perform on other domains and under different error distributions (Sakaguchi, Napoles, and Tetreault, 2017).[2]

Current publicly available datasets are restricted to non-native English essays [e.g. FCE (Yannakoudakis, Briscoe, and Medlock, 2011); CoNLL14 (Ng et al., 2014)], student essays [W&I+LOCNESS (Bryant et al., 2019; Granger, 1998)] or target a specific domain [scientific writing; AESW (Daudaravicius et al., 2016)]. Supervised systems trained on specific domains are less likely to be as effective at correcting distinctive errors from other domains, as is the case for systems trained on learner data with different native languages (Chollampatt, Hoang, and Ng, 2016; Nadejde and Tetreault, 2019). The recent BEA 2019 shared task (Bryant et al., 2019) encouraged research in the use of low-resource and unsupervised approaches; however, evaluation primarily targeted the restricted domain of student essays. We show that when applied to data outside of the language learning domain, current state-of-the-art systems exhibit low precision due to a tendency to over-predict errors. Recent work tackled the domain adaptation problem, and released GEC benchmarks from Wikipedia data and online comments [GMEG Wiki+Yahoo (Napoles, Nădejde, and Tetreault, 2019)]. However, these datasets present a high density of errors and represent a limited subset of the full distribution of errors in online writing.

2 Leacock et al. (2010) highlighted the variations in the distribution of errors in non-native and native English writings.

		CWEB-S	CWEB-G	Total
Dev	sent.	2,862	3,867	6,729
	tokens	68,857	79,689	148,546
	edits	895	1595	2490
Test	sent.	2,864	3,981	6,845
	tokens	68,459	80,684	149,143
	edits	1004	1679	2683
Total	sent.	5,726	7,848	13,574
	tokens	137,316	160,373	297,689
	websites	453	625	1,078
	parag.	659	630	1,289

Table 5.1: Distribution of sentences and tokens in the CWEB dataset.

Contributions: We (i) release a new dataset, CWEB (**C**orrected **Web**sites), of website data that is corrected for grammatical errors;[3] (ii) systematically compare it to previously released GEC corpora; (iii) benchmark current state-of-the-art GEC approaches on this data and demonstrate that they are heavily biased towards existing datasets with high error density, even after fine-tuning on our target domain; (iv) perform an analysis showing that a factor behind the performance drop is the inability of systems to rely on a strong internal language model in low error density domains.

We hope that the new dataset will contribute towards the development of robust GEC models in the open-domain setting.

5.2 CWEB DATASET

We create a new dataset of English texts from randomly sampled websites, and annotate it for grammatical errors. The source texts are randomly selected from the first 18 dumps of the CommonCrawl[4] dataset and represent a wide range of data seen online such as blogs, magazines, corporate or educational websites. These include texts written by native or non-native English speakers and professional as well as amateur online writers.

TEXT EXTRACTION To ensure English content, we exclude websites with country-code top-level domains; e.g., .fr, .de. We use the `jusText`[5] tool to retrieve the content from HTML pages (removing boilerplate

3 https://github.com/SimonHFL/CWEB
4 https://commoncrawl.org/
5 https://github.com/miso-belica/jusText

	# sents	type -token	tok /sent	err. sents (%)	edits /sent	# annotators	sent-$\mathcal{K}$	NEs /sents
JFLEG	747	0.44	18.9	86.4	3.6	4	0.53	0.35
FCE	2,695	0.39	15.6	67.8	2.6	1	-†	0.59
CoNLL14	1,312	0.39	22.9	75.8	2.7	2	0.25	0.31
W&I-A	1,036	0.43	18.0	80.5	3.6	1	-†	0.58
W&I-B	1,285	0.45	18.4	72.1	2.7	1	-†	0.52
W&I-C	1,068	0.47	20.1	53.8	1.9	1	-†	0.78
LOCNESS	988	0.47	23.4	52.2	1.8	1	-†	0.77
GMEG wiki	992	0.55	26.9	82.3	2.5	4	0.43	2.83
GMEG yahoo	1,000	0.46	16.9	50.5	2.7	4	0.51	0.59
AESW	52,124	0.52	23.9	36.1	1.6	1	-†	0.93
CWEB-S	2,864	0.56	23.9	24.5	1.5	2	0.39	1.44
CWEB-G	3,981	0.53	20.3	25.6	1.9	2	0.44	1.04

Table 5.2: Statistics on GEC Corpora; type–token is the average ratio of vocabulary size by the total number of tokens (calculated as an average over a sliding window of 1,000 tokens); ratio of edits per sentence is calculated on erroneous sentences; sent-$\mathcal{K}$ is sentence-level Cohen's Kappa score (†: calculated for datasets with > 1 annotator); NEs stands for Named Entities (extracted using Spacy).

elements and splitting the content into paragraphs). We heavily filter the data by removing paragraphs which contain non-English[6] and incomplete sentences. To ensure diversity of the data, we also remove duplicate sentences. Among the million sentences gathered, we select paragraphs randomly.

We split the data with respect to where they come from: sponsored[7] (CWEB-S) or generic[8] (CWEB-G) websites. The sponsored data represent a more focused domain (professional writing) than the generic one which includes writing from various proficiency levels.

ANNOTATION The data is corrected for errors by two expert annotators, trained for correcting grammatical errors in English text: not attempting to rewrite the text nor make fluency edits, but rather to make minimal edits – minimum number of edits to make the text grammatical. During error annotation, the annotators have access to the entire paragraph in which a sentence belongs, therefore using the context of a sentence to help them in the correction. Examples of

6 Using the `langdetect` package.
7 top-level domains: .gov, .edu, .mil, .int, and .museum.
8 top-level domains: .com, .info, .net, .org.

Error type	Example sentence
VERB:SVA	They develop positive relationships with swimmers and members, and ~~promotes~~ **promote** programs in order to generate more participation.
MORPH / ORTH	In a small ~~agriculture~~ **agricultural** town on the east side of Washington ~~state~~ **State** called Yakima.
PREP	[...] the distance between the two should be ~~on~~ **of** the order of 50 microns.

Table 5.3: Example sentences from the CWEB dataset. Erroneous text is struck through and corrections are in bold.

erroneous sentences from our data are shown in Table 5.3. Annotator agreement is calculated at the sentence level using Cohen's Kappa, i.e. we calculate whether annotators agree on which sentences are erroneous. This approach is preferable to relying on exact matching of error corrections, as as there are often many different ways to correct a sentence (Bryant and Ng, 2015). Kappa is 0.39 and 0.44 for sponsored (CWEB-S) and generic website (CWEB-G) data respectively, and Table 5.2 presents how our agreement results compare to those of existing GEC datasets. The table also includes a number of other statistics, and the different datasets are further analyzed, compared and contrasted in Section 5.5.

The texts are tokenized using SpaCy[9] and automatically labeled for error types (and converted into the M2 format) using the ERRor ANnotation Toolkit (ERRANT) (Bryant, Felice, and Briscoe, 2017).

RELEASE For each dataset, we release a development and a test set: we propose a roughly equal division of the data into the two splits, which presents a fair amount of errors to evaluate on (see Table 5.1).

To avoid copyright restrictions, we split the collected paragraphs into sentences and shuffle all sentences in order to break the original and coherent structure that would be needed to reproduce the copyrighted material. This approach has successfully been used in previous work for devising web-based corpora (Biemann et al., 2007; Schäfer, 2015). The data is available at https://github.com/SimonHFL/CWEB.

5.3 GEC CORPORA

We compare our data with existing GEC corpora which cover a range of domains and proficiency levels. Table 5.2 presents a number of different statistics and Table 5.4 their error-type frequencies.[10]

9 https://spacy.io/

10 See links to downloadable versions in Appendix A.2

	Punct	Verb	Other	Det	Noun	Prep	Spell	All
JFLEG	147.7	233.5	295.6	180.7	167.7	107.1	242.5	1675.6
FCE 2.1	112.3	176.7	138.3	149.1	105.4	113.8	107.8	1084.9
CoNLL14	65.5	200.5	158.1	134.9	116.8	92.7	26.0	919.6
W&I-A	244.8	300.0	237.3	159.1	139.8	137.2	79.3	1561.2
W&I-B	188.2	202.5	136.7	124.1	89.0	114.4	36.3	1050.7
W&I-C	100.4	79.4	57.4	65.8	49.9	64.9	16.3	504.1
LOCNESS	152.3	19.9	43.3	16.4	32.4	28.1	51.0	400.6
GMEG Wiki	230.0	48.1	93.8	40.3	63.6	37.1	86.9	732.3
GMEG Yahoo	194.0	24.2	98.0	22.6	26.2	21.1	68.0	635.3
AESW	80.6	17.8	42.7	33.7	16.8	11.4	5.1	239.2
CWEB-G	48.9	23.4	31.6	20.9	19.6	15.6	3.8	208.9
CWEB-S	48.7	13.1	21.0	19.7	12.8	9.8	2.4	147.2

Table 5.4: Number of error occurrences for the most frequent error types (per 10,000 token).

5.3.1 *English as a second language (ESL)*

JFLEG (Napoles, Sakaguchi, and Tetreault, 2017) The JHU Fluency-Extended GUG corpus consists of sentences written by English language learners (with different proficiency levels and L1s) for the TOEFL® exam, covering a range of topics. Texts have been corrected for grammatical errors and fluency.

FCE (Yannakoudakis, Briscoe, and Medlock, 2011) consists of 1,244 error corrected texts produced by learners taking the First Certificate in English exam, which assesses English at an upper-intermediate level. We use the data split made available for the BEA GEC shared task 2019 (Bryant et al., 2019).

CONLL14 (Ng et al., 2014) consists of (mostly argumentative) essays written by ESL learners from the National University of Singapore, which are annotated for grammatical errors by two native speakers of English.

WRITE&IMPROVE (W&I) (Bryant et al., 2019) Cambridge English Write & Improve (Yannakoudakis et al., 2018) is an online web platform that automatically provides diagnostic feedback to non-native English-language learners, including an overall language proficiency score based on the Common European Framework of Reference for

Languages (CEFR).[11] The W&I corpus contains 3,600 texts across 3 different CEFR levels – A (beginner), B (intermediate), and C (advanced) – that have been annotated for errors.[12]

5.3.2 *Other corpora*

LOCNESS (Bryant et al., 2019; Granger, 1998) The LOCNESS corpus consists of essays written by native English students. A sample of 100 essays has been annotated for errors with a 50:50 development/test split.[13]

GMEG WIKI (Napoles, Nădejde, and Tetreault, 2019) is devised based on edits in the Wikipedia revision history, and the writing therefore represents formal articles. Note that collecting sentences based on edits in the Wikipedia revision history introduces a substantial bias.[14] This means that evaluation results on this benchmark are not truly representative of how a system would perform when applied to realistic online data and full-length articles.

GMEG YAHOO (Napoles, Nădejde, and Tetreault, 2019) comprises paragraphs of informal web posts gathered from answers in the *Yahoo! Answers* platform. The style is informal, and contains slang terms and non-conventional mechanics.

