博士学位論文

論文題目　Interactive text rewriting
for non-native English speakers
（非英語母語話者のためのインタラクティブな書き換え）

提　出　者　東北大学大学院情報科学研究科
システム情報科学　　専　攻
学籍番号　C0ID2002
氏　名　　伊藤拓海

Interactive text rewriting for non-native English speakers

非英語母語話者のためのインタラクティブな書き換え

TOHOKU
UNIVERSITY

Takumi Ito

Graduate School of Information Sciences
Tohoku University

This dissertation is submitted for the degree of
Doctor of Information Science

January 2023

Acknowledgments

I thank my primary advisor, Professor Kentaro Inui, for his tremendous guidance and advice over the six years since my fourth year of undergraduate studies. He has given me valuable opportunities not only for research activities but also for entrepreneurial activities and study abroad and has provided me with tremendous support. I would like to express my sincere gratitude. I am deeply grateful to Professor Yoshifumi Kitamura for taking time out of his busy schedule to review this dissertation as a committee member. I am also grateful to Professor Jun Suzuki for various research meetings, often several times a week. He taught me essential skills for research activities, such as thinking about research, designing experiments, and writing papers. I would like to express my deepest gratitude to him. I thank Professor Kees van Deemter of Utrecht University for accepting me to study at Utrecht University for six months as a visiting researcher. He taught me many skills necessary for research, especially academic writing. Thank you also for taking care of my life in the Netherlands by inviting me to home parties and visiting museums. It was a precious experience.

Dr. Naomi Yamashita of NTT Communication Science Laboratories guided me in every detail of research design, experimental methods, and paper writing related to human-computer interaction (HCI) research. Her advice was beneficial, as no members in the Tohoku NLP lab had experience in HCI research. I appreciate your guidance and collaboration. Dr. Masato Hagiwara of Octanove Labs LLC helped me a lot with my thesis writing and implementation during my master's program. In particular, during the development of TEASPN, he instructed me on the use of GitHub and good coding practices, which was very useful for my subsequent research. I would like to express my gratitude. Dr. Hayato Kobayashi of Google, Inc. guided me in writing my thesis and designing experiments during my master's research. I greatly appreciate your assistance. Prof. Ge Gao of the University of Maryland gave me much advice on my HCI research. I would like to express my gratitude.

Dr. Masatoshi Hidaka of Edge intelligence systems Inc. has been a great help to me in my research and entrepreneurial activities. Notably, his support was essential to the development of the Langsmith Editor. Let's continue to work together to develop this service to make it even better. I would like to thank Dr. Tatsuki Kuribayashi for his tremendous support in my research and entrepreneurial activities. I am delighted that we could conduct various research projects. I look forward to working with you to make Langsmith more exciting. Dr. Shun Kiyono of LINE Corporation provided me with much guidance on programming and other techniques necessary for my research. In particular, I learned a great deal from him in the WMT 2020 News Translation Task. In addition to the research activities, we also enjoyed mountain trekking and skiing on our days off, which made our laboratory life fulfilling.

Ms. Mayumi Sugawara, Ms. Haruka Aizawa, and Ms. Yoriko Isobe have helped me with daily office work and preparations for crowdsourcing experiments. They also listened to my trivial problems, and thanks to their support, I have led a fulfilling laboratory life. Mr. Kenshi Yamaguchi helped me with critical technical aspects of my research, such as managing computer clusters and collecting data. I would like to express our sincere appreciation. I would like to thank Mr. Masakatsu Moriyama, Mr. Takashi Goto, and Mr. Toshihiko Yonemura of Langsmith Inc. for supporting my research and entrepreneurial activities.

Tohoku University's Graduate Program in Data Science and JSPS Research Fellowship for Young Scientists supported my research activity financially.

I would like to thank the member of Tohoku NLP Group and the NLP Group at Utrecht University for their various advice and generous support. Finally, I greatly thank my family and friends for their generous support.

Abstract

Writing a paper is daunting, especially for non-native English speakers (NNESs) with limited English proficiency. It takes more time to write a clear and understandable paper. The paper may not be accepted due to its English. This can have a negative impact not only on the career development of the researcher but also on the diversity of science as a whole. While collaborating with English-speaking co-authors or utilizing an English editing service can be helpful, these options may not be available to all NNESs. As a new option, we aim to reduce the disadvantages by using natural language processing (NLP) techniques, in particular, rewriting techniques. Rewriting is a framework that provides paraphrases and fluency-enhancing alternatives to human-written text.

While there has been some work, most writing assistance has focused on correcting surface-level issues such as grammar, spelling, and typographical errors. We broaden this focus to include the earlier revising stage, where sentences require adjustment to the information included or major rewriting, and propose a new writing assistance task. Rewriting models performing well in this task can help inexperienced authors by producing fluent, complete sentences given their rough drafts. To evaluate the rewriting models, we build a crowdsourced evaluation dataset consisting of incomplete sentences authored by non-native writers paired with their final versions extracted from published academic papers. We also investigate how data augmentation techniques can be employed to construct training data for developing rewriting models.

Using the rewriting model, we built a writing support system named Langsmith. Through a laboratory experiment, we demonstrated that Langsmith helps NNESs write papers in English. Langsmith is now available to the public.

Furthermore, we conducted user studies and interviews to investigate how NNESs use the writing support system. We find that many NNESs wrote using machine translation and struggled to assess suggestions from the rewriting system. Some blindly accepted the revisions, while others collected additional clues (e.g., machine translation of system suggestions) for assistance.

Based on our findings of this study, we conclude with a discussion of future research directions. Particularly, we emphasize the importance of collaboration between natural language processing and human-computer interaction.

Table of contents

List of figures

List of tables

Chapter 1

Introduction

Since English has become the primary language of global business, non-native English speakers (NNESs) are becoming required to communicate in English. In academia, in particular, non-native English speakers are disadvantaged because they are expected to write papers in English (Huang, 2010). Occasionally, papers are not accepted due to a lack of English proficiency, and researchers and students with low English proficiency may spend a significant amount of time and money writing papers (Ramírez-Castañeda, 2020). The resulting lack of paper acceptance can also affect the researcher's career and, as a result, may lead to a decrease in diversity in academia.

To overcome these linguistic barriers, writing-support systems based on natural language processing technology can be beneficial (Dale and Kilgarriff, 2011; Hagiwara et al., 2019). In particular, recent developments in neural networks have led to rapid advances in text generation technology, and the potential for new modes of assistance, such as rewriting to improve fluency as well as correcting grammatical and spelling errors, is gaining attention (Dale and Viethen, 2021).

While technologies are being developed that may be useful for writing support, a sufficient practice of what kind of support is feasible and advantageous for NNESs has not yet been accumulated. As a case study, we focus on assisting Japanese researchers and students who are NNESs in writing papers. Based on existing studies (Grangier and Auli, 2018; Napoles et al., 2017; Zhu et al., 2019), we examine how to realize several modes of writing support using natural language technology and build a writing support system named Langsmith. Although Langsmith has a number of features, its primary function is to rewrite text automatically. We then ask NNESs to use the writing support system and investigate whether they find it useful for writing, which modes are particularly beneficial, and how they use it. Furthermore, we discuss the directions for building more effective writing support systems for NNESs.

1.1 Research Issues

As of 2023, Numerous studies and tools for writing support have been developed. However, at the inception of our study, there was little data and few writing support tools available. Therefore, the following research issues were posed in stages.

- **How can we build rewriting models:** Building a rewriting model requires training data. For example, to construct a rewriting model that improves fluency, a large amount of pair data of a draft and the corresponding fluent sentence is needed. However, it is often not easy to collect such training data in large quantities. In addition, evaluation of the rewriting model is also an important process, and the automatic evaluation framework also requires pair data of draft text and corresponding fluent text. In this thesis, we explore the construction of data for evaluation and how to build a rewriting model.

- **Can automatic rewriting help NNESs write academic papers?:** Building a high-performance rewriting model on automatic evaluation may not imply that it will be useful for writing for non-native English speakers. Furthermore, issues other than performance, such as user interface, need to be considered when developing writing support system applications. We will build a writing support system with a rewriting model and conduct laboratory trials to assess whether it can be useful for NNESs writing.

- **How do NNESs use AI-powered rewriting tools?:** Although text generation technology is improving, sometimes the system generates text with errors. We conduct an interview study to understand how NNESs assess the system's suggestions and what factors lead them to form trust in the system.

1.2 Contributions

The contributions of this thesis are as follows:

- **Proposing new rewriting tasks in the academic domain and building evaluation datasets:** We propose a new rewriting task: generating fluent sentences from drafts.

We create an evaluation dataset for the task using a new crowdsourcing approach and made the datasets publicly available.

- **Adapting data augmentation methods to create training data for the rewriting model:** Training data is needed to model the newly proposed rewriting task, but crowdsourcing the data would be costly as with the evaluation data. We propose a framework by applying data augmentation techniques to build rewriting models.
- **Building a writing-support system and investigating the impact on writing by non-native English speakers:** We build a writing support system called Langsmith, with a built-in rewriting model. We demonstrate the effectiveness of Langsmith by asking non-native English-speaking students to write in English text using Langsmith.
- **Investigating the use of the writing-support system by non-native English speakers:** We release Langsmith to the public, analyze user logs and survey users to investigate how they use Langsmith. In addition, we conduct user studies and interviews to understand how non-native English speakers assess the tool's suggestions and how they form trust toward Langsmith.

1.3 Thesis Overview

The rest of this thesis is structured as follows:

- **Chapter 2: Background.** In this chapter, we discuss the difficulties non-native English speakers face when writing, and then we organize an overview of writing assistance research in NLP.
- **Chapter 3: Rewriting Tasks and Rewriting Models for Academic Writing Support.** We propose Sentence-level Revision (SentRev) as a new writing assistance task. Well-performing systems in this task can assist inexperienced non-native English speakers by producing fluent, complete sentences given their rough, incomplete drafts. We build an evaluation dataset consisting of incomplete sentences authored by non-native English writers paired with their final versions extracted from published academic papers for developing and evaluating SentRev models. We also create training data using data augmentation methods and establish baseline models.
- **Chapter 4: Langsmith: An Interactive Academic Text Revision System.** This chapter presents the Langsmith editor, which assists inexperienced, non-native researchers in writing English papers. Our system can suggest fluent, academic-style

sentences to writers based on their rough, incomplete phrases or sentences. We asked students who are not native English speakers to write a paper in English using Langsmith and demonstrated the effectiveness of Langsmith.

- **Chapter 5: How Non-native English speakers use AI-powered writing-support tools.** In this chapter, we investigate the use of Langsmith for non-English native speakers. In particular, we investigate whether and how non-native English speakers correctly evaluate the corrections provided by Langsmith. We investigated participant interactions with the tool through user studies and interviews. We found that most participants had difficulty evaluating the recommended revisions. Some blindly accepted the modifications, while others collected additional clues (e.g., machine translation of revised sentences) to assist them. Based on these findings, we discuss factors that shape NNESs' trust in AI-powered writing tools and their quality assessment of the recommended revisions.
- **Chapter 6: Conclusion and Future Work.** This chapter summarizes the contributions of this thesis and further discusses directions for future research on writing assistance for non-native English speakers.

The research in Chapter 3 was presented at The 12th International Conference on Natural Language Generation (INLG 2019) (Ito et al., 2019) and the Journal of Cognitive Science (Ito et al., 2020b). The research in Chapter 4 was presented at the 2020 Conference on Empirical Methods in Natural Language Processing (System Demonstrations) (Ito et al., 2020a).

Chapter 2

Background

This chapter discusses the challenges NNESs face when writing in English and the overall picture of writing support using NLP technology to address these challenges.

2.1 Language barriers of non-native English speakers

English is the dominant language in numerous domains, including academia, and NNESs, especially those with low English proficiency, may find themselves at a disadvantage due to a lack of knowledge of English grammar, collocations, phrases, and style (Flowerdew, 2007; Huang, 2010; Ramírez-Castañeda, 2020). Huang (2010) interviewed NNES Ph.D. students and found that, indeed, many felt at a disadvantage due to their limited English proficiency. NNES students appear to draft the content of their planned writing in their native languages and then translate it into the target language during the writing process (Cohen and Brooks-Carson, 2001). Thus, a lack of English writing skills can prolong writing time and, in some cases, lead to manuscript rejection despite the reporting of valuable research results due to inadequate communication (Flowerdew, 2007; Huang, 2010; Politzer-Ahles et al., 2020; Ramírez-Castañeda, 2020). Although translation and editing services can be used to address this language barrier, this comes at a steep financial cost (Ramírez-Castañeda, 2020). As well as having negative career impacts in 'publish or perish' academic environments, these barriers to publication by NNES researchers are detrimental to diversity and inclusion goals.