AESW (Daudaravicius et al., 2016) was released as part of the Automated Evaluation of Scientific Writing Shared Task. It is a collection of text extracts from published journal articles (mostly in physics and mathematics) along with their (sentence-aligned) corrected counterparts.[15]

5.4 SYSTEM PERFORMANCE

We evaluate performance on GEC benchmarks for two approaches to GEC that currently have state-of-the-art performance on CoNLL14. The first approach, that we refer to as GEC-PSEUDODATA and is proposed by Kiyono et al. (2019),[16] uses a transformer-based seq2seq model. The second approach uses the PIE system (Awasthi et al.,

11 https://www.cambridgeenglish.org/exams-and-tests/cefr/

12 Since error corrections on test sets are not publicly available, we carry out our analyses on the development sets.

13 See footnote 12.

14 Sentences that have been edited are more likely to contain grammatical errors, and grammatical errors will therefore be over-represented. This is reflected in the 82.3% erroneous sentence rate (see Table 5.2).

15 We exclude sentences that use AESW's normalization scheme (e.g. citations replaced with __CITE__), as the models we use are not trained with these special tokens.

16 www.github.com/butsugiri/gec-pseudodata; We use the PRETLARGE+SSE (fine-tuned) model.

	GEC-pseudodata system			PIE system		
	P	**R**	**$F_{0.5}$**	**P**	**R**	**$F_{0.5}$**
JFLEG	55.73	38.73	51.13	51.04	35.21	46.74
FCE 2.1	55.11	41.61	51.75	49.55	36.34	46.19
CoNLL14	44.96	29.03	40.35	43.47	27.93	38.95
W&I-A	54.89	37.92	50.38	50.24	36.10	46.59
W&I-B	54.86	35.14	49.32	49.12	31.20	44.06
W&I-C	44.53	32.04	41.31	39.12	27.13	35.94
LOCNESS	47.09	34.13	43.77	32.77	23.11	30.24
GMEG Wiki	52.81	23.02	41.89	44.71	19.66	35.58
GMEG Yahoo	37.57	32.26	36.00	33.08	26.97	31.29
AESW	14.05	13.24	13.88	8.78	9.67	8.94
CWEB-G	21.34	23.00	21.58	14.29	18.91	14.98
CWEB-S	17.27	15.75	16.91	5.73	8.78	6.15
CWEB-G+S	19.97	20.28	19.98	10.80	15.11	11.43

Table 5.5: Scores of two SOTA GEC systems on each domain. For both systems performance is substantially lower on CWEB than ESL domains. Scores are calculated against each individual annotator and averaged

2019)[17] which leverages a BERT-based architecture for local sequence transduction tasks. Both models are pre-trained on synthetic errors and fine-tuned on learner data from the train section of FCE (Yannakoudakis, Briscoe, and Medlock, 2011), Lang-8 (Mizumoto et al., 2011), and NUCLE (Dahlmeier, Ng, and Wu, 2013a) and for GEC-PSEUDODATA additionally on the W&I train split (Bryant et al., 2019).

Performance is evaluated using the $F_{0.5}$ metric calculated by ERRANT (Bryant, Felice, and Briscoe, 2017).[18] However, the more annotators a dataset has, the higher score a system will get on this data (Bryant and Ng, 2015). In order to perform a fair comparison of systems across datasets with a different number of annotators, we calculate the ERRANT score against each individual annotator and then take the average to get the final score.

Evaluation results are presented in Table 5.5. Across all datasets, we observe lower scores with the PIE system (-6.05 $F_{0.5}$ on average), while GEC-PSEUDODATA is consistently better. Overall $F_{0.5}$ ranges from around 30 to 52 for most datasets; however, when the models

17 www.github.com/awasthiabhijeet/PIE
18 www.github.com/chrisjbryant/errant

	P	R	$F_{0.5}$
CWEB-G	42.09	16.56	32.01
CWEB-S	35.91	12.96	26.46
CWEB (G+S)	39.89	15.2	30.0

Table 5.6: Scores of the GEC-PSEUDODATA system fine-tuned on CWEB data. Fine-tuning yields substantial improvements, but scores are still worse than on ESL domains. Scores are calculated against each individual annotator and averaged.

are evaluated on CWEB and AESW, we observe a substantial drop in performance, with the lowest $F_{0.5}$ score being the PIE system on CWEB-S (6.15). Precision, in particular, suffers due to the systems over-correcting sentences that should remain unchanged.

Using the GEC-PSEUDODATA system, on average, we find a higher $F_{0.5}$ on ESL corpora (JFLEG, FCE, CoNLL, W&I) compared to non-ESL ones (47.4 vs. 29.0). This demonstrates that GEC systems trained on language learning data do not perform as well on other domains and further work is needed to improve their generalization.

5.4.1 *Fine-tuning*

We investigate the extent to which the GEC-PSEUDODATA system can be adapted to our domain, and fine-tune it using our development sets.[19] We take 1,000 sentences from each of the development sets of CWEB-G and CWEB-S and use them as a development set for this experiment. The remaining 4,729 sentences of our development sets are used as training data for fine-tuning the GEC system.

In Table 5.6, we can see that fine-tuning substantially improves performance (around +10.0 $F_{0.5}$ across all CWEB sets). In particular, precision is improved (+20.8/+18.6 on CWEB-G/S) at the expense of recall (−6.4/−2.8 on CWEB-G/S). However, performance is still low compared to the language learning domain ($F_{0.5}$ of at least 41), further indicating that there is scope for developing more robust and general-purpose, open-domain GEC systems. For the purpose of future benchmarking, Appendix A.3 lists the system's ERRANT scores based on both annotators – as opposed to the average of individual annotator scores reported in Table 5.6.

5.5 ANALYSIS

In order to assess the impact our new dataset can have on the GEC field, we carry out analyses to show 1) to what degree the domain of our data is different from existing GEC corpora, and how existing GEC

19 We use the fine-tuning parameters of Kiyono et al. (2019).

systems are affected by the domain shift; and 2) that a factor behind the performance drop on CWEB data is the inability of systems to rely on a strong internal language model in low error density domains.

5.5.1 *Domain shift*

Moving from error correction in learner texts to error correction in diverse, online texts, many of which are written by professional writers, amounts to a drift in data distribution. In general, distributional drift comes in different flavors; given two distributions $P(\mathbf{X}, \mathbf{Y})$ and $Q(\mathbf{X}, \mathbf{Y})$:

COVARIATE SHIFT concerns change in the marginal distribution of the independent variable, i.e., $P(\mathbf{X}) \neq Q(\mathbf{X})$. In the context of grammatical errors, this refers to the degree to which the type of sentences written varies between domains. Table 5.2 clearly shows covariate shift effects: see, for example, differences in vocabulary variation (measured by the type–token ratio) and the frequency of named entities.

LABEL BIAS describes the change in distribution of the dependent variable, i.e., $P(\mathbf{Y}) \neq Q(\mathbf{Y})$. In terms of GEC, this refers to the difference in error distributions across domains. In Table 5.2, we can see that CWEB data contains errors that are substantially more sparse than other domains – a smaller proportion of sentences are erroneous, and these erroneous sentences also contain fewer edits compared to other domains. Additionally, looking at Table 5.4, we can see that almost all error types are substantially less frequent in our data than in existing benchmarks – for example, spelling errors are 38 times more prevalent in GMEG Wiki compared to CWEB-S.

Moving from learner text to web data involves both forms of drift: covariate shift and label bias. We further analyze the effects of these shifts on system performance.

5.5.1.1 *Impact of error density*

To demonstrate that the error density of corpora has a substantial impact on the performance of GEC systems, we vary the proportion of erroneous sentences in each dataset by either removing correct sentences or by adding correct sentences of the same domain.[20] By fixing the frequency of errors across datasets, we can observe, in isolation, how the systems are affected by co-variate shift across domains.

20 For each dataset, we apply the gold corrections on incorrect sentences, creating new examples of in-domain, correct sentences, which are then randomly selected for inclusion.

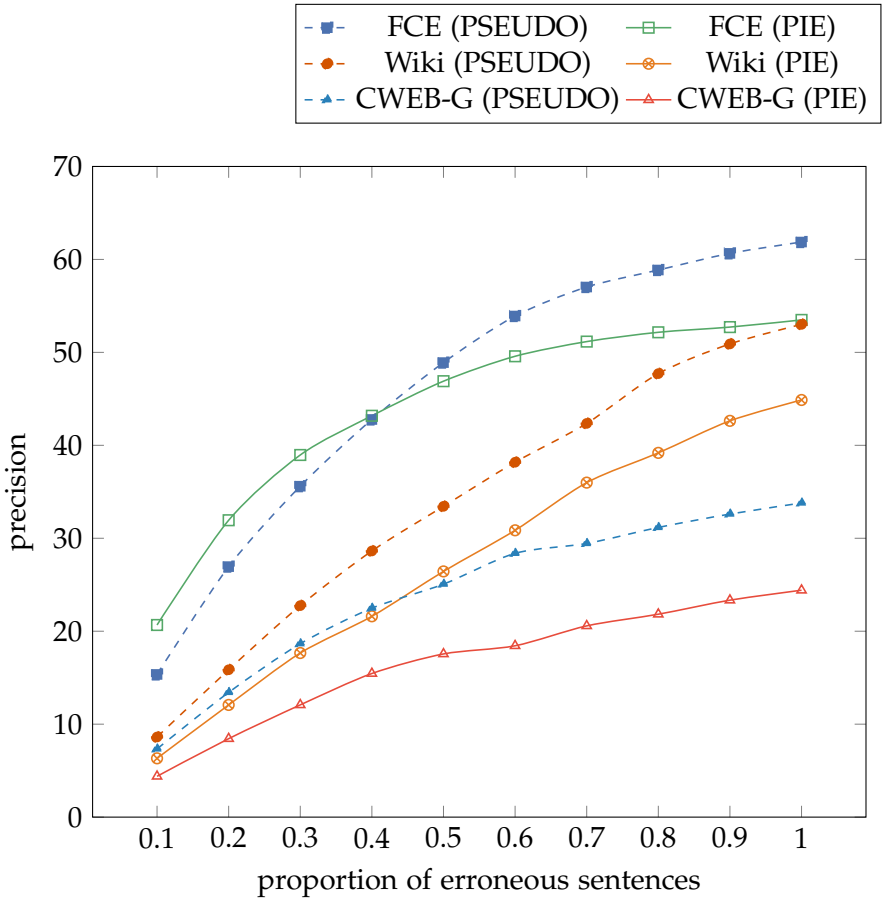


Figure 5.2: Precision as a function of the proportion of erroneous sentences in 3 different domains; comparing the GEC-PSEUDODATA (PSEUDO) and PIE systems.

Precision as a function of the proportion of erroneous sentences for selected datasets[21] is presented in Figure 5.2 (recall is unchanged).

For each domain, we observe precision being highly sensitive to the proportion of errors. This indicates that differences in error distribution across domains (i.e. label bias) is likely to be a large contributor to performance drop. We also observe the effect of covariate shift across the datasets: while the percentage of erroneous sentences is the same, precision differs for the different datasets which suggests that covariate shift across domains has an impact on the performance of the system.

5.5.1.2 *Analysis of gold edits*

In addition to error density, the type of errors present in the dataset also has an impact on the performance of GEC systems. We investigate how errors and their corresponding corrections differ across domains. In particular, we look at how gold edits in different domains change the sentence in terms of two factors: 1) How much do edits change

21 Scores for all datasets can be found in Appendix A.5.

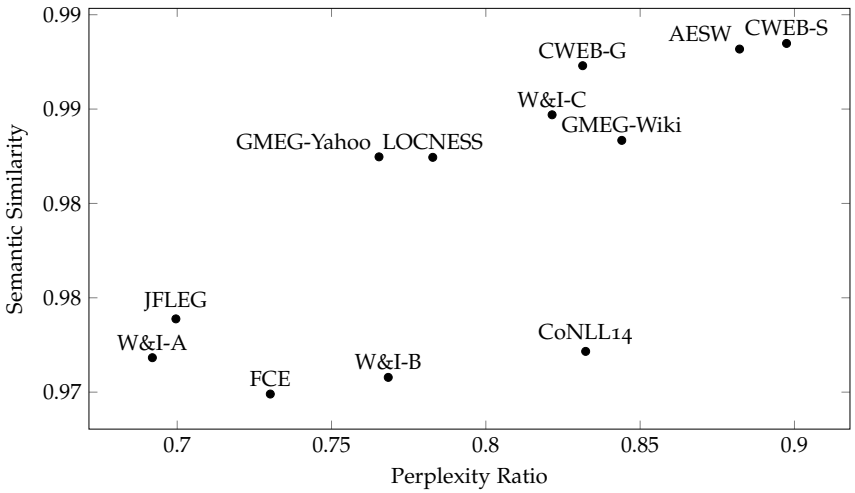


Figure 5.3: Average semantic similarity and perplexity ratio (sentence improvement) of sentences before and after being edited, plotted per dataset. The analysis is limited to sentences containing exactly one edit.

the semantics of the sentence, and 2) to what degree do edits improve the sentence.

We limit our analysis to sentences containing exactly one edit, as we are interested in how individual edits change a sentence, regardless of how domains differ in amounts of erroneous sentences and in the number of edits per sentence (Table 5.2).

Regarding 1), to measure the semantic change of a sentence after an edit is introduced, we use sentence embeddings generated by Sentence-BERT (Devlin et al., 2019) and calculate the cosine similarity between the original sentence and its corrected counterpart. Regarding 2), the degree of sentence improvement is calculated as the ratio of the perplexity of GPT-2 (Radford et al., 2019) on a sentence after and before it has been edited.

$$\Delta P = \frac{PPL(edited\_sentence)}{PPL(original\_sentence)}$$

A lower ratio suggests that the edited sentence is an improvement, since its perplexity is lower than the original sentence.

Using the outputs of machine learning models as a proxy for semantic change and sentence improvement inevitably introduces biases, but nevertheless provide valuable insights into domain differences.

CORPUS LEVEL In Figure 5.3, the average semantic similarity and perplexity ratio is plotted for each dataset. It is evident that ESL datasets consist of edits with a higher degree of semantic change and sentence improvements than datasets from more advanced speakers. CWEB and AESW in particular stand out, with edits that largely

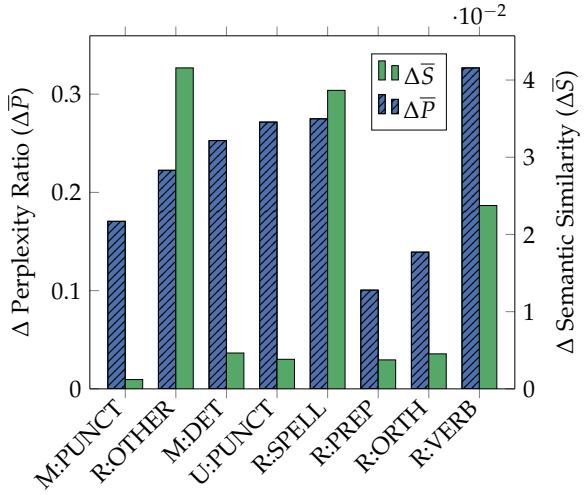


Figure 5.4: Difference in semantic similarity and perplexity ratio between CWEB-S and FCE for the most frequent error types (M: missing; R: replace; U: unnecessary).

retain the semantics of a sentence and that result in more subtle improvements.

ERROR TYPE LEVEL In order to gain further insight on what is driving the differences between datasets, we look separately at how edits of each error type change the sentence. We compare FCE and CWEB-S, which lie at opposite ends in Figure 5.3. For each dataset, we obtain an average of semantic similarity, $\overline{S}$, and perplexity ratio, $\overline{P}$, separately for sentences of each error type. Then, for each error type, the difference, Δ, between scores in the two datasets is calculated.

$$\Delta\overline{S} = \overline{S}_{\text{CWEB-S}} - \overline{S}_{\text{FCE}}$$

$$\Delta\overline{P} = \overline{P}_{\text{CWEB-S}} - \overline{P}_{\text{FCE}}$$

Figure 5.4 plots these differences for the most common error types. We can observe that, for all error types, edits in CWEB-S result in both a lower degree of semantic change and sentence improvement than edits in FCE. This is particularly evident for the error types R:OTHER, R:SPELL and R:VERB. These are open class errors, where the error and correction can be quite different. It is therefore reasonable that differences in edits' degree of semantic change and perplexity improvement across domains are particularly observed in these cases.[22]

22 Score differences for the R:SPELL error type seem to be driven by a different propensity of spelling errors being of a typographical vs. phonetical nature in the two datasets.

	P	R	$F_{0.5}$
JFLEG	57.55	21.59	43.07
FCE	51.33	17.39	36.92
CoNLL14	40.30	16.56	31.17
W&I-A	45.79	15.10	32.55
W&I-B	43.17	14.46	30.90
W&I-C	33.02	9.81	22.42
LOCNESS	42.09	16.09	31.81
GMEG Wiki	52.36	13.35	32.99
GMEG Yahoo	62.50	16.45	39.45
AESW	10.18	3.58	7.44
CWEB-G	15.20	5.96	11.54
CWEB-S	8.94	1.33	4.17

Table 5.7: Scores of a language model based GEC system. The lower scores on CWEB and AESW indicate an inability to rely on language modelling in low error-density domains.

5.5.2 *Language model importance*

We also investigate the degree to which systems can rely on a strong internal language model representation when evaluated against different domains. We examine this by looking at the performance of a purely language model based GEC system over the different datasets.

We build on the approach of Bryant and Briscoe (2018), using confusion sets to generate alternative versions of an input sentence and then deciding if any of the alternatives are preferable to the original version, based on language model probabilities. The authors use an n-gram language model, which we replace with GPT-2 (Radford et al., 2019) to see how a strong neural language model performs – this approach is similar to Alikaniotis and Raheja (2019). Hyperparameters are tuned for each dataset (see Appendix A.4 for details).

Table 5.7 displays the results on the different datasets. Recall and, in particular, precision is substantially lower on CWEB and AESW compared to other datasets. In general, scores are higher in domains with a higher proportion of errors and those containing edits which result in high perplexity improvements. In these cases systems can rely on a rough heuristic of replacing low probability sequences with high probability ones. However, in CWEB, where errors are fewer and more subtle, this leads to low precision, as perplexity alone cannot differentiate an erroneous sequence from a sequence that is rare but correct. Table 5.8 displays several examples of this, where false positive

False Positive Examples	Perplexity ratio
All types of work are ~~callings~~ **called** to individuals.	0.34
Get started ~~at~~ **with** ACC	0.51
That ~~is~~ **was** actually kind of fun!	0.69

Table 5.8: Examples of false positives on the CWEB dataset that improve perplexity substantially – even more than the average gold edit in CWEB (0.86 perplexity ratio).

corrections suggested by the language model based GEC system have large perplexity improvements.

This analysis suggests that the inability to rely on a strong internal language model representation can negatively impact SOTA system performance on CWEB and on low error density domains in general. This would mean that having large amounts of error examples for training is more important in high-level domains.

5.6 CONCLUSION

We release a new GEC benchmark, CWEB, consisting of website text generated by English speakers at varying levels of proficiency. Comparisons against existing benchmarks demonstrate that CWEB differs in many respects: 1) in the distribution of sentences (higher vocabulary variation and named entity frequency); 2) in error density (lower); and 3) in the types of edits and their impact on language model perplexity and semantic change.

We showed that existing state-of-the-art GEC models achieve considerably lower performance when evaluated on this new domain, even after fine-tuning. We argue that a factor behind this is the inability of systems to rely on a strong internal language model in low error density domains.

We hope that the dataset shall broaden the target domain of GEC beyond learner and/or exam writing and facilitate the development of robust GEC models in the open-domain setting.

Part IV

NON-ENGLISH LANGUAGES

6

DATA STRATEGIES FOR LOW-RESOURCE GRAMMATICAL ERROR CORRECTION

ABSTRACT

Grammatical Error Correction (GEC) is a task that has been extensively investigated for the English language. However, for low-resource languages the best practices for training GEC systems have not yet been systematically determined. We investigate how best to take advantage of existing data sources for improving GEC systems for languages with limited quantities of high quality training data. We show that methods for generating artificial training data for GEC can benefit from including morphological errors. We also demonstrate that noisy error correction data gathered from Wikipedia revision histories and the language learning website Lang8, are valuable data sources. Finally, we show that GEC systems pre-trained on noisy data sources can be fine-tuned effectively using small amounts of high quality, human-annotated data.

6.1 INTRODUCTION

Grammatical Error Correction (GEC) research has thus far been mostly focused on the English language. One reason for this narrow focus is the difficulty of the task – even for English, which has a reasonable amount of high quality data, the task is challenging. Another reason for the English-centric research has been the lack of available GEC benchmark datasets in other languages, which has made it harder to develop GEC systems on these languages.

In the past few years, there are several languages for which GEC benchmarks have become available (Boyd et al., 2014; Davidson et al., 2020; Náplava and Straka, 2019; Rozovskaya and Roth, 2019). Simultaneously, there has been considerable progress in GEC for English using cheap data sources such as artificial data and revision logs (Grundkiewicz and Junczys-Dowmunt, 2014; Grundkiewicz, Junczys-Dowmunt, and Heafield, 2019; Lichtarge et al., 2019). Since these resources are language-agnostic, the time is ripe for investigating these techniques for other languages.

Pretraining GEC systems on artificially generated errors is now common practice for English. Grundkiewicz and Junczys-Dowmunt (2019) showed good results on English, Russian, and German, using a rule based error generation approach that Náplava and Straka (2019) extended to Czech. This approach employed the Aspell dictio-

	Gold			WikiEdits	Lang8
	Train	Dev	Test		
es	10,143	1,408	1,127	4,871,833	1,214,781
de	19,237	2,503	2,337	9,160,287	863,767
ru	4,980	2,500	5,000	8,482,683	684,936
cs	42,210	2,485	2,676	1,193,447	17,061

Table 6.1: Number of sentences for each language.

nary to create confusion sets of phonologically and lexically similar words. In this work, we additionally investigate the usefulness of morphology-based confusion sets. For English, model-based error generation approaches have also been shown to be useful (Kiyono et al., 2019).