2.2 Natural language processing for writing-support.

To overcome the above-mentioned barriers to writing and publishing activity, a number of studies have been conducted in the field of NLP related to writing assistance (Dale and

Table 2.1 Writing-support tools. GEC, CMP, and TR stand for grammatical error correction, auto-completion, and text rewriting, respectively.

		Features		
Tool	Domain	GEC	CMP	TR
Langsmith	Academic	✓	✓	✓
Wordtune	General			✓
Grammarly	General	✓		✓
QuillBot	General	✓		✓
Ginger	General	✓		✓
Trinka	Academic	✓		✓
Write With Transformer	General & Academic		✓	

Kilgarriff, 2011; Ito et al., 2019). A variety of writing support tools have also been developed in recent years. Table 2.1 lists several writing support tools. Writing assistance can take many types, including the summary display (Dang et al., 2022) and example sentence searches (Boisson et al., 2013; Soyer et al., 2015); here, we focus on three features commonly used in recent writing support tools (Dale and Viethen, 2021): grammatical error correction, autocompletion, and rewriting.

2.2.1 Grammatical error correction

Grammatical error correction (GEC) is the task of correcting text that contains grammatical, spelling, or other errors (Ng et al., 2014; Yuan and Briscoe, 2016). It has long been addressed in NLP and is implemented today as a basic feature of most editors. Although grammatical and spelling errors are naturally likely to be made by native speakers as well, GEC tasks have evolved with the goal of helping NNESs (Dale and Kilgarriff, 2011). Most benchmark datasets are based on texts written by non-native English speakers (Dahlmeier et al., 2013; Mizumoto et al., 2011).

2.2.2 Autocompletion

Text completion is a task that generates text from human-written prompts. Completing text is also a typical function of a text completion application (*Write With Transformer*[1] and *Smart Compose* (Chen et al., 2019)). Research and development of this framework are often focused on applications such as the generation of stories and slogans, which require creativity. With the recent development of neural language models, which can generate very fluent

[1] https://transformer.huggingface.co

sentences, a framework has been proposed in which the language model is regarded as another author, and writing is based on the generated text. In particular, with the advent of GPT-3 (Brown et al., 2020) and the release of its API[2], research and development of tools utilizing this functionality have been active in recent years (Lee et al., 2022).

2.2.3 Rewriting

A rewriting task is a task that performs rewriting on the user-written text and aims not only to correct errors, as in GEC, but also to generate paraphrases (Zhou and Bhat, 2021), improve fluency (Napoles et al., 2017), transform style (Jin et al., 2022a), and simplify the text (Al-Thanyyan and Azmi, 2021). Encoder-Decoder architecture (Vaswani et al., 2017) is often used to build the rewriting model. We also use this architecture to build the rewriting model (Chapter 3).

Recent studies have focused on interactive writing support (Du et al., 2022; Dwivedi-Yu et al., 2022; Schick et al., 2023; Sun et al., 2021) because human writing is not a linear process but an iterative one. For example, Du et al. (2022) have published a dataset annotated edit-intention, which aims to tell the system what it should change, such as fluency and style. Our study also follows this trend and aims to develop systems that support rewriting interactively with humans.

2.3 Automatic academic writing assistance

Academic writing assistance has gained considerable attention in NLP because writing papers is an important task for researchers (Dale and Kilgarriff, 2011; Daudaravičius, 2015; Lee and Webster, 2012; Wu et al., 2010; Yimam et al., 2020). For example, some shared tasks of detecting and correcting grammatical errors were organized to assist NNESs in writing their papers (Dale and Kilgarriff, 2011; Daudaravičius, 2015). The rewriting of academic domains has also been actively studied. Over the past few years, a number of datasets have been created to support academic domain rewriting (Dong et al., 2021; Du et al., 2022; Mita et al., 2022). In addition, other technologies have been studied to support academic writing from various perspectives, such as automatic abstract generation (Wang et al., 2019), automatic table and figure description generation (Moosavi et al., 2021), automatic citation generation (Wang et al., 2021), and even automatic review generation (Wang et al., 2020; Yuan et al., 2022). Academic paper writing is one of the many writing activities worth pursuing.

[2] https://openai.com/api/

Chapter 3

Rewriting Tasks and Rewriting Models for Academic Writing Support

3.1 Introduction

Writing an academic paper can be a daunting task, even for experienced writers who are native or near-native English speakers. Inexperienced, non-native speakers find themselves in an even more difficult circumstance. In addition to grammatical or spelling errors, their sentences may lack fluency, have an unnatural style, contain collocation errors, or have missing words that they could not remember or did not know the appropriate expressions. Such writers, especially students with limited academic experience, may often have difficulty putting their ideas and findings into words, even if the ideas are sound and contribute to the research community. Improving the quality of writing is thus a concern for individual researchers and the academic community.

Writing assistance technologies have been extensively studied in natural language processing (NLP) (Brill and Moore, 2000; Grangier and Auli, 2018; Ng et al., 2014). Our goal is to help inexperienced writers in writing fluent, grammatical sentences.

Models developed for academic writing assistance using existing datasets can serve as a support system during the final stages by editing a nearly finished version of the paper. For example, Daudaravičius (2015) collects scientific papers before and after professional editing from publishing companies, and Dale and Kilgarriff (2011) extract published papers that still contain errors and correct the errors to obtain target text fragments.

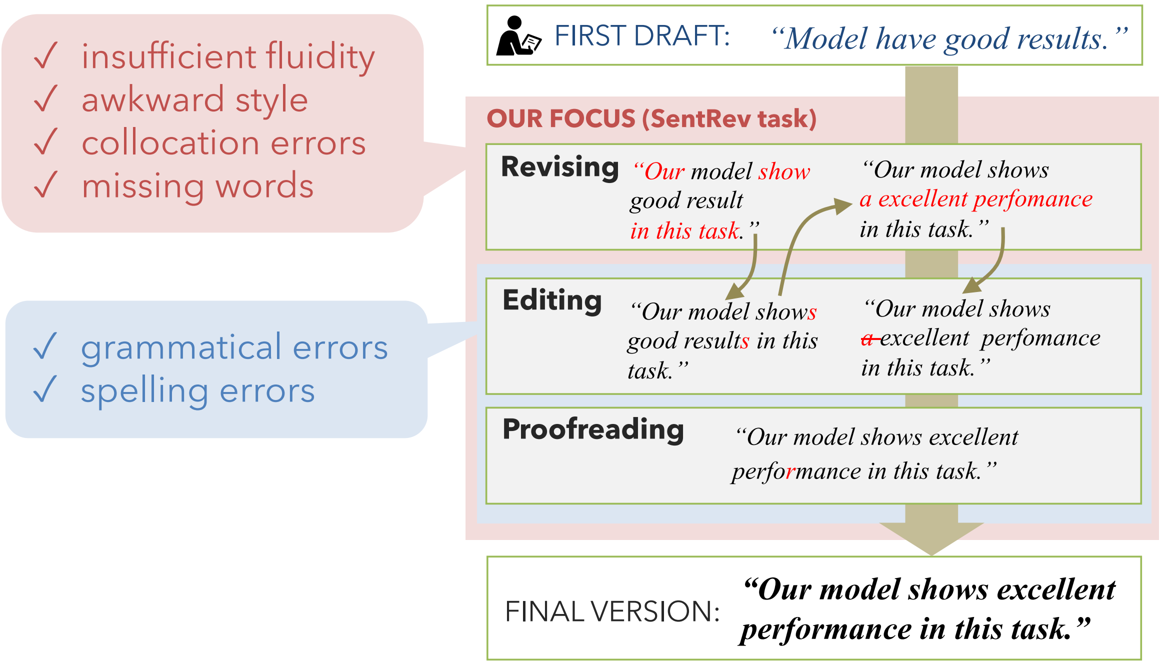

✓ insufficient fluidity
✓ awkward style
✓ collocation errors
✓ missing words
✓ grammatical errors
✓ spelling errors
FIRST DRAFT: "Model have good results."
OUR FOCUS (SentRev task)
Revising
"Our model show good result in this task."
"Our model shows a excellent perfomance in this task."
Editing
"Our model shows good results in this task."
"Our model shows ~~a~~ excellent perfomance in this task."
Proofreading
"Our model shows excellent performance in this task."
FINAL VERSION: "Our model shows excellent performance in this task."

Figure 3.1 Overview of the estimated process of writing a sentence *"Our model shows excellent performance in this task."* and focus of this study. Writing activity consists of four stages: (i) drafting, (ii) revising, (iii) editing, and (iv) proofreading.

Process-writing pedagogy, on the other hand, asserts that writing involves several processes (Buchman et al., 2000; Seow, 2002; Susser, 1994), as shown in Figure 3.1. This study addresses the challenge of automatic assistance in the final review process (*proofreading* and *editing*) and the earlier stages of writing (*revising*). In the revising stage, writers may drastically modify the wording and supplement some words, a highly demanding task for non-native or inexperienced writers. Supporting the revising stage has been less explored in NLP.

In this study, we design a new type of academic writing assistance task, Sentence-level Revision (SentRev), where a system takes an early draft of a sentence as input and generates a revised, error-free, proofread version.

A major issue in addressing this assistance task is that evaluation datasets are scarce because early-stage drafts are not usually publicly available. To overcome this limitation, we construct an evaluation dataset of pairs of draft sentences and their final versions, the *Set of Modified Incomplete TecHnical paper sentences* (SMITH), that we created using crowdsourcing approaches. We then evaluate the quality of our dataset and in-depth analyze the characteristics of the obtained drafts. Finally, we train baseline models and report the performance for our task on the SMITH evaluation dataset.

Our contribution is fourfold:

- We propose a new task called Sentence-level Revision (SentRev).
- We create an evaluation dataset, SMITH, for SentRev using a new crowdsourcing approach and release it.[1]
- We compare the characteristics of our dataset with major datasets and analyze the obtained draft sentences.
- We establish baseline models.

3.2 The Sentence-level Revision task

Table 3.1 Examples of sentence-level revisions in our SMITH dataset. Our task is to transform the draft sentences into their corresponding reference sentences.

Draft	*However, the F1 score of KBP 2017 coupus* <*> *decreased by the sub event base rule.*
Reference	*However, subevent based constraints slightly reduced the F1 scores on KBP 2017 corpus.*
Draft	*But, there are some important difference to* <*> *our work unique.*
Reference	*However, there exist several key differences that make our work unique.*

SentRev is the task of revising and editing incomplete draft sentences to create final versions. Examples of sentence-level revision are shown in Table 3.1. A draft sentence, x, may have several types of problems. Surface-level problems such as typographical, spelling, or grammatical errors are common. Wording problems, such as collocation errors or expressions being stylistically odd or inappropriate for the academic domain, are typical of rough sentences written by non-native, inexperienced writers. The third type of error is *information gaps*. Information gaps are cases where the author likely could not find the appropriate wording for the idea he or she wanted to convey, such as a specific expression common in the academic domain or a technical term. In addition, a draft sentence may be missing sections without the author being aware of this. Solving the above problems in a draft sentence would elevate the draft sentence x to its final or nearly final version y with greatly improved fluency. Ideally, a single error-free and correctly filled-in final version should be generated while considering the context of the sentence. However, as a first step, an assistance system

[1] https://github.com/taku-ito/INLG2019_SentRev

may output *likely candidates* for the user to choose from or be inspired by, which would be realistic for a real-world application (see Chapter 4).

SentRev is to generate likely final versions y from early-draft sentences x. For this purpose, we provide an evaluation dataset, Smith, comprising pairs of drafts and their final versions (X, Y).