State-of-the-art English GEC systems also make use of lower quality data sources, such as Wikipedia revision histories and crowd-sourced corrections from the language learning website Lang8 (Lichtarge et al., 2019; Mizumoto et al., 2011). Given that it is possible to extract data from both Wikipedia and Lang8 in multiple languages, it would be interesting to determine if this data will help improve GEC for non-English languages. Boyd (2018) have already shown promising results for German using Wikipedia revisions with a custom language-specific filtering method.

CONTRIBUTIONS In this work we investigate data strategies for Grammatical Error Correction on languages without large quantities of high quality training data. In particular we answer the following questions: i) Can artificial error generation methods benefit from including morphological errors?; ii) How can we best make use of noisy GEC data when other data is limited?; iii) How much gold training data is necessary?

6.2 GEC DATA SOURCES

6.2.1 *Gold data*

In recent years, high quality GEC datasets have been made available in several languages – in this work we look into Spanish (es), German (de), Russian (ru), and Czech (cs). An overview of the number of sentences for each language is shown in Table 6.1.

SPANISH COWS-L2H (Davidson et al., 2020) is a corpus of learner Spanish corrected for grammatical errors, gathered from essays writ-

ten by mostly beginner level Spanish students at the University of California at Davis.

GERMAN Falko-Merlin (Boyd, 2018) is a parallel error-correction corpus generated by merging two German learner corpora, the Falko (Reznicek et al., 2012) and Merlin (Boyd et al., 2014) corpus. The Falko part of the corpora is gathered from essays from advanced German learners, while Merlin consists of essays from a wider range of proficiency levels.

RUSSIAN RULEC-GEC (Rozovskaya and Roth, 2019) is a GEC-annotated subset of the RULEC corpus. The sources of the corpora are essays and papers written in a university setting by non-native Russian speakers of various levels.

CZECH AKCES-GEC (Náplava and Straka, 2019) is a GEC corpus generated from a subset of the AKCES corpora, which consists of texts written by non-native speakers of Czech.

6.2.2 *Artificial data*

Text can be easily manipulated to destroy its grammatical structure, for example by deleting a word, or swapping the order of two words. Given that large quantities of text in multiple languages are available on the internet it is easy to produce large amounts of artificial training data. Even though these types of rule-based corruption methods do not always simulate realistic errors by human writers, it has been shown that they are still very useful for pre-training GEC models (Grundkiewicz and Junczys-Dowmunt, 2014; Grundkiewicz, Junczys-Dowmunt, and Heafield, 2019; Lichtarge et al., 2019).

Both rule-based and model-based methods for generating artificial data have been shown to be important components of top-performing GEC systems for English, with model-based methods currently yielding the best results (Kiyono et al., 2019). However, model-based methods typically need a large amount of training data to be able to produce an errorful data set that matches the distribution of human writers. For our low-resource setting we therefore employ a rule-based approach.

Rule based error creation approaches using insertion, deletion and replacement operations to corrupt sentences have given good results on both English and other languages (Grundkiewicz and Junczys-Dowmunt, 2019; Náplava and Straka, 2019). Here, for word replacement operations, the Aspell dictionary is commonly used to generate confusion sets of lexically and phonetically similar words that are plausible real-world confusions (Grundkiewicz, Junczys-Dowmunt, and Heafield, 2019). Another potential source of confusion sets, which

we explore in this work, is Unimorph, a database of morphological variants of words available for many languages[1] (Kirov et al., 2018).

6.2.3 *Noisy data*

WIKIPEDIA EDITS Wikipedia is a publicly available, online encyclopedia for which all content is communally created and curated, and is currently available for 316 languages.[2] Wikipedia maintains a revision history of each page, making it possible to extract edits made between subsequent revisions. A subset of the edits contain corrections for grammatical errors. However there are many other types of edits unrelated to the GEC task, such as stylistic changes, change in context, vandalism etc. This noise poses a challenge for training GEC systems.

Data from the Wikipedia edit history is commonly used for training English GEC systems (Grundkiewicz and Junczys-Dowmunt, 2014; Lichtarge et al., 2019), and has also been shown useful for German, when using a custom language-specific filtering method (Boyd, 2018). In order to keep our experiments language-independent, we do not use this filtering method. Instead, we expect that the effects of noise in the Wikipedia data would be mitigated by the subsequent finetuning on gold data. For our experiments, we use the data generation scripts from Lichtarge et al. (2019) to gather training examples from the Wikipedia edit history (see Table 6.1); we refer to this data source as WIKIEDITS.

LANG8 Lang8 is a social language learning website, where users can post texts in a language they are learning, which are then corrected by other users who are native or proficient speakers of the language. The website contains relatively large quantities of sentences with their corrections (Table 6.1) which can be used for training GEC models (Mizumoto et al., 2011). Lang8, however, also contains considerable noise. The corrections may include additional comments. Also, there is high variability in the language proficiency of users providing the corrections.

6.3 SYSTEMS

For all experiments we use the *Transformer* sequence-to-sequence model (Vaswani et al., 2017) available in the Tensor2tensor library.[3] The model is trained with early stopping, using Adafactor as optimizer with inverse square root decay (Shazeer and Stern, 2018). A detailed overview of hyperparameters is listed in Appendix A.6.[4]

1 http://unimorph.org
2 https://meta.wikimedia.org/wiki/List_of_Wikipedias
3 https://github.com/tensorflow/tensor2tensor
4 We used the "transformer_clean_big_tpu" setting

	cs	de	ru	es
		Artificial		
Unimorph	71.08	60.87	32.91	44.68
Aspell	71.53	**63.49**	32.86	**48.22**
Aspell+Unimorph	**71.90**	62.55	**35.95**	48.20
		WikiEdits		
WikiEdits	55.14	58.00	23.92	47.35
Artificial→WikiEdits	**74.64**	**66.74**	40.68	**52.56**
Artificial+WikiEdits	72.91	66.66	**42.80**	51.55
		Summary		
N&S (2019)	**80.17**	**73.71**	**50.20**	-
G&J (2019)	-	70.24	34.46	-
Artificial	71.90	63.49	35.95	48.22
+ WikiEdits	74.64	66.74	42.80	52.56
+ Lang8	75.07	69.24	44.72	**57.32**

Table 6.2: $F_{0.5}$ scores of experiments on the ARTIFICIAL, WIKIEDITS, and LANG8 data sources.

We compare our results to two baseline GEC systems, Grundkiewicz and Junczys-Dowmunt (2019) (G&J) and Náplava and Straka (2019) (N&S), which have both been evaluated on Russian and German, and for Náplava and Straka (2019) additionally on Czech. Both of these systems are pretrained on artificial data and finetuned on gold data. When training the models several strategies were used: source and target word dropouts, edit-weighted maximum likelihood estimation and checkpoint averaging. In this work we do not employ these techniques because our focus is primarily on comparing methods for data collection and generation and less on surpassing the state-of-the-art.

6.4 EXPERIMENTS

We evaluate our models using $F_{0.5}$ score computed using the MaxMatch scorer (Dahlmeier and Ng, 2012). For all experiments, the reported scores are computed for the model trained on the specified data source, further finetuned on the gold training data.

6.4.1 *Creating artificial data*

We first investigate if artificial data creation methods can benefit from the inclusion of morphology-based confusion sets generated from Unimorph.

We train the systems on 10 million examples generated from the WMT News Crawl using the rule-based method from Náplava and Straka (2019) which is a modification of the method presented by Grundkiewicz and Junczys-Dowmunt (2019).

First, for each sentence a word-level (or character-level) error probability is sampled from a normal distribution with a predefined mean and standard deviation. The number of words (or characters) to corrupt are then decided by multiplying the probability by the number of words (or characters) in the sentence. Each corruption is then performed using one of the following operations: insert, swap-right, substitute and delete. Furthermore, at the word level an operation to change the casing is included and at the character level an operation to replace diacritics is included. The operation to apply is selected based on probabilities estimated from the development sets. All parameters used in our experiments are presented in Appendix A.7.

When creating the artificial data we report three experiments – for the word substitution operation a replacement word is chosen from a confusion set generated by either 1) Aspell; 2) Unimorph; or 3) Aspell or Unimorph with equal likelihood (Aspell + Unimorph).

Table 6.2 shows that only using Unimorph performs the worst. This is expected since the system would only learn to correct morphological substitution errors. Mixing Aspell and Unimorph works better for Russian and Czech but for the other languages, using Aspell alone performs better. Thus including Unimorph can help for morphological rich languages, such as Russian and Czech. We will refer to the best performing artificially created dataset for each language as ARTIFICIAL.

6.4.2 *Including noisy data*

We next investigate whether data extracted from Wikipedia revisions and Lang8 can improve our systems even further.

WIKIEDITS We perform three experiments: 1) training on WIKIEDITS from scratch; 2) fine-tuning on WIKIEDITS, starting from models pre-trained on ARTIFICIAL (ARTIFICIAL→WIKIEDITS); and 3) training on an equal-proportion mix of ARTIFICIAL and WIKIEDITS (ARTIFICIAL + WIKIEDITS). From Table 6.2, training only on WIKIEDITS performs worse than the models trained solely on ARTIFICIAL. However, fine-tuning the ARTIFICIAL-trained model on WIKIEDITS gives a large improvement. This suggests that the model primed for the GEC task by pre-training on ARTIFICIAL can better handle the noise in WIKIEDITS.

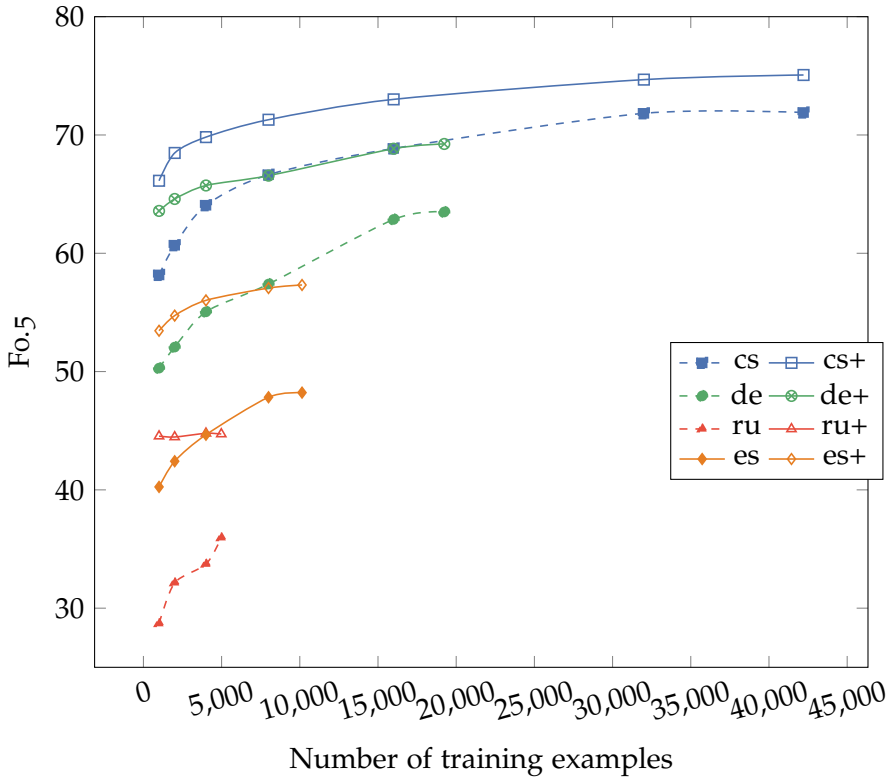


Figure 6.1: GEC performance ($F_{0.5}$) for different amounts of gold training data. Systems have been pretrained on ARTIFICIAL. The + denotes system has additionally been pretrained on WIKIEDITS and LANG8

Mixing the two sources is generally worse, indicating that WIKIEDITS, despite its noise, is of a higher quality and contains realistic GEC errors. However, this is not the case for Russian, where it is better to mix the two data sources. This suggests that Russian Wikipedia revisions are more likely to be unrelated to GEC, and mixing it with ARTIFICIAL regularizes this noise.