3.3 The Smith dataset

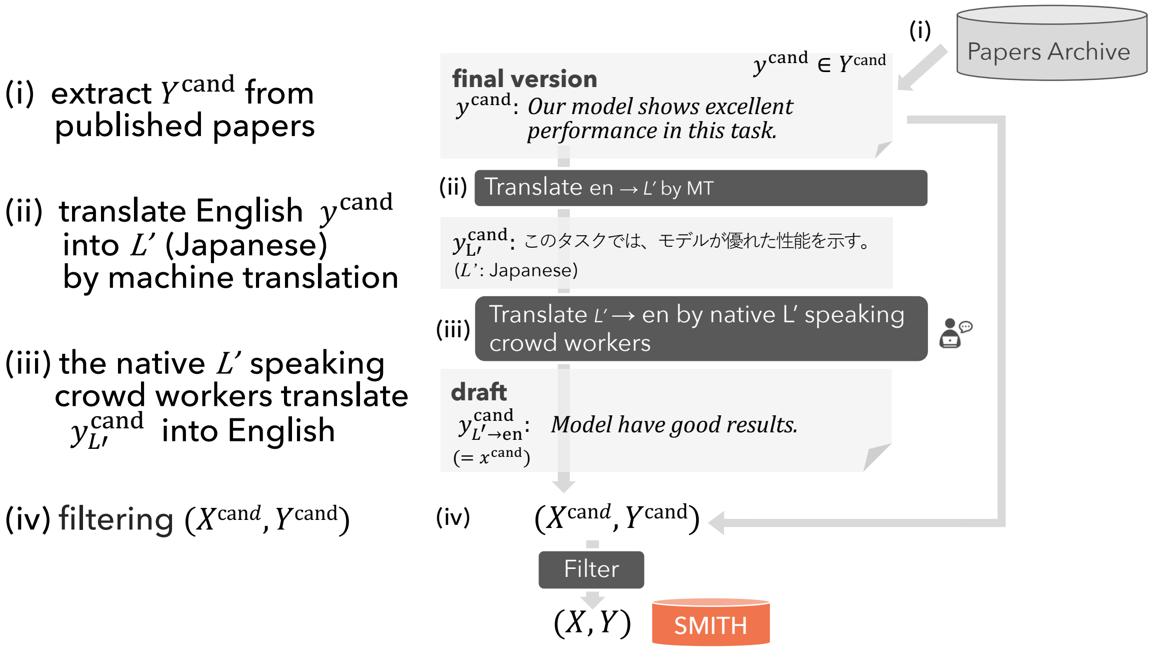

(i) extract Y^{cand} from published papers
(ii) translate English y^{cand} into L' (Japanese) by machine translation
(iii) the native L' speaking crowd workers translate $y^{\mathrm{cand}}_{L'}$ into English
(iv) filtering $(X^{\mathrm{cand}}, Y^{\mathrm{cand}})$
(i)
Papers Archive
$y^{\mathrm{cand}} \in Y^{\mathrm{cand}}$
final version
y^{cand}: Our model shows excellent performance in this task.
(ii)
Translate en → L' by MT
$y^{\mathrm{cand}}_{L'}$: このタスクでは、モデルが優れた性能を示す。
(L': Japanese)
(iii)
Translate L' → en by native L' speaking crowd workers
draft
$y^{\mathrm{cand}}_{L'\to\mathrm{en}}$: Model have good results.
$(= x^{\mathrm{cand}})$
(iv)
$(X^{\mathrm{cand}}, Y^{\mathrm{cand}})$
Filter
(X, Y)
SMITH

Figure 3.2 Overview of the crowdsourcing protocol for creating an evaluation dataset, Smith, for the SentRev task.

3.3.1 Dataset creation method

Process overview Although we cannot collect "drafts" X from published papers, we can easily collect the "final versions" Y. In addition, we have access to non-native, inexperienced writers through crowdsourcing services. Our evaluation dataset creation process combines these two factors (Figure 3.2). The protocol consists of the following four phases:

(i) Collecting many sentences written by experts Y^{cand} from published papers.

(ii) Translating them into another language L', resulting in sentences $Y^{\mathrm{cand}}_{L'}$.

(iii) Asking native speakers of L' to translate $Y_{L'}^{\mathrm{cand}}$ back into English $Y_{L'\rightarrow\mathrm{en}}^{\mathrm{cand}}$ through crowdsourcing. Henceforth, we denote $Y_{L'\rightarrow\mathrm{en}}^{\mathrm{cand}}$ as X^{cand}.

(iv) Filtering the pairs of $(X^{\mathrm{cand}},\ Y^{\mathrm{cand}})$ to ensure the quality of the dataset $(X,\ Y)$.

This setting is analogous to the situation non-native writers face, as Cohen and Brooks-Carson (2001) report that non-native speakers tend to formulate in their native language and mentally translate to the target second language. We assume that most crowd workers have never written an academic paper and that the target users of SentRev-based systems also include this type of inexperienced writers.

First, we create many candidate pairs of drafts and reference sentences $(X^{\mathrm{cand}},\ Y^{\mathrm{cand}})$ and then filter them to create the quality-controlled set $(X,\ Y)$. The following subsections detail this process.

Collecting final version sentences We collected sentences Y^{cand} from the ACL Anthology Sentence Corpus (AASC).[2] We extracted the sentences that satisfied the following conditions from the AASC as Y^{cand}:

- accepted to the Annual Meeting of the Association for Computational Linguistics (2018),
- between 70 and 120 characters,
- does not include mathematical symbols, citation tokens, URLs, Greek letters, or other special symbols defined in AASC, and
- has no clear conversion mistakes when automatically extracted from PDFs.

Creating draft sentences We used Japanese as L'. First, we translated Y^{cand} into Japanese using Google Translate.[3] We denote the Japanese versions of Y^{cand} by $Y_{\mathrm{ja}}^{\mathrm{cand}}$. To guarantee the quality of $Y_{\mathrm{ja}}^{\mathrm{cand}}$, the authors, who were native Japanese speakers, inspected all the sentences from $Y_{\mathrm{ja}}^{\mathrm{cand}}$ and removed those that at least one speaker judged to be incorrect translations.

Next, we asked each Japanese crowd worker to translate three sentences from $Y_{\mathrm{ja}}^{\mathrm{cand}}$ into English $Y_{\mathrm{ja}\rightarrow\mathrm{en}}^{\mathrm{cand}}$ within 15 minutes. The appropriate time limit and rules were determined based on several trials.

The workers were allowed to insert the special symbol <∗> in places where they could not think of a good expression for that position in their answer $Y_{\mathrm{ja}\rightarrow\mathrm{en}}^{\mathrm{cand}}$. This instruction revealed the information gaps that the authors of the drafts consciously left empty. An author may

[2] https://github.com/KMCS-NII/AASC

[3] https://translate.google.com/

Table 3.2 Criteria for evaluating workers. L.D denotes the Levenshtein distance.

Criteria	Judgment
Writing time is too short (< 2 minutes)	Reject
All answers are too short (< 4 words)	Reject
No answer ends with "." or "?"	Reject
Contain identical answers	Reject
Some answers have Japanese words	Reject
No answer is recognized as English	Reject
Some answers are too short (< 4 words)	-2 points
Some answers use fewer than 4 kinds of words	-2 points
Too close to machine translation result (20 <= L.D. <= 30)	-0.5 points/ans
Too close to machine translation result (10 <= L.D. <= 20)	-1.5 points/ans
Too close to machine translation result (L.D. <= 10)	Reject
All answers end with "." or "?"	+1 points
Some answers have <*>	+1 points
All answers are written in English	+1 points

also be unaware that a draft sentence is missing sections. 306 workers participated in our crowdsourcing task.

Quality control We designed specific filtering criteria and applied them to the workers because Yahoo! crowdsourcing,[4] a Japanese crowdsourcing service, does not provide filtering based on the worker's writing skills or abilities. The filtering was based on the writing activities of the workers. We evaluated each worker using the three answers they produced according to the criteria outlined in Table 3.2. We then accepted work only from workers who received a score of 0 or higher as valid. We determined the hyperparameters through trial experiments. We used spaCy-CLD[5] for language detection.

In addition, to exclude instances with a too large gap, we automatically filtered out the obtained $(x^{\mathrm{cand}}, y^{\mathrm{cand}}) \in (X^{\mathrm{cand}}, Y^{\mathrm{cand}})$ whose unigram overlap coefficient was considerably low:

$$\frac{|\,U(x^{\mathrm{cand}}_{\mathrm{checked}}) \cap U(y^{\mathrm{cand}})|}{\min\{|\,U(x^{\mathrm{cand}}_{\mathrm{checked}})|, |\,U(y^{\mathrm{cand}})|\}} < \alpha \ ,$$

[4] https://crowdsourcing.yahoo.co.jp/

[5] https://github.com/nickdavidhaynes/spacy-cld

where $U(\cdot)$ is the set of tokens excluding stop-words and special tokens (<∗>). $x^{\text{cand}}_{\text{checked}}$ is the spell-checked version[6] of x^{cand}. α is set to 0.4, which was determined in trial experiments. We collected 10,804 pairs of draft and their final versions, which cost us approximately US$4,200, including the trial rounds of crowdsourcing.

Unfortunately, work produced by unmotivated workers could have evaded the aforementioned filters and lowered the quality of our dataset. For example, the workers could have circumvented the filter by simply repeating common phrases in academic writing, such as "We apply we apply". To estimate the frequency of such instances, we sampled 100 (x, y) pairs from (X, Y) and asked a Japanese- and English-fluent NLP researcher (who was not one of the authors of this paper) to check for examples where x was entirely unrelated to x_{ja}, which was presented to the crowd workers when producing x. The expert observed no completely inappropriate examples but noted a small number of clearly subpar translations. Thus, 95% of sentence pairs were determined to be appropriate. This result demonstrates that, overall, our method was suitable for creating the dataset and confirms the quality of SMITH.

3.3.2 Statistics

Table 3.3 shows the statistics of our SMITH dataset and a comparison with major datasets for building a writing assistance system (Daudaravičius, 2015; Mizumoto et al., 2011; Napoles et al., 2017). Our dataset, which comprises 10k sentence pairs, is six times larger than JFLEG, which contains both grammatical errors and nonfluent wording. Moreover, our dataset simulates significant editing—99% of the pairs have some changes between the draft and its corresponding reference, and 33% of the draft sentences contain gaps indicated by the special token <∗>. We also measured the amount of change from the drafts X to the references Y by calculating the Levenshtein distance between them. A higher Levenshtein distance in our dataset indicated more significant differences between them compared with major GEC datasets. This finding implies that our dataset emulates more drastic rephrasing.

3.4 Analysis of the Smith dataset

In this section, we run extensive analyses on the sentences written by non-native workers (*draft* sentences X) and the original sentences extracted from the set of accepted papers (*reference* sentences Y). To perform these analyses, we randomly selected 500 pairs from SMITH.

[6] We corrected spelling errors using https://github.com/barrust/pyspellchecker

Table 3.3 Comparison with existing datasets. w/mask and w/change denote the percentage of source sentences with mask tokens and the percentage where the source and target sentences differ, respectively.

dataset	size	w/mask	w/change	levenshtein distance
Lang-8	2.1M	-	42%	3.5
AESW	1.2M	-	39%	4.8
JFLEG	1.5k	-	86%	12.4
SMITH	10k	33%	99%	47.0

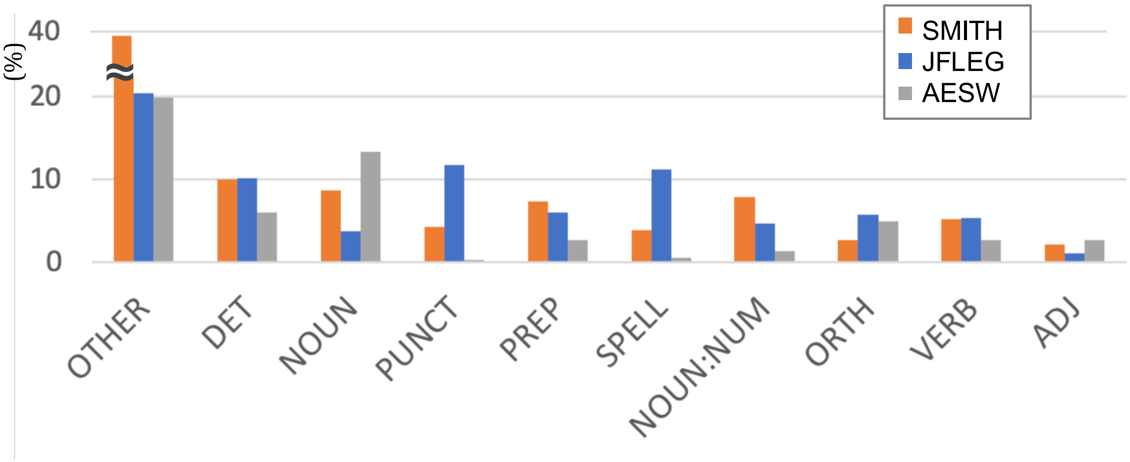
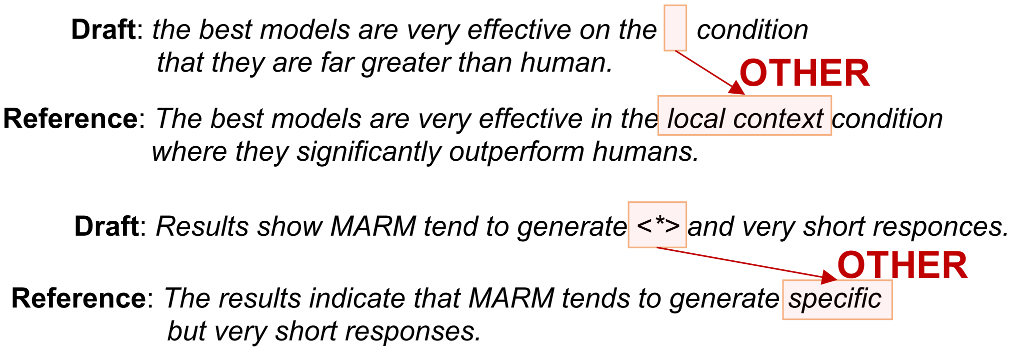


Figure 3.3 Comparison of the top 10 frequent errors observed in the 3 datasets.