LANG8 Fine-tuning the best model from the WIKIEDITS experiments on LANG8 improves performance on all languages (Table 6.2), which confirms the utility of this data source as a valuable source of grammatical corrections.

6.4.3 *How much gold data do we need?*

Human annotated (Gold) data is a scarce resource, as human annotations are expensive. Therefore it is important to determine how much data is necessary to train useful GEC systems in new languages. We analyze the performance of systems finetuned on increasingly larger subsets of available data.

We investigate two scenarios: 1) finetuning a model pretrained only on ARTIFICIAL, and 2) finetuning a model pretrained on ARTIFICIAL,

WikiEdits, and Lang8 (using the best method from previous experiments). This ablation allows us to assess whether noisy data sources can ameliorate the need for gold data.

Performance curves (Figure 6.1) flatten out quickly at around 15k sentences, suggesting that not much data is needed. This is especially the case when the system has additionally been trained on WikiEdits and Lang8. This demonstrates that it is possible to obtain a reasonable quality without much human-annotated data in new languages.

6.5 CONCLUSION

In this paper we have investigated how best to make use of available data sources for GEC in low resource scenarios. We have shown a set of best practices for using artificial data, Wikipedia revision data and Lang8 data, that gives good results across four languages.

We show that using Unimorph for generating artificial data is useful for Russian and Czech, which are morphologically rich languages. Using Wikipedia edits is a valuable source of data, despite its noise. Lang8 is an even better source of high-quality GEC data, despite its smaller size and uncertainties associated with crowdsourcing. When using gold data for fine-tuning, even small amounts of data can yield good results. This is especially true when the initial model has been pretrained on Wikipedia edits and Lang8. We expect this work to provide a good starting point for developing GEC systems for a wider range of languages.

Part V

CONCLUSION

7

DISCUSSION OF THE CONTRIBUTIONS

Throughout this thesis, we have made contributions to the GEC field, pushing the field towards greater usability in an industrial setting. In this final chapter, we present a summary and discussion of our contributions.

In the first part of the thesis, we focused on the issue of current GEC systems being dependant on large amounts of expensive data not available outside academia. The first research question asked in this thesis was:

How can data-scarcity in GEC be dealt with?

In Chapter 3, we did a deep-dive into the benefits of artificial error generation, by looking through the narrow lens of subject-verb agreement error detection. We showed that including large amounts of artificially generated data when training the systems yields better and more robust models. In particular, it makes the system more robust to challenging linguistic phenomena and other errors in the sentence. As artificial data can be produced cheaply in large amounts, this approach is an effective avenue for dealing with data-scarcity in industrial GEC systems. In Chapter 4, we presented an unsupervised approach to GEC based on the noisy channel framework. This approach leveraged strong pre-trained language models and a channel model estimated from edits extracted from Wikipedia revisions. While the system did not yield state-of-the-art results, it still highlighted a viable avenue for creating GEC systems without annotated training data. It is also likely that this approach will benefit from future generations of improved language models.

The second part of this thesis focused on domain generalization of current GEC systems to answer the second research question:

How do GEC systems perform outside the ELL domain?

In Chapter 5, we introduced a new GEC benchmark, CWEB, consisting of website text annotated for correctness, and showed that state-of-the-art GEC systems do not generalize well to this domain; While these systems perform well in the ELL domain, text from more advanced writers poses a challenge to them. In particular, we show that systems perform poorly on text with a low density of errors and suggest that a factor behind this is GEC systems' inability to rely on a strong internal language model in low-error density domains. While GEC has been shown to generalize to some domains outside the ELL domain, this work indicates that more effort is needed to develop open-domain GEC systems.

The third part of this thesis focused on GEC for non-English languages, to answer the third research question:

How can GEC be broadened to non-English languages?

In Chapter 6, we showed a set of strategies for leveraging available data sources of lower quality to achieve good results across a range of languages. Pre-training GEC models on artificially generated data served as a strong starting point while being a cheap and effective method for all investigated languages. For morphologically rich languages, GEC systems further benefited from including morphological confusions when creating artificial errors. The Wikipedia revision history, freely available for many languages, also proved a useful resource for extracting training data, despite containing a lot of noise in the form of non-GEC-related edits. Data extracted from the website Lang8 proved an even better source of very high-quality data. Finally, we showed that systems, when pre-trained on noisy data, can be finetuned effectively on just small amounts of expert-annotated data. This work has demonstrated that data strategies commonly used for English GEC can be successfully transferred to non-English languages and highlights the feasibility of inexpensively broadening GEC systems to new languages.

In sum, during this PhD project, several of the obstacles holding back GEC in industry have been bridged. GEC can now be adopted in industry with low data acquisition costs and work well across many languages. However, more effort is still needed for GEC to generalize to an open-domain setting.

Part VI

APPENDIX

APPENDIX

A.1 RESULTS PER ERROR TYPE

Error type	#	All models	C+B+G	C+B+beam	C+G+beam	B+G+beam
M:Punct	422	65.66	65.86	65.54	64.15	65.65
R:Adj	24	8.93	7.69	8.20	7.81	15.38
R:Adv	17	17.24	12.20	17.24	16.67	13.16
R:Conj	5	2.70	1.92	2.65	2.65	2.36
R:Det	129	23.34	19.92	23.15	22.24	23.29
R:Morph	128	35.71	29.48	28.12	31.18	35.09
R:Noun	70	25.42	23.81	25.21	23.08	23.81
R:Noun:Infl	19	40.00	38.46	37.31	69.57	46.67
R:Noun:Num	290	47.18	43.82	42.59	47.11	46.46
R:Orth	349	4.59	4.58	4.57	4.61	4.60
R:Other	618	13.95	13.30	14.24	13.29	15.07
R:Part	15	40.23	44.12	38.89	33.98	41.67
R:Prep	292	42.24	39.47	41.46	40.46	42.01
R:Pron	50	37.04	34.25	32.22	34.04	35.48
R:Spell	321	76.22	73.66	75.59	70.85	75.02
R:Verb	134	10.10	9.76	9.35	11.57	10.47
R:Verb:Form	169	49.32	44.53	17.86	46.58	48.03
R:Verb:Infl	7	96.77	96.77	96.77	96.77	96.77
R:Verb:SVA	146	76.06	72.73	72.66	73.16	74.88
R:Verb:Tense	160	26.56	26.88	26.61	31.73	30.03
U:Pron	21	0.00	20.00	0.00	18.52	20.00
U:Punct	118	39.69	39.13	39.91	39.79	39.91
All types	4498	40.17	37.44	35.08	39.06	39.73

Table A.1: Span-level correction results ($F_{0.5}$) for different error types (we do not show results for the error types that we do not predict). **C**: Channel Model, **B**: BERT, **G**: GPT-2.

A.2 DATASET DOWNLOAD LINKS

- JFLEG: https://github.com/keisks/jfleg
- FCE: https://www.cl.cam.ac.uk/research/nl/bea2019st/#data
- CoNLL14: https://www.comp.nus.edu.sg/~nlp/conll14st.html
- Write&Improve-A/B/C: https://www.cl.cam.ac.uk/research/nl/bea2019st/#data
- LOCNESS: https://www.cl.cam.ac.uk/research/nl/bea2019st/#data
- GMEG Yahoo/Wiki: https://github.com/grammarly/GMEG
- AESW: http://textmining.lt/aesw/aesw2016down.html

A.3 NON-AVERAGED FINE-TUNING SCORES

	P	**R**	**$F_{0.5}$**
CWEB-G	53.88	34.24	48.33
CWEB-S	43.65	31.1	40.39
CWEB (all)	50.25	33.2	45.57

Table A.2: Scores of the GEC-PSEUDODATA system fine-tuned on CWEB data, calculated against both annotators.

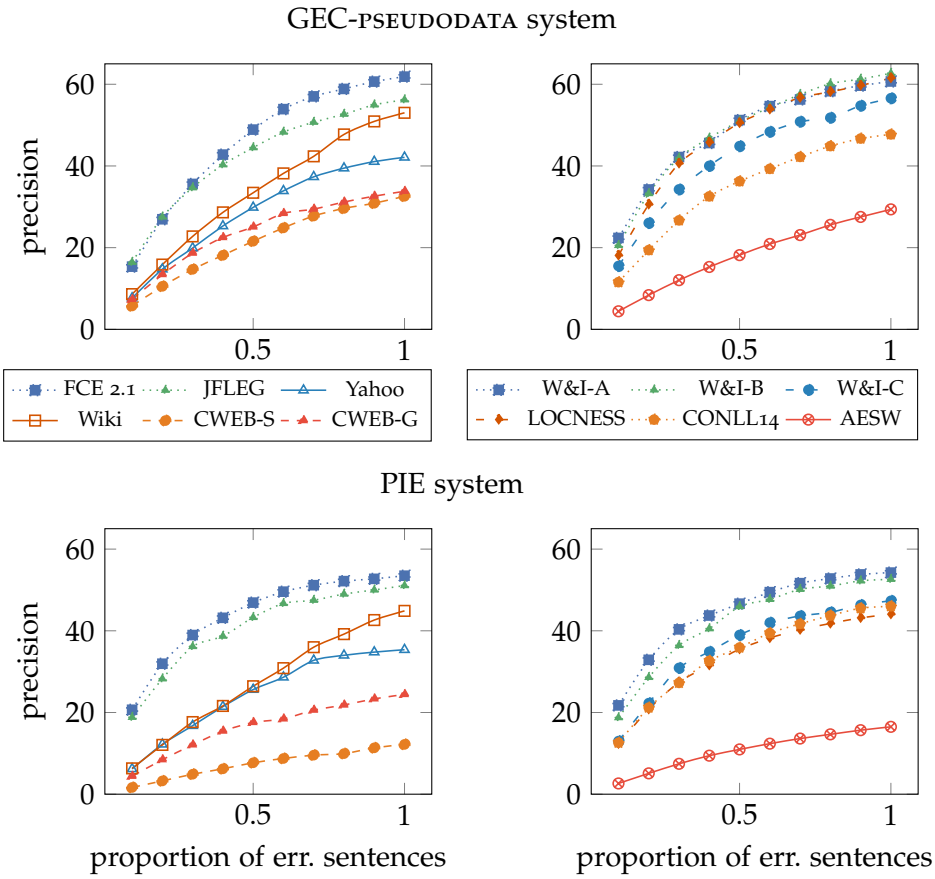
A.4 LANGUAGE MODEL GEC HYPERPARAMETER TUNING

A threshold, τ, determines the degree of probability improvement needed before an alternative sentence is preferred. For each dataset, we find τ, in the 0.9 to 1.0 range, resulting in the best development set $F_{0.5}$. For CoNLL14, we tune on CoNLL13; for W&I, we use the dedicated training sets; for LOCNESS, there is no training set available and so we tune on the W&I subset of advanced texts (W&I-C).