3.4.1 Error type comparison

To estimate the distributions of error types between the source and target sentences, we used ERRANT (Bryant et al., 2017; Felice et al., 2016)[7]. We then compared them with three datasets: SMITH, AESW (the same domain as SMITH), and JFLEG (which has a relatively close Levenshtein distance to SMITH). To calculate the error type distributions on AESW and JFLEG, we randomly sampled 500 pairs of source and target sentences from each corpus. Figure 3.3 shows the results of the comparison. Although all datasets contained a mix of error types and operations, the SMITH dataset featured more "OTHER" operations than the other two datasets. A manual review of some samples of "OTHER" operations revealed that they tend to inject information missing in the draft sentence (see Figure 3.4). This finding

[7] https://github.com/chrisjbryant/errant



Draft: *the best models are very effective on the condition that they are far greater than human.*

Reference: *The best models are very effective in the local context condition where they significantly outperform humans.*

Draft: *Results show MARM tend to generate <*> and very short responces.*

Reference: *The results indicate that MARM tends to generate specific but very short responses.*

Figure 3.4 Examples of "OTHER" operations predicted by the ERRANT toolkit.

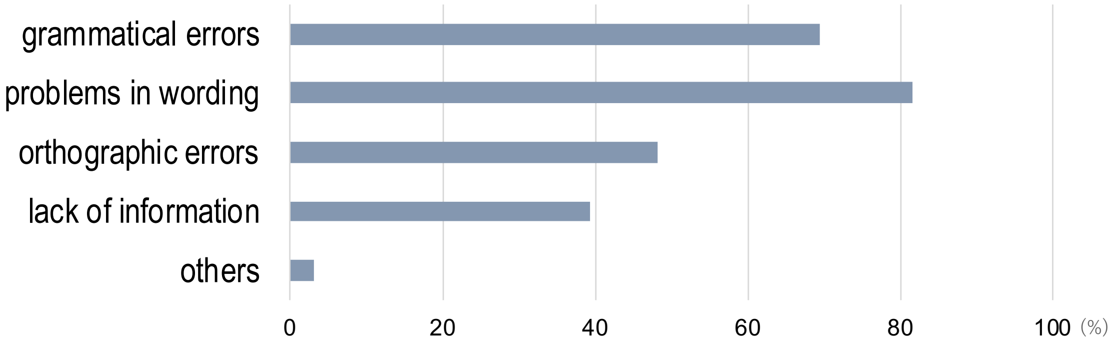


Figure 3.5 Result of the English experts' analyses of error types in draft sentences on our SMITH dataset. The scores show the ratio of sentences where the targeted type of errors occurred.

confirms that our dataset emphasizes a new, challenging "infilling" task setting for writing assistance.

3.4.2 Human error type analysis

To understand the characteristics of our dataset in detail, an annotator proficient in English (who is not an author of this paper) analyzed the types of errors in the draft sentences (see Figure 3.5). The most frequent errors were *fluency problems* (e.g., "In these *ways*" instead of "In these *methods*,")—characterized by errors in academic style and wording, which are beyond the scope of traditional GEC. Another notable type of error that occurred frequently in our dataset was a *lack of information*, which further distinguishes our dataset from other datasets.

Table 3.4 Comparison of the draft and reference sentences in SMITH. FRE and perplexity scores were calculated once in each sentence and then averaged over all the sentences in the development set of SMITH.

data	FRE	passive voice (%)	word repetition (%)	perplexity
Draft X	45.5	34.0	33.0	1373
Reference Y	40.0	29.6	28.6	147

3.4.3 Human fluency analysis

We outsourced the scoring of the fluency of the given draft and reference sentence pairs to three English-proficient annotators. The vast majority of draft sentences x (94.8%) were deemed less fluent than their corresponding reference sentence y, confirming that achieving high performance with our dataset requires the ability to transform unpolished sentences into more fluent sentences.

3.4.4 Sentence-level linguistic characteristics

We conducted various linguistic measures on the dataset sentences including Flesch Reading Ease (FRE) (Flesch, 1948), passive voice[8], word repetition, and perplexity, which are presented in Table 3.4. FRE assesses the *readability* of a text by considering the average number of words per sentence and the average number of syllables per total word. An FRE score ranges from 0 to 100; the higher the score, the easier the text is to read. The draft sentences consistently demonstrated higher FRE scores than their reference counterparts, which may be attributed to the latter containing more sophisticated language and technical terms.

In addition, workers tended to use the passive voice and repeat words within a narrow span, and both those phenomena should be avoided in academic writing. Further analysis was conducted on lexical tendencies between drafts and references. Some words and phrases were more often observed in the reference sentences than in the draft sentences, and vice-versa. Figure 3.6 visualizes these biases, where words more often observed in the draft sentences (e.g., *will*, *is not*, *if*, and *I*,) are plotted in the upper-left corner, and those found more frequently in the references (e.g., *can be*, *no*, *when*, and *they*) are plotted in the lower-right corner. The difference also includes a widely-used spelling (*data set* vs. *dataset*) and a common plurality (*method* vs. *methods*). The plot was created using the scattertext toolkit (Kessler, 2017).

[8] https://github.com/armsp/active_or_passive

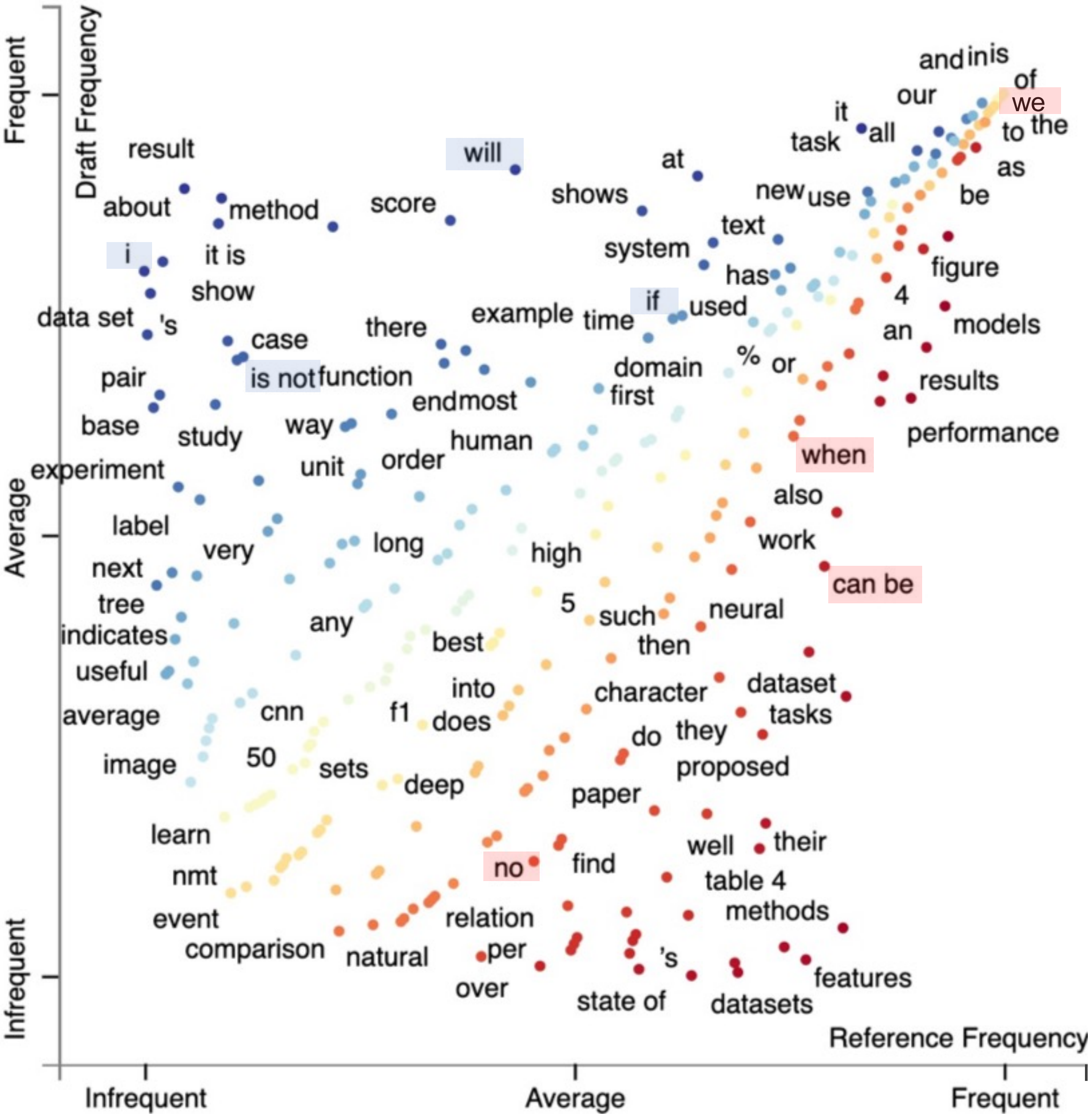


Figure 3.6 Characteristic words and phrases in draft sentences and reference sentences in the development set of SMITH.

Finally, we analyzed the draft and the reference sentences using perplexity calculated by a 5-gram language model trained on ACL Anthology papers.[9] The higher perplexity scores in the draft sentences (Table 3.4) suggest they possess properties that are not appropriate for academic writing, such as less fluent wording.

[9] Perplexity is calculated with the implementation available in the KenLM (Heafield, 2011) (https://github.com/kpu/kenlm), tuned on AASC (excluding the texts used for building the SMITH).

Table 3.5 Examples of the generated training dataset.

method	original	generated
Heuristic	Besides , the recognizer successfully rejected only 15 out of 42 negative sentences .	recognizer Besides successfully , the informativeness rejected of out <*>
Grammatical error generation	We plan to **analyze** these direct communications **and** interaction of sentiments **expressed** in these sequences of posts .	We plan to **analysis** the direct communication interaction of sentiments **express** in these sequence of posts .
Style removal	This experiment **suggested** that there were ambiguities in these pointing gestures and **led to a redesign** of the system .	This experiment **indicated** the ambiguity found in the pointing gestures and **caused a renewal** of the system .
Entailed sentence generation	Figure 2 **illustrates the effectiveness** of different features class.	There is different feature in figure 2 .

3.5 Experiments

3.5.1 Baseline models

We built three baseline models and evaluated them on the SMITH dataset.

Heuristic noising and denoising model

We can access a great deal of the final versions of academic papers. Noising and denoising approaches have gained attention in the GEC and machine translation fields (Edunov et al., 2018; Lichtarge et al., 2019; Xie et al., 2018). We combined these two factors to train baseline models on noised final version sentences.

First, we collected 4,898,146 sentences Y^{aasc} from the AASC dataset that met the following conditions: (i) not included in the SMITH dataset, (ii) not too long or too short (between 5 and 35 tokens), (iii) over 50% of the characters were letters of the alphabet. We then constructed a training dataset ($X^{\text{aasc}}_{\text{hrst}}$, Y^{aasc}) by adding noise to Y^{aasc}.

As the most straightforward approach for noising, we used a set of heuristic rules by randomly deleting, replacing, and swapping words in the reference sentences. Algorithm 1 shows the noising algorithm in the heuristic noising method. In particular, the rules included deleting words with a 0.1 probability, replacing words with a token that appeared over 10,000 times in Y_{aasc} with a 0.1 probability, and shuffling the sentence randomly while maintaining the original adjacent words within a three-word proximity. Additionally, we randomly

Algorithm 1: Heuristic noising

```
INPUT: $x = \{w_0, w_1, \cdots, w_n\}$

$x = \text{delete}(x, 0.1)$
# 10% of the tokens in x are deleted.
$x = \text{replace}(x, 0.1)$
# 10% of the tokens in x are replaced with common terms in ACL.
$x = \text{permutate}(x)$
# permutate the tokens in x.
$r \leftarrow \text{Uniform}(0, 0.5)$
$m = \text{int}(x.\text{length} * r)$
$c = 0$
while $c < m$ do
    $n \leftarrow \text{sample}(\{j \in \mathcal{N} \mid 1 \leq j \leq m - c\})$
    $(s, e) \leftarrow \text{sample}(\{n\text{-grams of } x\})$
    $x = " x_{:s-1} + \text{<*>} + x_{e+1:} "$
    $c = c + n$
end while
# $r \times 100\%$ of the tokens in x are masked.
```

replaced up to half of the words with a <*> token. This method generated 4.8 million heuristically noised sentences.

Subsequently, we trained a denoising model (a mapping function from $X^{\text{aasc}}_{\text{hrst}}$ to Y^{aasc}) by using Transformer (Vaswani et al., 2017) implemented in fairseq (Ott et al., 2019). We used an Adam optimizer (Kingma and Ba, 2015) with $\alpha = 0.0005$, $\beta_1 = 0.9$, $\beta_2 = 0.98$, and $\epsilon = 10e^{-8}$. We set the maximum number of tokens per minibatch to 3,000 and the maximum number of updates to 500,000. We also set the dropout rate to 0.3. The input and output sentences were tokenized and then segmented into character bigrams. In the decoding process, we used a beam width of 5. This model is our first baseline model for the SentRev task (henceforth, H-ND).

Enc-Dec noising and denoising model

As an extension of the heuristic noising and denoising model, we enhanced the noising methods to better simulate the characteristics of X in SMITH than the heuristic rules in Section 3.5.1. As described in Section 3.4, the drafts were likely to have (i) grammatical errors, (ii) stylistically inappropriate wording, and (iii) missing words. We used three neural Encoder-Decoder (Enc-Dec) models to generate the synthetic draft sentences.

Table 3.6 Results of quantitative evaluation. Gramm. denotes the grammaticality score and PPL denotes perplexity.

Model	BLEU	ROUGE-L	BERT-P	BERT-R	BERT-F	P	R	$F_{0.5}$	Gramm.	PPL
Draft X	9.8	46.8	75.9	78.2	77.0	-	-	-	92.9	1454
H-ND	8.2	45.0	77.0	76.1	76.5	5.4	2.9	4.6	94.1	406
ED-ND	**15.4**	**51.1**	**80.9**	**80.0**	**80.4**	21.8	**12.8**	**19.2**	96.3	**236**
GEC	11.9	49.0	80.8	79.1	79.9	**22.2**	6.2	14.6	**96.7**	414
Reference Y	-	-	-	-	-	-	-	-	96.5	147

Grammatical error generation: Here, we trained an Enc-Dec model that introduces synthetic grammatical errors to "clean" sentences using a "flipped" dataset from GEC (clean → erroneous). We used nonidentical sentence pairs (source, target) from the Lang-8, AESW, and JFLEG datasets.

Style removal: To generate stylistically unnatural sentences in the academic domain, we used paraphrasing, which preserves a sentence's content while disregarding its style. We used the ParaNMT-50M dataset (Wieting and Gimpel, 2018), a paraphrase dataset automatically created using neural machine translation. We extracted parallel sentences with annotated paraphrase scores between 0.7 and 0.95 from the ParaNMT-50M dataset and also used swapped pairs of source and target sentences.

Entailed sentence generation: To simulate the missing words in the draft sentences, we trained an Enc-Dec model that generated a sentence entailed with the given sentence. We extracted entailed sentence pairs from the SNLI (Bowman et al., 2015) and the MultiNLI (Williams et al., 2018) datasets.

Random noising beam search: As Xie et al. (2018) pointed out, a beam search often yields too conservative hypotheses. This tendency leads the noising models with the standard beam search to generate synthetic draft sentences similar to their references. Therefore, we applied the random noising beam search (Xie et al., 2018) on three Enc-Dec noising models. During the beam search, we added $r\beta$ to the scores of the hypotheses, where r is a value sampled from a uniform distribution over the interval $[0, 1]$, and β is a penalty hyperparameter set to 5.

We obtained 14.6M sentence pairs of $(X^{\text{aasc}}_{\text{encdec}}, Y^{\text{aasc}})$ by applying these Enc-Dec noising models to Y^{aasc}. To train the denoising model, we used both data $(X^{\text{aasc}}_{\text{hrst}}, Y^{\text{aasc}})$ and $(X^{\text{aasc}}_{\text{encdec}},$

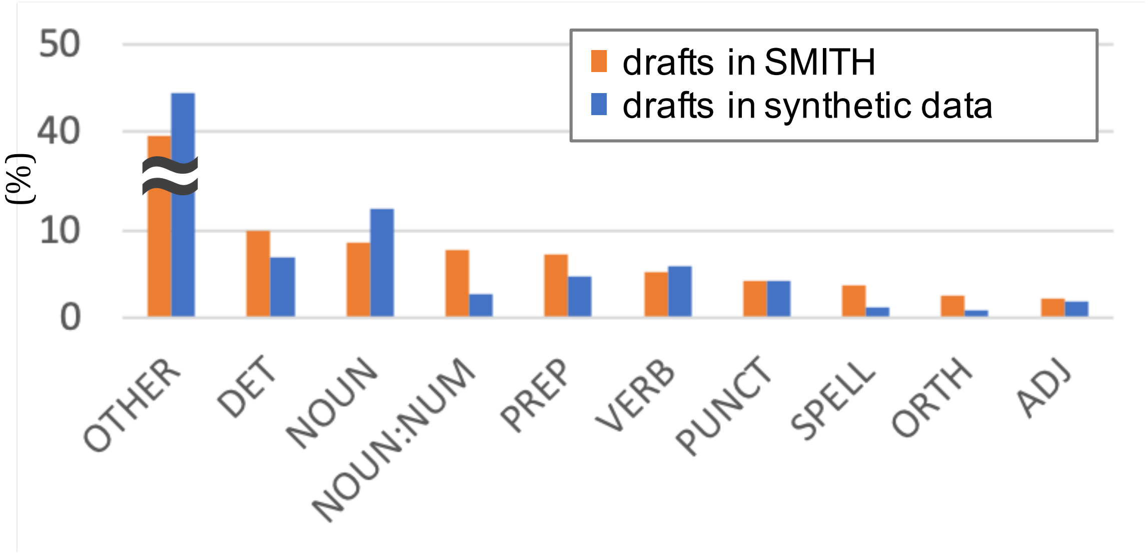


Figure 3.7 Comparison of the 10 most frequent error types in Smith and synthetic drafts created by the Enc-Dec noising methods.

Y^{aasc}). The model architecture was the same as the heuristic model. This denoising model is our second baseline model (ED-ND). To facilitate research in the SentRev task, we released all the 19.6M synthetic data.[10]

Analysis of the synthetic data: We analyzed the error type distribution of the synthetic data used for training the Enc-Dec noising and denoising model with ERRANT (Figure 3.7). The error type distribution from the synthetic dataset had similar characteristics to the one from the development set in Smith (real-draft). Kullback–Leibler divergence between these error type distributions was 0.139. This result would support our assumption that the SentRev task is a combination of GEC, style transfer, and a completion-type task.

Table 3.5 shows examples of the training data generated by the noising models described in Section 3.5. Heuristic noising, the rule-based noising method, created ungrammatical sentences. The grammatical error generation model added grammatical errors (e.g., *plan to analyze* → *plan to analysis*). The style removal model generated stylistically unnatural sentences for the academic domain (e.g., *redesign* → *renewal*). The entailed sentence generation model caused a lack of information.

[10] https://github.com/taku-ito/INLG2019_SentRev

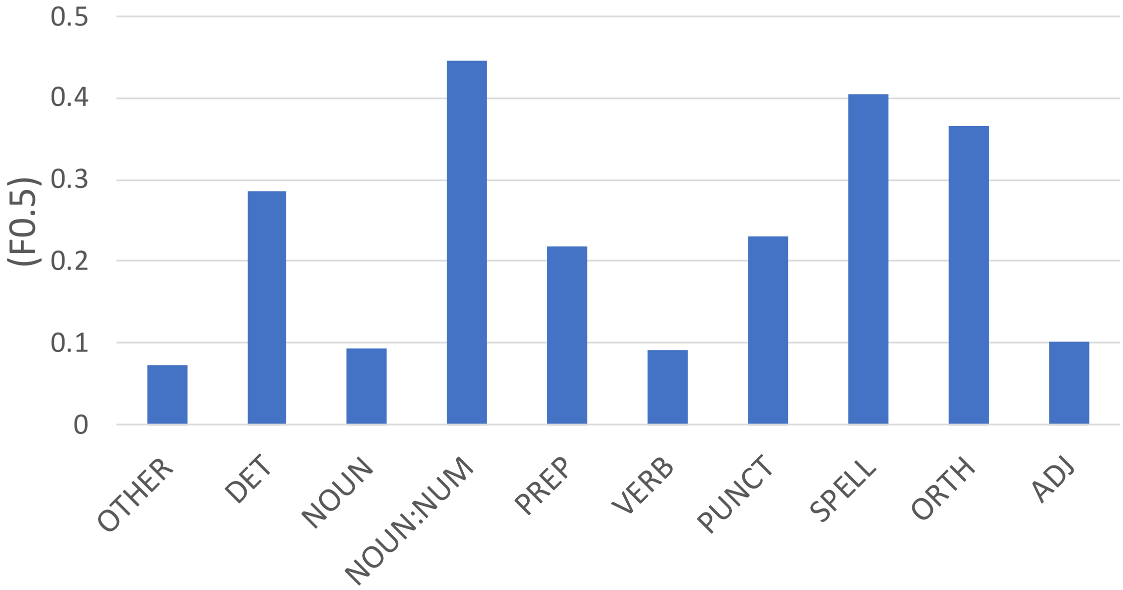


Figure 3.8 Performance of the ED-ND baseline model on top 10 most error types in SMITH.

GEC model

The GEC task is closely related to SentRev. We examined the performance of the current state-of-the-art GEC model (Zhao et al., 2019) in the SentRev task. We applied spelling correction prior to using the GEC model following Zhao et al. (2019).

3.5.2 Evaluation metrics

The SentRev task is not easy to evaluate because it can consider various valid candidate revisions to a given context. As one solution, we evaluated the performance from multiple perspectives using various reference and reference-free evaluation metrics. We used BLEU (Papineni et al., 2002), ROUGE-L (Lin, 2004), and $F_{0.5}$ score, which are widely used metrics in machine translation, style-transfer, and GEC, respectively. We used nlg-eval (Sharma et al., 2017) to compute the BLEU and ROUGE-L scores. We calculated $F_{0.5}$ scores with ERRANT. In addition, to handle the lexical and compositional diversity of valid revisions, we used BERTScore (Zhang et al., 2020), a contextualized embedding-based evaluation metric. We also used two reference-free evaluation metrics: grammaticality score (Napoles et al., 2016) and perplexity. Grammaticality was scored as $1 - (N_{\text{errors in sentence}} / N_{\text{tokens in sentence}})$, where the number of grammatical errors in a sentence is obtained using LanguageTool.[11]

[11] https://github.com/languagetool-org/languagetool/releases/tag/v3.2

Using a language model tuned to the academic domain, we use perplexity to evaluate the stylistic validity and fluency of a complemented sentence. We favored n-gram language models over neural language models for reproducibility and calculated the score in the same manner described in Section 3.4.3.

3.6 Results

Table 3.6 shows the performance of the baseline models. We observed that the ED-ND model outperforms the other models across most evaluation metrics. This finding suggests that the Enc-Dec noising methods induced noise closer to drafts in SMITH than the heuristic noising.