	τ
JFLEG	0.97
FCE 2.1	0.97
CoNLL14	0.98
W&I-A	0.98
W&I-B	0.98
W&I-C	0.97
LOCNESS	0.97
GMEG Wiki	0.96
GMEG Yahoo	0.91
AESW	0.96
CWEB-G	0.96
CWEB-S	0.93

Table A.3: Best performing threshold τ for each domain.

A.5 PRECISION AS A FUNCTION OF THE PROPORTION OF ERRONEOUS SENTENCES



Precision as a function of the proportion of erroneous sentences in each domain.

A.6 MODEL HYPERPARAMETERS

An overview of model hyperparameters used for our GEC system:

- 6 layers for both the encoder and the decoder.
- 8 attention heads.
- A dictionary of 32k word pieces.
- Embedding size $d_{model} = 1024$.
- Position-wise feed forward network at every layer of inner size $d_{ff} = 4096$.
- Batch size = 4096.
- For inference we use beam search with a beam width of 4.
- When pretraining we set the learning rate to 0.2 for the first 8000 steps, then decrease it proportionally to the inverse square root of the number of steps after that.
- When finetuning, we use a constant learning rate of 3×10^{-5}.

A.7 ARTIFICIAL DATA PARAMETERS

Language	**Token-level operations**					**Character-level operations**				
	sub	**ins**	**del**	**swap**	**recase**	**sub**	**ins**	**del**	**swap**	**toggle diacritics**
es	0.69	0.17	0.11	0.01	0.02	0.25	0.25	0.25	0.25	0
cs	0.7	0.1	0.05	0.1	0.05	0.2	0.2	0.2	0.2	0.2
de	0.64	0.2	0.1	0.01	0.05	0.25	0.25	0.25	0.25	0
ru	0.65	0.1	0.1	0.1	0.05	0.25	0.25	0.25	0.25	0

Table A.4: Language specific parameters for token- and character-level noising operations. For all languages word error rate is set to 0.15 and character error rate to 0.02

BIBLIOGRAPHY

Alikaniotis, Dimitris and Vipul Raheja (Aug. 2019). "The Unreasonable Effectiveness of Transformer Language Models in Grammatical Error Correction." In: *Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications*. Florence, Italy: Association for Computational Linguistics, pp. 127–133. DOI: 10.18653/v1/W19-4412. URL: https://www.aclweb.org/anthology/W19-4412.

Andersen, Øistein E., Helen Yannakoudakis, Fiona Barker, and Tim Parish (June 2013). "Developing and testing a self-assessment and tutoring system." In: *Proceedings of the Eighth Workshop on Innovative Use of NLP for Building Educational Applications*. Atlanta, Georgia: Association for Computational Linguistics, pp. 32–41. URL: https://www.aclweb.org/anthology/W13-1704.

Appelman, Alyssa and Mike Schmierbach (2018). "Make No Mistake? Exploring Cognitive and Perceptual Effects of Grammatical Errors in News Articles." In: *Journalism & Mass Communication Quarterly* 95.4, pp. 930–947. DOI: 10.1177/1077699017736040. URL: https://doi.org/10.1177/1077699017736040.

Awasthi, Abhijeet, Sunita Sarawagi, Rasna Goyal, Sabyasachi Ghosh, and Vihari Piratla (Nov. 2019). "Parallel Iterative Edit Models for Local Sequence Transduction." In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Hong Kong, China: Association for Computational Linguistics, pp. 4260–4270. DOI: 10.18653/v1/D19-1435. URL: https://www.aclweb.org/anthology/D19-1435.

BNC-Consortium et al. (2007). "The British National Corpus, Version 3." In: *Distributed by Bodleian Libraries, University of Oxford*. URL: http://www.natcorp.ox.ac.uk/.

Bazerman, Charles (2013). *A theory of literate action. Volume 2 : literate action*. eng. Perspectives on Writing. Fort Collins, Colorado ; The WAC Clearinghouse. ISBN: 1-60235-479-0.

Biemann, Chris, Gerhard Heyer, Uwe Quasthoff, and Matthias Richter (2007). "The Leipzig Corpora Collection: Monolingual Corpora of Standard Size." In: *Proceedings of the Corpus Linguistics Conference*. CL2007. URL: http://ucrel.lancs.ac.uk/publications/CL2007/paper/190_Paper.pdf.

Boyd, Adriane (Nov. 2018). "Using Wikipedia Edits in Low Resource Grammatical Error Correction." In: *Proceedings of the 2018 EMNLP Workshop W-NUT: The 4th Workshop on Noisy User-generated Text*. Brussels, Belgium: Association for Computational Linguistics, pp. 79–

84. DOI: 10.18653/v1/W18-6111. URL: https://www.aclweb.org/anthology/W18-6111.

Boyd, Adriane, Jirka Hana, Lionel Nicolas, Detmar Meurers, Katrin Wisniewski, Andrea Abel, Karin Schöne, Barbora Štindlová, and Chiara Vettori (May 2014). "The MERLIN corpus: Learner language and the CEFR." In: *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*. Reykjavik, Iceland: European Language Resources Association (ELRA), pp. 1281–1288. URL: http://www.lrec-conf.org/proceedings/lrec2014/pdf/606_Paper.pdf.

Brockett, Chris, William B. Dolan, and Michael Gamon (July 2006). "Correcting ESL Errors Using Phrasal SMT Techniques." In: *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*. Sydney, Australia: Association for Computational Linguistics, pp. 249–256. DOI: 10.3115/1220175.1220207. URL: https://www.aclweb.org/anthology/P06-1032.

Bryant, Christopher and Ted Briscoe (June 2018). "Language Model Based Grammatical Error Correction without Annotated Training Data." In: *Proceedings of the Thirteenth Workshop on Innovative Use of NLP for Building Educational Applications*. New Orleans, Louisiana: Association for Computational Linguistics, pp. 247–253. DOI: 10.18653/v1/W18-0529. URL: https://www.aclweb.org/anthology/W18-0529.

Bryant, Christopher, Mariano Felice, Øistein E. Andersen, and Ted Briscoe (Aug. 2019). "The BEA-2019 Shared Task on Grammatical Error Correction." In: *Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications*. Florence, Italy: Association for Computational Linguistics, pp. 52–75. DOI: 10.18653/v1/W19-4406. URL: https://www.aclweb.org/anthology/W19-4406.

Bryant, Christopher, Mariano Felice, and Ted Briscoe (July 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. DOI: 10.18653/v1/P17-1074. URL: https://www.aclweb.org/anthology/P17-1074.

Bryant, Christopher and Hwee Tou Ng (July 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. DOI: 10.3115/v1/P15-1068. URL: https://www.aclweb.org/anthology/P15-1068.

Cai, Dongfeng, Yonghua Hu, Xuelei Miao, and Yan Song (Dec. 2009). "Dependency Grammar Based English Subject-Verb Agreement Evaluation." In: *Proceedings of the 23rd Pacific Asia Conference on Language, Information and Computation, Volume 1*. Hong Kong: City University of Hong Kong, pp. 63–71. URL: https://www.aclweb.org/anthology/Y09-1008.

Chen, Danqi and Christopher Manning (Oct. 2014). "A Fast and Accurate Dependency Parser using Neural Networks." In: *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Doha, Qatar: Association for Computational Linguistics, pp. 740–750. DOI: 10.3115/v1/D14-1082. URL: https://www.aclweb.org/anthology/D14-1082.

Chollampatt, Shamil, Duc Tam Hoang, and Hwee Tou Ng (Nov. 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. DOI: 10.18653/v1/D16-1195. URL: https://www.aclweb.org/anthology/D16-1195.

Chollampatt, Shamil and Hwee Tou Ng (2018a). "A Multilayer Convolutional Encoder-Decoder Neural Network for Grammatical Error Correction." In: *Proceedings of AAAI 2018*. URL: https://www.aaai.org/ocs/index.php/AAAI/AAAI18/paper/download/17308/16137.

– (2018b). "Neural Quality Estimation of Grammatical Error Correction." In: *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Brussels, Belgium: Association for Computational Linguistics, pp. 2528–2539. DOI: 10.18653/v1/D18-1274. URL: https://www.aclweb.org/anthology/D18-1274.

Dahlmeier, Daniel and Hwee Tou Ng (June 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. URL: https://www.aclweb.org/anthology/N12-1067.

Dahlmeier, Daniel, Hwee Tou Ng, and Siew Mei Wu (June 2013a). "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. URL: https://www.aclweb.org/anthology/W13-1703.

Building a large annotated corpus of learner English: The NUS Corpus of Learner English. (2013b), pp. 22 –31.

Dale, Robert, Ilya Anisimoff, and George Narroway (June 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. URL: https://www.aclweb.org/anthology/W12-2006.

Dale, Robert and Adam Kilgarriff (Sept. 2011). "Helping Our Own: The HOO 2011 Pilot Shared Task." In: *Proceedings of the 13th European Workshop on Natural Language Generation*. Nancy, France: Association for Computational Linguistics, pp. 242–249. URL: https://www.aclweb.org/anthology/W11-2838.

Daudaravicius, Vidas, Rafael E. Banchs, Elena Volodina, and Courtney Napoles (June 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. DOI: 10.18653/v1/W16-0506. URL: https://www.aclweb.org/anthology/W16-0506.

Davidson, Sam, Aaron Yamada, Paloma Fernandez Mira, Agustina Carando, Claudia H. Sanchez Gutierrez, and Kenji Sagae (May 2020). "Developing NLP Tools with a New Corpus of Learner Spanish." English. In: *Proceedings of the 12th Language Resources and Evaluation Conference*. Marseille, France: European Language Resources Association, pp. 7238–7243. ISBN: 979-10-95546-34-4. URL: https://www.aclweb.org/anthology/2020.lrec-1.894.

De Felice, Rachele and Stephen Pulman (June 2007). "Automatically Acquiring Models of Preposition Use." In: *Proceedings of the Fourth ACL-SIGSEM Workshop on Prepositions*. Prague, Czech Republic: Association for Computational Linguistics, pp. 45–50. URL: https://www.aclweb.org/anthology/W07-1607.

Devlin, Jacob, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova (June 2019). "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding." In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Minneapolis, Minnesota: Association for Computational Linguistics, pp. 4171–4186. DOI: 10.18653/v1/N19-1423. URL: https://www.aclweb.org/anthology/N19-1423.

Felice, Mariano 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: Human Language Technologies*. Denver, Colorado: Association for Computational Linguistics, pp. 578–587. DOI: 10.3115/v1/N15-1060. URL: https://www.aclweb.org/anthology/N15-1060.