The current state-of-the-art GEC model showed higher precision but lower recall in $F_{0.5}$. This suggests that the SentRev task requires the model to make a more drastic change in the drafts than in GEC. In addition, the GEC model, trained in the general domain, showed the worst performance in perplexity. This indicates that the general GEC model did not reflect academic writing style upon revision and that the SentRev task requires academic domain-aware rewriting.

Table 3.7 shows examples of the models' output. In the first example, the ED-ND model drastically revised the draft. The middle example demonstrates that our models replaced the <∗> token with plausible words. The last example is the case where our model underperformed by making erroneous edits such as changing *"Chart4"* to *"Figure2"*, and suggesting odd content (*"relation between model and gold standard and piason"*). This may be due to inadvertently introducing noise while generating the training datasets. Appendix .1 shows more examples of generated sentences. We analyzed the performance of the ED-ND baseline model by error types using ERRANT. The results are shown in Figure 3.8. Overall, typical grammatical errors such as noun number errors or orthographic errors are well corrected, but the model struggles with drastic revisions ("OTHER" type errors).

3.7 Related work

Some of the information overlaps with what was presented in Chapter 2, but we reorganize the works related to the SentRev task.

3.7.1 Writing assistance in the academic domain

Several shared tasks for supporting academic writing have been organized in NLP. The Helping Our Own (HOO) 2011 Pilot Shared Task (Dale and Kilgarriff, 2011) aimed to promote the development of tools and techniques to assist authors in writing, with a specific focus on writing within the NLP community. The Automated Evaluation of Scientific Writing (AESW) Shared Task (Daudaravičius, 2015) was organized to promote tools to help write scientific papers. The HOO dataset was created by finding errors in published papers and editing the errors. AESW contains a collection of text extracts from journal papers before and after proofreading. Rather than adding finishing touches to almost completed sentences, our task is to convert unfinished, rough drafts into complete sentences. In addition, these works tackled the task of the *identification* of errors, while SentRev goes further by *rewriting* the drafts.

Other datasets for revisions are available in the academic domain (Lee and Webster, 2012; Tan and Lee, 2014; Zhang et al., 2017). Thus, we provide a notable contribution by exploring the methods to create a dataset of revisions with a scalable crowdsourcing approach. By contrast, Zhang et al. (2017) recruited 60 students over two weeks, and Lee and Webster (2012) collected data from a language learning project where over 300 tutors reviewed academic essays written by 4500 students.

3.7.2 Grammatical error correction

GEC is the task of correcting errors in text, such as spelling, punctuation, grammar, and word choice (Ng et al., 2014; Yuan and Briscoe, 2016). GEC falls within the *editing* and *proofreading* stages of the writing process, while SentRev encompasses GEC and a broader range of text generation (e.g., increasing the fluency and complementing missing information). Napoles et al. (2017) and Sakaguchi et al. (2016) explored fluency edits to correct grammatical errors and to make the text more "native sounding." This direction is similar to SentRev, but more adjustments are needed in SentRev.

3.7.3 Style transfer

Style transfer is the task of rephrasing the text in accordance with specific stylistic properties while preserving the text's original semantic content (Logeswaran et al., 2018; Prabhumoye et al., 2018). From the perspective of automatic academic writing assistance, assistance systems are required to convert nonacademic-style drafts into academic-style drafts. This type of transfer is regarded as a subproblem in the *revising* stage of the writing process.

3.7.4 Text infilling

The drafts in the *revising* stage may contain gaps denoted with <∗>. This setting is similar to *text infilling* Zhu et al. (2019), masking-based language modeling Devlin et al. (2019); Fedus et al. (2018), or the *sentence completion task* Zweig et al. (2012), where the models have to replace mask tokens with plausible words. Systems for these tasks are expected to keep all the original tokens unchanged and only fill the <∗> token with one or more other tokens. This is a different design from SentRev.

3.8 Conclusion

We proposed the SentRev task, where an incomplete, rough draft sentence is revised into a more fluent, complete sentence in the academic domain. We created the SMITH dataset with crowdsourcing for the development and evaluation of this task and built baseline models using a synthetic training dataset. We believe this task can increase the effectiveness of the academic writing process. While this chapter has provided an intrinsic evaluation of the models, the subsequent chapters will investigate whether the model for the SentRev task improves the non-native English speakers' writing.

Table 3.7 Examples of the output from the baseline models. Bold text indicates tokens introduced by the model.

Draft	The global modeling using the reinforcement learning in all documents is our work in the future .
H-ND	The global modeling **of** the reinforcement learning **using** all documents **in** our work **is** the future .
ED-ND	**In our future work , we plan to explore the use of** global modeling **for** reinforcement learning in all documents .
GEC	Global modelling using reinforcement learning in all documents is our work in the future .
Reference	The global modeling using reinforcement learning for a whole document is our future work .
Draft	Also , the above <*> efficiently calculated by dynamic programming .
H-ND	Also , the above **results are calculated** efficiently by dynamic programming .
ED-ND	Also , the above **probabilities are calculated** efficiently by dynamic programming .
GEC	Also , the above **is** efficiently calculated by dynamic programming .
Reference	Again , the above equation can be efficiently computed by dynamic programming .
Draft	Chart4 : relation model and gold % between KL and piason .
H-ND	**Table 1 : Charx-** relation between **gold and piason and KL** .
ED-ND	**Figure 2 : CharxDiff** relation between **model and gold standard and piason** .
GEC	Chart4 : relation model and gold % between KL and person .
Reference	Table 4 : KL and Pearson correlation between model and gold probability .

Chapter 4

Langsmith: An Interactive Academic Text Revision System

4.1 Introduction

Diversity and inclusion in the natural language processing (NLP) community are encouraged. In fact, at the latest NLP conference at the time of writing[1], papers were submitted from more than 50 countries. However, one obstacle can limit this diversity: *The papers must be written in English.* Writing papers in English can be a daunting task, especially for inexperienced, non-native speakers. These writers often struggle to put their ideas into words.

To address this problem, we built the *Langsmith* editor, an assistance system for writing NLP papers in English.[2] The main feature in Langsmith is a revision function, which suggests fluent, academic-style sentences based on writers' rough, incomplete drafts.

The drafts might be so rough that it becomes challenging to understand the user's intended meaning to use as inputs. In addition, several potentially plausible revisions can exist for the drafts, especially when the input draft is incomplete.

Based on such difficulties, our system provides two ways for users to customize the revision: the users can (i) request specific revisions, and (ii) select a suitable revision from diverse candidates (Figure 4.1). In particular, the request stage allows users to specify the parts that require intensive revision.

[1] The 58th Annual Meeting of the Association for Computational Linguistics

[2] See https://www.youtube.com/channel/UCjHeZPe0tT6bWxVVvum1bFQ for the screencast.

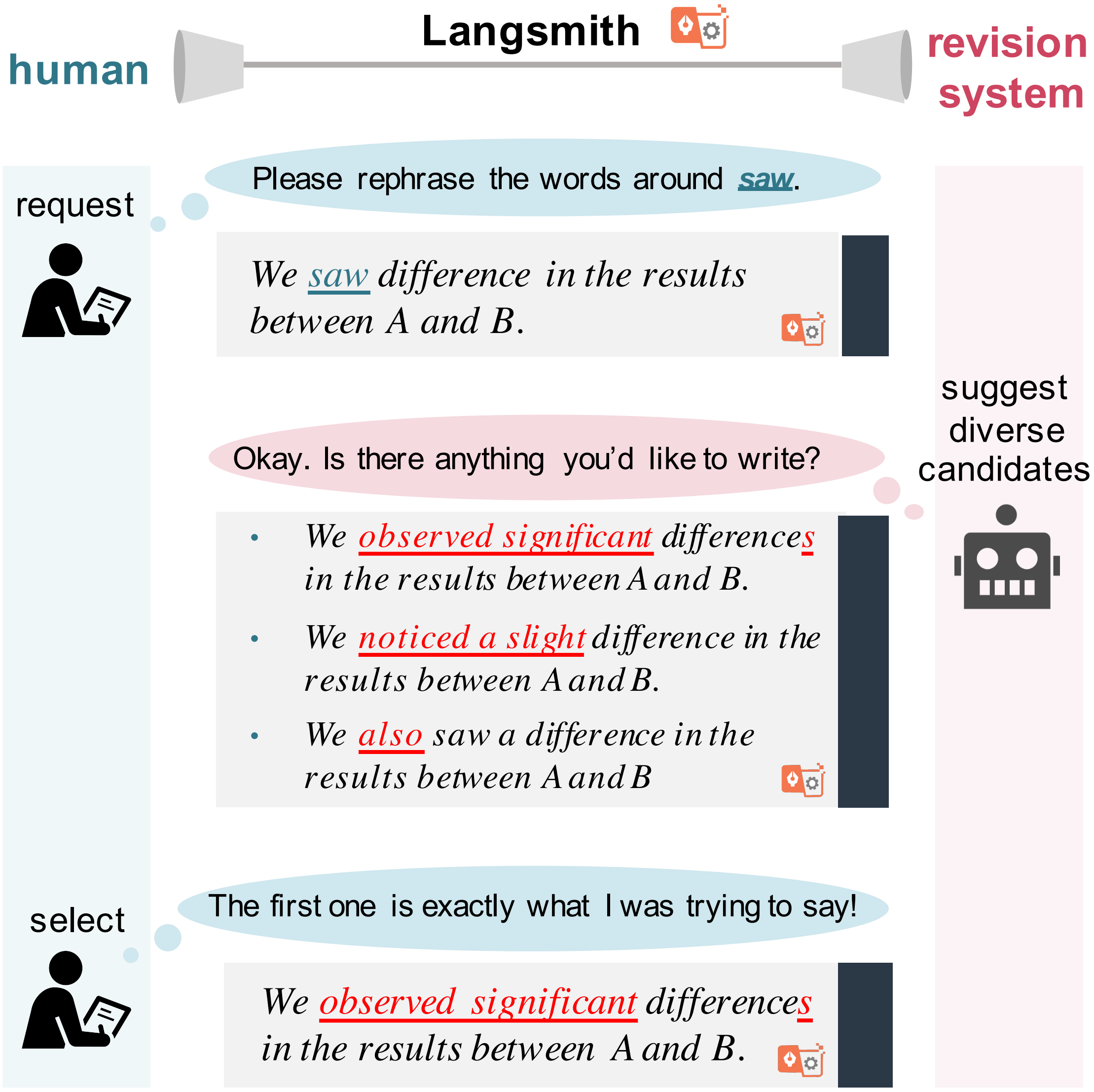
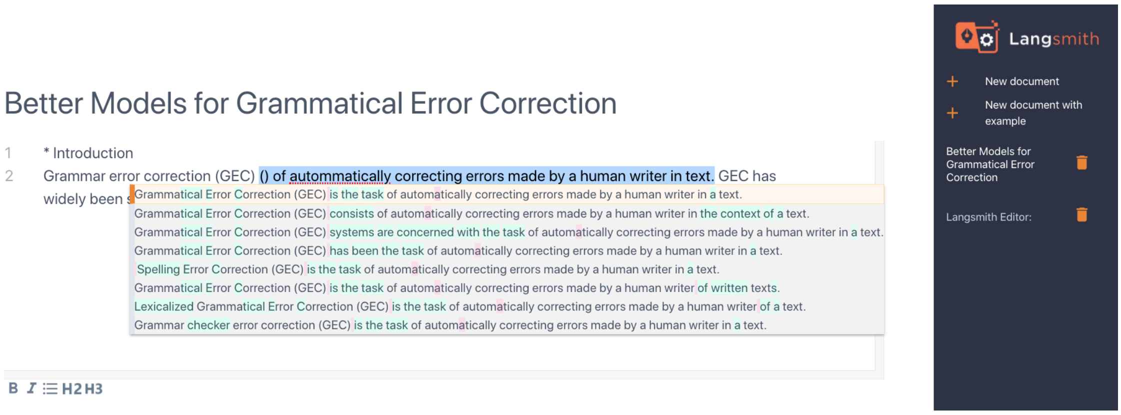

Langsmith
human
revision system
request
Please rephrase the words around saw.
We saw difference in the results between A and B.
suggest diverse candidates
Okay. Is there anything you'd like to write?
We observed significant differences in the results between A and B.
We noticed a slight difference in the results between A and B.
We also saw a difference in the results between A and B
select
The first one is exactly what I was trying to say!
We observed significant differences in the results between A and B.

Figure 4.1 An overview of interactively writing texts with a revision system.