Felice, Mariano and Zheng Yuan (Apr. 2014). "Generating artificial errors for grammatical error correction." In: *Proceedings of the Student Research Workshop at the 14th Conference of the European Chapter of the Association for Computational Linguistics*. Gothenburg, Sweden: Association for Computational Linguistics, pp. 116–126. DOI: 10.

3115/v1/E14-3013. URL: https://www.aclweb.org/anthology/E14-3013.

Felice, Mariano, Zheng Yuan, Øistein E. Andersen, Helen Yannakoudakis, and Ekaterina Kochmar (June 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. DOI: 10.3115/v1/W14-1702. URL: https://www.aclweb.org/anthology/W14-1702.

Fukushima, Kunihiko (1980). "Neocognitron: A Self-Organizing Neural Network Model for a Mechanism of Pattern Recognition Unaffected by Shift in Position." In: *Biological Cybernetics* 36, pp. 193–202.

Gamon, Michael, Jianfeng Gao, Chris Brockett, Alexandre Klementiev, William B. Dolan, Dmitriy Belenko, and Lucy Vanderwende (2008). "Using Contextual Speller Techniques and Language Modeling for ESL Error Correction." In: *Proceedings of the Third International Joint Conference on Natural Language Processing: Volume-I*. URL: https://www.aclweb.org/anthology/I08-1059.

Ge, Tao, Furu Wei, and Ming Zhou (2018). "Reaching Human-level Performance in Automatic Grammatical Error Correction: An Empirical Study." In: *CoRR* abs/1807.01270. arXiv: 1807.01270. URL: http://arxiv.org/abs/1807.01270.

Granger, Sylviane (1998). "The computer learner corpus: a versatile new source of data for SLA research." In: Sylviane Granger, editor, Learner English on Computer. Addison Wesley Longman, London and New York, pp. 3–18.

Grundkiewicz, Roman and Marcin Junczys-Dowmunt (2014). "The WikEd Error Corpus: A Corpus of Corrective Wikipedia Edits and its Application to Grammatical Error Correction." In: *International Conference on Natural Language Processing*. Springer, pp. 478–490.

– (June 2018). "Near Human-Level Performance in Grammatical Error Correction with Hybrid Machine Translation." In: *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*. New Orleans, Louisiana: Association for Computational Linguistics, pp. 284–290. DOI: 10.18653/v1/N18-2046. URL: https://www.aclweb.org/anthology/N18-2046.

– (Nov. 2019). "Minimally-Augmented Grammatical Error Correction." In: *Proceedings of the 5th Workshop on Noisy User-generated Text (W-NUT 2019)*. Hong Kong, China: Association for Computational Linguistics, pp. 357–363. DOI: 10.18653/v1/D19-5546. URL: https://www.aclweb.org/anthology/D19-5546.

Grundkiewicz, Roman, Marcin Junczys-Dowmunt, and Kenneth Heafield (Aug. 2019). "Neural Grammatical Error Correction Systems with Unsupervised Pre-training on Synthetic Data." In: *Proceedings of*

the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications. Florence, Italy: Association for Computational Linguistics, pp. 252–263. DOI: 10.18653/v1/W19-4427. URL: https://www.aclweb.org/anthology/W19-4427.

Gulordava, Kristina, Piotr Bojanowski, Edouard Grave, Tal Linzen, and Marco Baroni (June 2018). "Colorless Green Recurrent Networks Dream Hierarchically." In: *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*. New Orleans, Louisiana: Association for Computational Linguistics, pp. 1195–1205. DOI: 10.18653/v1/N18-1108. URL: https://www.aclweb.org/anthology/N18-1108.

Hochreiter, Sepp and Jürgen Schmidhuber (1997). "Long Short-term Memory." In: *Neural Computation* 9. ISSN: 0899-7667. URL: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.676.4320&rep=rep1&type=pdf.

Junczys-Dowmunt, Marcin and Roman Grundkiewicz (Nov. 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. DOI: 10.18653/v1/D16-1161. URL: https://www.aclweb.org/anthology/D16-1161.

Jurafsky, Daniel and James H. Martin (2009). *Speech and Language Processing*. 2nd. Pearson London. ISBN: 0131873210.

Kaili, Zhu, Chuan Wang, Ruobing Li, Yang Liu, Tianlei Hu, and Hui Lin (2018). "A Simple but Effective Classification Model for Grammatical Error Correction." In: *CoRR* abs/1807.00488. arXiv: 1807.00488. URL: http://arxiv.org/abs/1807.00488.

Kasewa, Sudhanshu, Pontus Stenetorp, and Sebastian Riedel (2018). "Wronging a Right: Generating Better Errors to Improve Grammatical Error Detection." In: *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Brussels, Belgium: Association for Computational Linguistics, pp. 4977–4983. DOI: 10.18653/v1/D18-1541. URL: https://www.aclweb.org/anthology/D18-1541.

Kemighan, Mark D., Kenneth W. Church, and William A. Gale (1990). "A Spelling Correction Program Based on a Noisy Channel Model." In: *COLING 1990 Volume 2: Papers presented to the 13th International Conference on Computational Linguistics*. URL: https://www.aclweb.org/anthology/C90-2036.

Kirov, Christo, John Sylak-Glassman, Roger Que, and David Yarowsky (May 2016). "Very-large Scale Parsing and Normalization of Wiktionary Morphological Paradigms." In: *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*. Portorož, Slovenia: European Language Resources Association (ELRA),

pp. 3121–3126. URL: https://www.aclweb.org/anthology/L16-1498.

Kirov, Christo et al. (May 2018). "UniMorph 2.0: Universal Morphology." In: *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*. Miyazaki, Japan: European Language Resources Association (ELRA). URL: https://www.aclweb.org/anthology/L18-1293.

Kiyono, Shun, Jun Suzuki, Masato Mita, Tomoya Mizumoto, and Kentaro Inui (Nov. 2019). "An Empirical Study of Incorporating Pseudo Data into Grammatical Error Correction." In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Hong Kong, China: Association for Computational Linguistics, pp. 1236–1242. DOI: 10.18653/v1/D19-1119. URL: https://www.aclweb.org/anthology/D19-1119.

Kuncoro, Adhiguna, Chris Dyer, John Hale, Dani Yogatama, Stephen Clark, and Phil Blunsom (July 2018). "LSTMs Can Learn Syntax-Sensitive Dependencies Well, But Modeling Structure Makes Them Better." In: *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Melbourne, Australia: Association for Computational Linguistics, pp. 1426–1436. DOI: 10.18653/v1/P18-1132. URL: https://www.aclweb.org/anthology/P18-1132.

Lample, Guillaume, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer (June 2016). "Neural Architectures for Named Entity Recognition." 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. 260–270. DOI: 10.18653/v1/N16-1030. URL: https://www.aclweb.org/anthology/N16-1030.

Leacock, Claudia, Martin Chodorow, Michael Gamon, and Joel Tetreault (2010). "Automated Grammatical Error Correction for Language Learners." In: *Synthesis lectures on human language technologies* 3.1, pp. 1–134. URL: http://aclweb.org/anthology/C14-3004.

Lichtarge, Jared, Chris Alberti, Shankar Kumar, Noam Shazeer, Niki Parmar, and Simon Tong (June 2019). "Corpora Generation for Grammatical Error Correction." In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Minneapolis, Minnesota: Association for Computational Linguistics, pp. 3291–3301. DOI: 10.18653/v1/N19-1333. URL: https://www.aclweb.org/anthology/N19-1333.

Linzen, Tal, Emmanuel Dupoux, and Yoav Goldberg (2016). "Assessing the Ability of LSTMs to Learn Syntax-Sensitive Dependencies." In: *Transactions of the Association for Computational Linguistics* 4, pp. 521–

535. DOI: 10.1162/tacl_a_00115. URL: https://www.aclweb.org/anthology/Q16-1037.

Makarenkov, Victor, Lior Rokach, and Bracha Shapira (2019). "Choosing the Right Word: Using Bidirectional LSTM Tagger for Writing Support Systems." In: *CoRR* abs/1901.02490. arXiv: 1901.02490. URL: http://arxiv.org/abs/1901.02490.

Manaster-Ramer, Alexis (1987). "Subject-Verb Agreement in Respective Coordinations and Context Freeness." In: *Computational Linguistics* 13, pp. 64–65. URL: https://www.aclweb.org/anthology/J87-1006.

Marcus, Mitchell P., Beatrice Santorini, and Mary Ann Marcinkiewicz (1993). "Building a Large Annotated Corpus of English: The Penn Treebank." In: *Computational Linguistics* 19.2, pp. 313–330. URL: https://www.aclweb.org/anthology/J93-2004.

Mays, Eric, Fred J. Damerau, and Robert Mercer (Jan. 1990). "Context Based Spelling Correction." In: *Information Processing Management* 27, pp. 517–522. DOI: 10.1016/0306-4573(91)90066-U.

Mikolov, Tomas, Ilya Sutskever, Kai Chen, Greg S. Corrado, and Jeff Dean (2013). "Distributed Representations of Words and Phrases and their Compositionality." In: *Proceedings of NIPS 2013*. URL: https://papers.nips.cc/paper/5021-distributed-representations-of-words-and-phrases-and-their-compositionality.pdf.

Miller, George A (1995). "WordNet: a Lexical Database for English." In: *Communications of the ACM* 38.11. URL: http://l2r.cs.uiuc.edu/Teaching/CS598-05/Papers/miller95.pdf.

Mizumoto, Tomoya, Mamoru Komachi, Masaaki Nagata, and Yuji Matsumoto (Nov. 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. URL: https://www.aclweb.org/anthology/I11-1017.

Mutton, Andrew, Mark Dras, Stephen Wan, and Robert Dale (June 2007). "GLEU: Automatic Evaluation of Sentence-Level Fluency." In: *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*. Prague, Czech Republic: Association for Computational Linguistics, pp. 344–351. URL: https://www.aclweb.org/anthology/P07-1044.

Nadejde, Maria and Joel Tetreault (Nov. 2019). "Personalizing Grammatical Error Correction: Adaptation to Proficiency Level and L1." In: *Proceedings of the 5th Workshop on Noisy User-generated Text (W-NUT 2019)*. Hong Kong, China: Association for Computational Linguistics, pp. 27–33. DOI: 10.18653/v1/D19-5504. URL: https://www.aclweb.org/anthology/D19-5504.

Nagata, Ryo and Kazuhide Nakatani (Aug. 2010). "Evaluating performance of grammatical error detection to maximize learning effect." In: *Coling 2010: Posters*. Beijing, China: Coling 2010 Organizing Com-

mittee, pp. 894–900. URL: https://www.aclweb.org/anthology/C10-2103.

Náplava, Jakub and Milan Straka (Nov. 2019). "Grammatical Error Correction in Low-Resource Scenarios." In: *Proceedings of the 5th Workshop on Noisy User-generated Text (W-NUT 2019)*. Hong Kong, China: Association for Computational Linguistics, pp. 346–356. DOI: 10.18653/v1/D19-5545. URL: https://www.aclweb.org/anthology/D19-5545.

Napoles, Courtney, Maria Nădejde, and Joel Tetreault (Mar. 2019). "Enabling Robust Grammatical Error Correction in New Domains: Data Sets, Metrics, and Analyses." In: *Transactions of the Association for Computational Linguistics* 7, pp. 551–566. DOI: 10.1162/tacl_a_00282. URL: https://www.aclweb.org/anthology/Q19-1032.

Napoles, Courtney, Keisuke Sakaguchi, and Joel Tetreault (Apr. 2017). "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. URL: https://www.aclweb.org/anthology/E17-2037.

Ng, Hwee Tou, Siew Mei Wu, Ted Briscoe, Christian Hadiwinoto, Raymond Hendy Susanto, and Christopher Bryant (June 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. DOI: 10.3115/v1/W14-1701. URL: https://www.aclweb.org/anthology/W14-1701.

Ng, Hwee Tou, Siew Mei Wu, Yuanbin Wu, Christian Hadiwinoto, and Joel Tetreault (Aug. 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. URL: https://www.aclweb.org/anthology/W13-3601.

Nivre, Joakim et al. (2017). *Universal Dependencies 2.1*. LINDAT/CLARIN digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University. URL: http://hdl.handle.net/11234/1-2515.

Omelianchuk, Kostiantyn, Vitaliy Atrasevych, Artem Chernodub, and Oleksandr Skurzhanskyi (July 2020). "GECToR – Grammatical Error Correction: Tag, Not Rewrite." In: *Proceedings of the Fifteenth Workshop on Innovative Use of NLP for Building Educational Applications*. Seattle, WA, USA → Online: Association for Computational Linguistics, pp. 163–170. DOI: 10.18653/v1/2020.bea-1.16. URL: https://www.aclweb.org/anthology/2020.bea-1.16.

Park, Jong C., Martha Palmer, and Clay Washburn (Mar. 1997). "An English Grammar Checker as a Writing Aid for Students of English as a Second Language." In: *Fifth Conference on Applied Natural Lan-*

guage Processing: Descriptions of System Demonstrations and Videos. Washington, DC, USA: Association for Computational Linguistics, pp. 24–24. DOI: 10.3115/974281.974296. URL: https://www.aclweb.org/anthology/A97-2014.

Radford, Alec, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever (2019). "Language Models are Unsupervised Multitask Learners." In: *OpenAI Blog*.

Rei, Marek (July 2017). "Semi-supervised Multitask Learning for Sequence Labeling." In: *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Vancouver, Canada: Association for Computational Linguistics, pp. 2121–2130. DOI: 10.18653/v1/P17-1194. URL: https://www.aclweb.org/anthology/P17-1194.

Rei, Marek, Mariano Felice, Zheng Yuan, and Ted Briscoe (Sept. 2017). "Artificial Error Generation with Machine Translation and Syntactic Patterns." In: *Proceedings of the 12th Workshop on Innovative Use of NLP for Building Educational Applications*. Copenhagen, Denmark: Association for Computational Linguistics, pp. 287–292. DOI: 10.18653/v1/W17-5032. URL: https://www.aclweb.org/anthology/W17-5032.

Rei, Marek and Helen Yannakoudakis (Aug. 2016). "Compositional Sequence Labeling Models for Error Detection in Learner Writing." In: *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Berlin, Germany: Association for Computational Linguistics, pp. 1181–1191. DOI: 10.18653/v1/P16-1112. URL: https://www.aclweb.org/anthology/P16-1112.

– (Sept. 2017). "Auxiliary Objectives for Neural Error Detection Models." In: *Proceedings of the 12th Workshop on Innovative Use of NLP for Building Educational Applications*. Copenhagen, Denmark: Association for Computational Linguistics, pp. 33–43. DOI: 10.18653/v1/W17-5004. URL: https://www.aclweb.org/anthology/W17-5004.

Reznicek, Marc, Anke Lüdeling, Cedric Krummes, Franziska Schwantuschke, Maik Walter, Karin Schmidt, Hagen Hirschmann, and Torsten Andreas (2012). "Das Falko-Handbuch. Korpusaufbau und Annotationen Version 2.01." In:

Rozovskaya, Alla, Kai-Wei Chang, Mark Sammons, and Dan Roth (Aug. 2013). "The University of Illinois System in the CoNLL-2013 Shared Task." In: *Proceedings of the Seventeenth Conference on Computational Natural Language Learning: Shared Task*. Sofia, Bulgaria: Association for Computational Linguistics, pp. 13–19. URL: https://www.aclweb.org/anthology/W13-3602.

Rozovskaya, Alla and Dan Roth (June 2011). "Algorithm Selection and Model Adaptation for ESL Correction Tasks." In: *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*. Portland, Oregon, USA: Association

for Computational Linguistics, pp. 924–933. URL: https://www.aclweb.org/anthology/P11-1093.

– (Mar. 2019). "Grammar Error Correction in Morphologically Rich Languages: The Case of Russian." In: *Transactions of the Association for Computational Linguistics* 7, pp. 1–17. DOI: 10.1162/tacl_a_00251. URL: https://www.aclweb.org/anthology/Q19-1001.

Sakaguchi, Keisuke, Courtney Napoles, and Joel Tetreault (Sept. 2017). "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. DOI: 10.18653/v1/W17-5019. URL: https://www.aclweb.org/anthology/W17-5019.

Schäfer, Roland (2015). "Processing and querying large web corpora with the COW14 architecture." In: *Proceedings of the 3rd Workshop on Challenges in the Management of Large Corpora*. CMLC-3. Institut für Deutsche Sprache. URL: https://ids-pub.bsz-bw.de/files/3826/Schaefer_Processing_and_querying_large_web_corpora_2015.pdf.

Schneider, David and Kathleen F. McCoy (1998). "Recognizing Syntactic Errors in the Writing of Second Language Learners." In: *COLING 1998 Volume 2: The 17th International Conference on Computational Linguistics*. URL: https://www.aclweb.org/anthology/C98-2191.

Shazeer, Noam and Mitchell Stern (2018). "Adafactor: Adaptive Learning Rates with Sublinear Memory Cost." In: *arXiv:1804.04235*.

Smith, Charles R., Kathleen E. Kiefer, and Patricia S. Gingrich (1984). "Computers come of age in writing instruction." In: *Comput. Humanit.* 18.3-4, pp. 215–224. DOI: 10.1007/BF02267225. URL: https://doi.org/10.1007/BF02267225.

Smith, Sarah (2016). "Strategic error as style: Finessing the grammar checker." In:

Srivastava, Nitish, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov (2014). "Dropout: A Simple Way to Prevent Neural Networks from Overfitting." In: *The Journal of Machine Learning Research* 15.1, pp. 1929–1958. URL: http://www.jmlr.org/papers/volume15/srivastava14a/srivastava14a.pdf?utm_content=buffer79b43&utm_medium=social&utm_source=twitter.com&utm_campaign=buffer.

Stahlberg, Felix, Christopher Bryant, and Bill Byrne (June 2019). "Neural Grammatical Error Correction with Finite State Transducers." In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Minneapolis, Minnesota: Association for Computational Linguistics, pp. 4033–4039. DOI: 10.18653/v1/N19-1406. URL: https://www.aclweb.org/anthology/N19-1406.

Stahlberg, Felix and Shankar Kumar (Nov. 2020). "Seq2Edits: Sequence Transduction Using Span-level Edit Operations." In: *Proceedings of*

the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP). Online: Association for Computational Linguistics, pp. 5147–5159. DOI: 10.18653/v1/2020.emnlp-main.418. URL: https://www.aclweb.org/anthology/2020.emnlp-main.418.

Sun, Guihua, Gao Cong, Xiaohua Liu, Chin-Yew Lin, and Ming Zhou (2007). "Mining Sequential Patterns and Tree Patterns to Detect Erroneous Sentences." In: *Proceedings of AAAI 2007*. URL: https://www.aaai.org/Papers/AAAI/2007/AAAI07-147.pdf.

Tetreault, Joel, Jennifer Foster, and Martin Chodorow (July 2010). "Using Parse Features for Preposition Selection and Error Detection." In: *Proceedings of the ACL 2010 Conference Short Papers*. Uppsala, Sweden: Association for Computational Linguistics, pp. 353–358. URL: https://www.aclweb.org/anthology/P10-2065.

Tomiyana, Machiko (1980). "Grammatical Errors Communication Breakdown." In: *TESOL Quarterly* 14.1, pp. 71–79. ISSN: 00398322. URL: http://www.jstor.org/stable/3586810.

Toutanova, Kristina, 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 Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*, pp. 252–259. URL: https://www.aclweb.org/anthology/N03-1033.

Turner, Jenine and Eugene Charniak (Apr. 2007). "Language Modeling for Determiner Selection." In: *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Companion Volume, Short Papers*. Rochester, New York: Association for Computational Linguistics, pp. 177–180. URL: https://www.aclweb.org/anthology/N07-2045.

Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin (2017). "Attention Is All You Need." In: *Proceedings of Advances in neural information processing systems (NIPS 2017)*. URL: https://papers.nips.cc/paper/7181-attention-is-all-you-need.pdf.

Wang, Yuzhu and Hai Zhao (Oct. 2015). "A Light Rule-based Approach to English Subject-Verb Agreement Errors on the Third Person Singular Forms." In: *Proceedings of the 29th Pacific Asia Conference on Language, Information and Computation: Posters*. Shanghai, China, pp. 345–353. URL: https://www.aclweb.org/anthology/Y15-2040.

Xie, Ziang, Anand Avati, Naveen Arivazhagan, Dan Jurafsky, and Andrew Y. Ng (2016). "Neural Language Correction with Character-Based Attention." In: *CoRR* abs/1603.09727. arXiv: 1603.09727. URL: http://arxiv.org/abs/1603.09727.

Xie, Ziang, Guillaume Genthial, Stanley Xie, Andrew Ng, and Dan Jurafsky (June 2018). "Noising and Denoising Natural Language: Diverse Backtranslation for Grammar Correction." In: *Proceedings of the 2018 Conference of the North American Chapter of the Association*

for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers). New Orleans, Louisiana: Association for Computational Linguistics, pp. 619–628. DOI: 10.18653/v1/N18-1057. URL: https://www.aclweb.org/anthology/N18-1057.

Yannakoudakis, Helen, Øistein E Andersen, Ardeshir Geranpayeh, Ted Briscoe, and Diane Nicholls (2018). "Developing an automated writing placement system for ESL learners." In: *Applied Measurement in Education* 31.3, pp. 251–267. URL: https://doi.org/10.1080/08957347.2018.1464447.

Yannakoudakis, Helen, Ted Briscoe, and Ben Medlock (June 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. URL: https://www.aclweb.org/anthology/P11-1019.

Yannakoudakis, Helen, Marek Rei, Øistein E. Andersen, and Zheng Yuan (Sept. 2017). "Neural Sequence-Labelling Models for Grammatical Error Correction." In: *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. Copenhagen, Denmark: Association for Computational Linguistics, pp. 2795–2806. DOI: 10.18653/v1/D17-1297. URL: https://www.aclweb.org/anthology/D17-1297.

Zeiler, Matthew D. (2012). "ADADELTA: An Adaptive Learning Rate Method." In: *arXiv preprint arXiv:1212.5701*. arXiv: 1212.5701. URL: https://arxiv.org/pdf/1212.5701.