Our experiments demonstrate the effectiveness of our system. Specifically, students whose first language is Japanese, which differs greatly from English, managed to write better drafts when working with Langsmith.

Langsmith has other assistance features as well, such as text completion with a neural language model. Furthermore, the communication between the server and the web frontend is achieved via a protocol specialized in writing software called the Text Editing Assistance Smartness Protocol for Natural Language (TEASPN) (Hagiwara et al., 2019). We hope that



Better Models for Grammatical Error Correction

1 * Introduction

2 Grammar error correction (GEC) () of autommatically correcting errors made by a human writer in text. GEC has widely been s

Grammatical Error Correction (GEC) is the task of automatically correcting errors made by a human writer in a text.

Grammatical Error Correction (GEC) consists of automatically correcting errors made by a human writer in the context of a text.

Grammatical Error Correction (GEC) systems are concerned with the task of automatically correcting errors made by a human writer in a text.

Grammatical Error Correction (GEC) has been the task of automatically correcting errors made by a human writer in a text.

Spelling Error Correction (GEC) is the task of automatically correcting errors made by a human writer in a text.

Grammatical Error Correction (GEC) is the task of automatically correcting errors made by a human writer of written texts.

Lexicalized Grammatical Error Correction (GEC) is the task of automatically correcting errors made by a human writer of a text.

Grammar checker error correction (GEC) is the task of automatically correcting errors made by a human writer in a text.

B I ☰ H2 H3
Langsmith
+ New document
+ New document with example
Better Models for Grammatical Error Correction
Langsmith Editor:

Figure 4.2 Screenshot of Langsmith. The revision feature suggests various revisions for the input "*Grammar error correction (GEC) () of automatically correcting errors made by a human writer in text.*" The characters highlighted in green are added to the original sentence, and the red points indicate tracked deletions.

our system will help the NLP community and researchers, especially those lacking a native command of English.[3]

4.2 The *Langsmith* editor

4.2.1 Overview

This section presents Langsmith, a web-based text editor for academic writing assistance (Figure 4.2). The system has the following three features: (i) text revision, (ii) text completion, and (iii) a grammatical/spelling error checker. These features are activated when users select a text span, type a word, or push a special key.

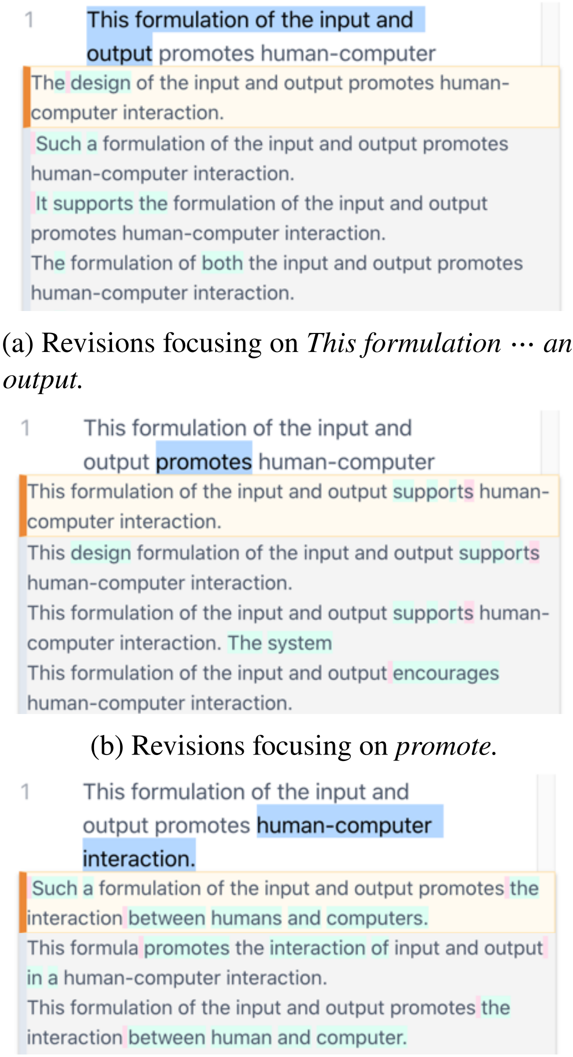
As a case study, this work focuses on paper writing in the NLP domain. Thus, each assistance feature is specialized in the NLP domain. The following sections explain the details of each feature.

4.2.2 Revision feature

The revision feature, the main feature of Langsmith, suggests better sentences in terms of fluency and style for a given draft sentence (Figure 4.2). This feature is activated when the user selects a sentence or smaller unit.

Writers sometimes struggle to put their ideas into words. Thus, the input draft for the revision systems can be incomplete or less informative. Based on such a challenging situa-

[3] This paper was also written using Langsmith.



1 This formulation of the input and output promotes human-computer

The design of the input and output promotes human-computer interaction.

Such a formulation of the input and output promotes human-computer interaction.

It supports the formulation of the input and output promotes human-computer interaction.

The formulation of both the input and output promotes human-computer interaction.

(a) Revisions focusing on *This formulation ··· and output.*

1 This formulation of the input and output promotes human-computer

This formulation of the input and output supports human-computer interaction.

This design formulation of the input and output supports human-computer interaction.

This formulation of the input and output supports human-computer interaction. The system

This formulation of the input and output encourages human-computer interaction.

(b) Revisions focusing on *promote.*

1 This formulation of the input and output promotes human-computer interaction.

Such a formulation of the input and output promotes the interaction between humans and computers.

This formula promotes the interaction of input and output in a human-computer interaction.

This formulation of the input and output promotes the interaction between human and computer.

(c) Revisions focusing on *human–computer interaction.*

Figure 4.3 The focus of the revision depends on the parts selected by users.

tion, we examine the REQUEST and SELECT framework to help users discover sentences that better match what the user wanted to write.

Request stage

Langsmith provides two ways for users to request a specific revision, which can prevent unnecessary revisions from being provided to the user.

First, users can specify where the system should intensively revise a text.[4] That is, when a part of a sentence is selected, the system intensively rephrases the words around the selected part.[5] Figure 4.3 demonstrates the change of the revision focus, depending on the selected text span. Note that controlling the revision focus was not explored in the original sentence-level revision task (chapter 3). This feature is also inspired by Grangier and Auli (2018).

Second, users can insert placeholder symbols, "()", at specific points in a sentence. The system revises the sentence by replacing the symbol with an appropriate expression regarding its context. The input for the revision in Figure 4.2 also has the placeholder symbol. Here, for example, the symbol is replaced with "the task."

Select stage.

The system provides several revisions (Figure 4.2). Note that there is typically more than one plausible revision in terms of fluency and style, in contrast to correcting surface-level errors (Napoles et al., 2017).

The diversity of the output revisions is encouraged using diverse beam search (Vijayakumar et al., 2018). In addition, these revisions are ordered by a language model that is fine-tuned for NLP papers. That is, revisions with lower perplexity are listed in the upper part of the suggestion box. Furthermore, the revisions are highlighted in colors, which makes it easier to distinguish the characteristics of each revision.

Implementation of revision feature

We trained the revision model using the slightly modified version of the synthetic training data introduced in Chapter 3. They created several types of synthetic training data with several noising methods; (i) heuristic noising method, (i) grammatical error generation, (iii) style removal, and (iv) entailed sentence generation. We used the data created by the heuristic noising method, style removal, and the entailed sentence generation for training the revision model. Note that we did not use the data generated by the grammatical error generation because grammatical error correction feature was implemented separately from the revision feature in Langsmith.

We attached the edit marks to the subpart of the training data generated by the style removal method. Let $x_{1:N} = (x_1, x_2, \cdots, x_N)$ and $y_{1:T} = (y_1, y_1, \cdots, y_M)$ be an input sentence with N tokens and its revision with M tokens, respectively. Here x was the synthetic draft

[4] The system performs sentence-level revisions. Hence the users are instructed to select the non-sentence-crossing area.

[5] We allow the system to correct the parts outside the selected span because sometimes the revision for a specific part requires another adjustment for the other parts.

sentence generated by the style removal method from y. The training dataset consists of the pairs of (x, y).

For each (x, y), we first determined if each word in x was rewritten compared to y. We assumed that a token $x_i \in x$ was rewritten if a token with the same lemma as x_i was not in $\{y_j | \max(0, i-3) \leq j \leq \min(M, i+3)\}$. Here we obtained a sequence $c \in \{0, 1\}^N$, where each element c_i corresponds to whether the token x_i was rewritten or not. If x_i was written in y, c_i is 1; otherwise c_i is 0. Then, we defined a score $r(c)$ for each (x, y) as follows:

$$r(c) = \frac{\sum_{i=1}^{N} c_i}{|c|}$$

where $|\cdot|$ returns the length of the vector. If $r(c) > 0.4$, we did not attach the edit marks.

When $r(c) \leq 0.4$, we obtained a span $s = (a, b)$ for x and c as follows:

$$\operatorname*{argmax}_{(a,b)\in\mathcal{S}} \sum_{i=a}^{b} c'_i - \sum_{i=0}^{a-1} c'_i - \sum_{i=b+1}^{N+1} c'_i$$

$$\begin{aligned} \text{where} \quad c'_i &= \begin{cases} 10 & (c_i = 1) \\ 0 & (i = 0, N+1) \\ -1 & (\text{otherwise}) \end{cases} \\ \mathcal{S} &= \{(a, b) \mid a, b \in 1, \cdots, N, a \leq b\} \end{aligned}$$

Based on the obtained $s = (a, b)$, we inserted `<?` before the token x_a, and `?>` after the token x_b. We included the data with special symbols added by such a procedure in the training data.

When the users select a subsequence of a sentence in Langsmith, the edit marks are attached to the input sentence. For example, if the user selects a span "promote" in the sentence "This formulation of the input and output promotes human-computer interaction.", the input to the revision feature is formatted as follows: `This formulation of the input and output <? promotes ?> human-computer interaction.`

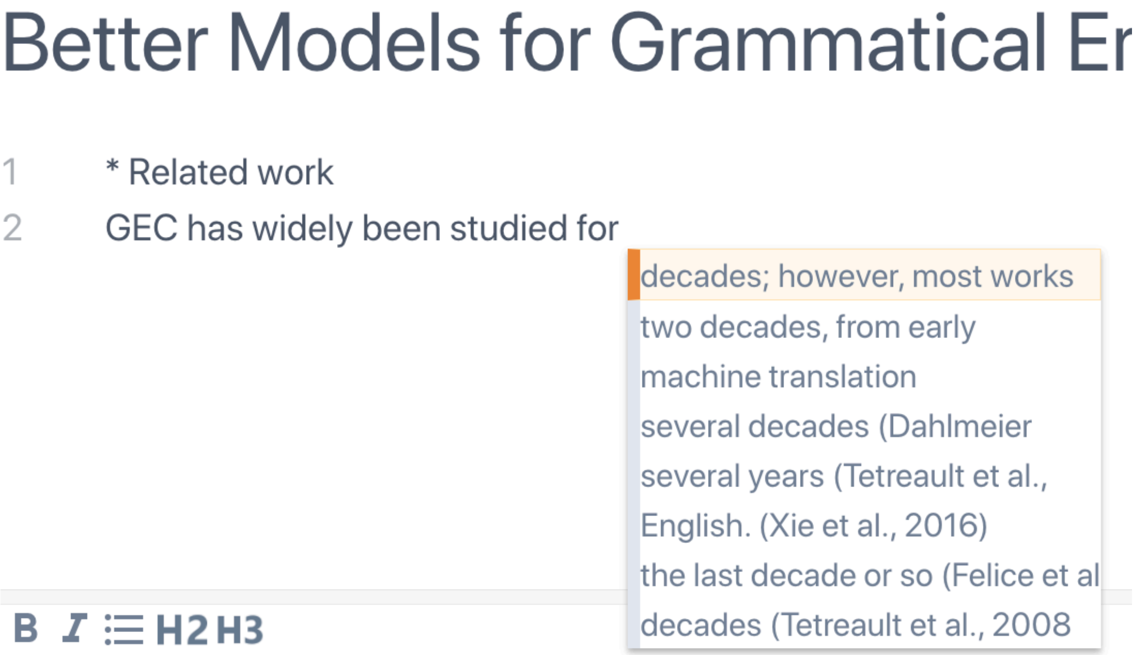
Table 4.1 shows the hyperparameters of the revision model. In the decoding phase, we used the diverse beam search (Vijayakumar et al., 2018). Beam size is set to 15. The diverse beam group and the diverse beam strength are 15 and 1.0, respectively.

Specifically, we first obtained top-15 hypotheses, and then these hypotheses were reranked by the language model. Here, the language model considers 20 tokens in the left context and 20 tokens in the right context beyond the sentence. We excluded the hypotheses with a perplexity greater than 1.3 times the perplexity of the input. We finally showed the

Table 4.1 Hyperparameters of the revision feature.

Fairseq model	architecture	lightconv_iwslt_de_en
Optimizer	algorithm	Adam
	learning rate	5e-4
	adam epsilon	1e-08
	adam betas	(0.9, 0.98)
	weight decay	0.0001
	clip norm	0.0
Learning rate scheduler	type	inverse_sqrt
	warmup updates	4000
	warmup init lrarning rate	1e-7
	min learning rate	1e-9
Training	batch size	24,000 tokens
	updates	1,050,530 steps

top-8 revisions re-ranked to the users. The language model used for re-ranking is the same as the model used for the completion feature (Section 4.2.3).



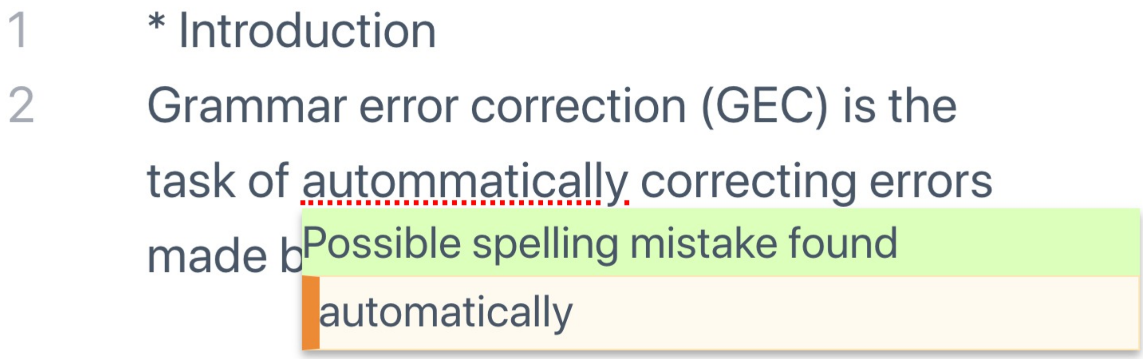
Better Models for Grammatical Er

* Related work

GEC has widely been studied for

- decades; however, most works
- two decades, from early
- machine translation
- several decades (Dahlmeier
- several years (Tetreault et al.,
- English. (Xie et al., 2016)
- the last decade or so (Felice et al
- decades (Tetreault et al., 2008

Figure 4.4 An example of the completion feature. These suggestions are conditioned by the left context, section name (*Related work*) and the paper title (*Better Models for Grammatical Error Correction.*)



* Introduction

Grammar error correction (GEC) is the task of autommatically correcting errors made b

Possible spelling mistake found

automatically

Figure 4.5 The interface of the error correction feature. Errors are automatically highlighted with a red line. The corrections are suggested when the user hovers over the highlighted words.

4.2.3 Other features

Completion feature

When the user presses the `Tab` key, the completion feature generates plausible preceding phrases from the cursor point (Figure 4.4). This feature can consider the paper title and section name as well as the text to the left of the cursor.

We used GPT-2 small (117M) (Radford et al., 2019). To fine-tune the pre-trained GPT-2 on academic domain, we collected 234,830 PDFs of the papers published in ACL Anthology[6] by 2019. Then, we used GROBID (GRO, 2022) for extracting the text information from the PDF files. Table 4.2 shows the training data format. The title name is omitted with 20% probability, and the order of the sections in the same paper was shuffled. Table 4.3 shows the hyperparameters for fine-tuning. We used an implementation in Transformers (Wolf et al., 2019), and used nucleus sampling (Holtzman et al., 2020) with $p = 0.97$ to generate the texts.

Error correction feature.

We used LanguageTool,[7] an open-source grammatical/spelling error correction tool. Each time the text changes, this feature is called upon. The detected errors are then automatically highlighted with red lines (Figure 4.5). The corrections are listed when the user hovers over the highlighted words.

[6] https://www.aclweb.org/anthology

[7] https://github.com/languagetool-org/languagetool/releases/tag/v3.2

Table 4.2 The format of the training data for the completion model.

```
@ Title @

* Section name
Texts in the section
...

* Section name
Texts in the section
⟨|endoftext|⟩

@ Title (of another paper) @
...
```

4.2.4 Protocol

Langsmith was developed based on the TEASPN Software Development Kit (Hagiwara et al., 2019).[8] TEASPN defines a set of APIs for writing software (e.g., text editors) to communicate with servers that implement NLP technologies (e.g., revision model). We extended the protocol to convey title and section information in the completion feature. Since Langsmith is a browser-based tool and frequently communicates with a web server running models, we used WebSocket to achieve smooth communication.

4.3 Experiments and results

We demonstrate the effectiveness of human–machine interactions in revising drafts implemented in our system. We also check whether the REQUEST stage in the revision feature works adequately.

4.3.1 On the revised draft quality

Settings. We suppose a situation where a person writes a draft in their native language (non-English language), translates it to English, and then revises it further to create an English-language draft. In order to simulate this situation, we first collected Japanese-language version of the abstract sections from eight Japanese peer-reviewed journals.[9] Then, the abstracts

[8] https://github.com/teaspn/teaspn-sdk

[9] We used the journals accepted at https://www.anlp.jp/en/index.html.

Table 4.3 Hyperparameters for fine-tuning LMs.

Model	architecture	gpt2
Optimizer	algorithm	Adam
	learning rate	5e-5
	adam epsilon	1e-8
	adam betas	(0.9, 0.999)
	weight decay	0.0
	clip norm	1.0
Learning rate scheduler	type	linear
	warmup updates	0
	max learning rate	5e-5
	total epochs (just used for scheduling)	100
Training	batch size	262,144 tokens
	updates	138,300 steps

were translated into English with an off-the-shelf translation system[10]. We considered the translated abstracts as first drafts. The task is to revise the first drafts. Expert translators created reference final drafts from the Japanese versions of the drafts.[11] We evaluated the quality of the revised versions by comparing them with the corresponding final drafts.

We compared three versions of revised drafts to evaluate the effectiveness of Langsmith:

- one fully and automatically revised by Langsmith (MACHINE-ONLY revision)
- one revised by a human writer without Langsmith (HUMAN-ONLY revision), and
- one revised by a human writer using assistance features in Langsmith (HUMAN&MACHINE revision).

The following paragraphs explain how we obtained the above three versions of the revisions. Table 4.4 shows the statistics of the drafts collected.

Machine-only revision. We automatically applied the revision feature to the drafts (each sentence) without the REQUEST and Select stages. For each sentence, the revision with the highest generation probability was selected.[12] We created one MACHINE-ONLY revision for each first draft.

[10] https://translate.google.co.jp

[11] We used https://www.ulatus.com/.

[12] The hyperparameters for decoding revisions were the same as the revision feature in Langsmith. Reranking with the language model was also employed.

Table 4.4 Statistics of the drafts. The scores are averaged over the drafts. The values following "±" denote the standard deviation of the scores. The column "word type" shows the number of types of the tokens used in the drafts.

drafts	length	word types
Final drafts (reference)	199 ± 52	108 ± 17
HUMAN&MACHINE	192 ± 40	101 ± 17
HUMAN-ONLY	192 ± 43	100 ± 16
MACHINE-ONLY	199 ± 58	105 ± 22
First drafts	202 ± 56	104 ± 22

Table 4.5 Comparison of the revision quality. The scores are averaged over the corresponding revisions. Higher scores indicate that the drafts are closer to the final drafts.

Condition	BLEURT
HUMAN&MACHINE	**-0.08**
HUMAN-ONLY	-0.14
MACHINE-ONLY	-0.18
First drafts	-0.36

Human-only revision. Human writers revise a given first draft. The writers can only access to the error correction feature. This setting simulates the situations that writers typically face.

Human&Machine revision. Human writers revise a given first draft with full access to the Langsmith features.

Human writers. We asked 16 undergraduate and master's students at an NLP laboratory to revise the first drafts in terms of fluency and style. The students were Japanese natives, representatives of the inexperienced researchers in a country where the spoken language is considerably different from English. Each participant revised two different first drafts, one with the HUMAN-ONLY setting and the other one with the HUMAN&MACHINE setting.

Half of the participants first revised a draft with the HUMAN-ONLY setting, and then revised another draft with the HUMAN&MACHINE setting; the other half performed the same task in the opposite order. Ultimately, we collected two HUMAN&MACHINE revisions and two HUMAN-ONLY revisions for each first draft.

Table 4.6 Results of the user study about (I)-(VI). The scores denote the percentage of the participants who chose the option.

Q.	Strongly agree	Slightly agree	Slightly disagree	Strongly disagree
(I)	87.5	12.5	0.0	0.0
(II)	50.0	50.0	0.0	0.0
(III)	62.5	31.3	6.3	0.0
(IV)	12.5	50.0	31.3	6.3
(V)	75.0	12.5	6.3	6.3
(VI)	43.8	43.8	12.5	0.0

Comparison and results

We compared the quality of the three versions of the revised drafts: MACHINE-ONLY revision, HUMAN-ONLY revision, and HUMAN&MACHINE revision. We compared the revised drafts with their corresponding final draft using BLEURT (Sellam et al., 2020), the state-of-the-art automatic evaluation metric for natural language generation tasks.[13] BLEURT is designed to evaluate the similarity of a given sentence pair. Thus, we first split each draft into sentences, and each sentence in the first drafts was aligned with the most similar sentence in the corresponding final draft. Sentence splitting was achieved by spaCy. Note that the references have been created so that the sentence separation does not change from the original first draft. Finally, we calculated the similarity of each sentence pair with BLEURT, and averaged the results. Note that the score is not in the range [0, 1], and a higher score means that the revision is closer to the final draft.

Table 4.5 shows that HUMAN&MACHINE revisions were significantly better[14] than MACHINE-ONLY and HUMAN-ONLY revisions. The results suggest the effectiveness of human–machine interaction achieved in Langsmith. Since this experiment was relatively small in scale and only used an automatic evaluation metric, we will conduct a larger-scale experiment with human evaluations in the future.

4.3.2 User study

After the experiments outlined in Section 4.3.1, we asked the participants about the usability of Langsmith. The 16 participants were instructed to evaluate the following statements:

(I) Langsmith was more helpful than the Baseline environment for the revision task.

[13] We used BLEURT-Base with 128 max tokens: https://storage.googleapis.com/bleurt-oss/bleurt-base-128.zip.

[14] We applied a bootstrap hypothesis test (Koehn, 2004), and the score of HUMAN&MACHINE was significantly higher than the HUMAN-ONLY and MACHINE-ONLY scores ($p < 0.05$).

Table 4.7 Results of the user study about helpful features. The scores denote the percentage of the participants who chose the feature (multiple choice question).

Feature	percentage
revision	100
completion	31.3
correction	62.5

(II) Comparing the text written by the two environments, the text written with Langsmith was better.

(III) The feature of specifying where to intensively revise was helpful.

(IV) The placeholder feature in the revision feature was helpful.

(V) Providing more than one output from the revision feature was helpful.

(VI) Providing more than one output from the completion feature was helpful.

The participants evaluated the statements (I)-(VI) on a four-point scale: (a) strongly agree, (b) slightly agree, (c) slightly disagree, and (d) strongly disagree. In addition, the participants answered whether each feature was helpful in writing.

Results

Tables 4.6 and 4.7 show the results of our user study. From the responses to (I) and (II), we observed that the users were satisfied with the writing experience with Langsmith. The responses to (III), (IV), and (V) support the idea that our Request and Select stages are helpful. Here, using the place holders was relatively not helpful. The responses to (VI) also suggest that showing several candidates does not bother the users. Table 4.7 displays the result of whether each feature was helpful in writing. The result indicates that the revision feature was the most useful for creating drafts using the implemented features.

4.3.3 Sanity check of the Request stage

Finally, we checked the validity of our method to control the revision based on the selected part of the sentence (Figure 4.3).

Settings

We randomly collected 1,000 sentences from the first drafts created with the translation system. In each sentence with T tokens $x = (w_1, \cdots, w_T)$, we randomly inserted edit marks to specify a certain span $s = (i, j)$ in x $(1 \leq i < j \leq T, 1 \leq j - i \leq 5)$. Specifically, special tokens were inserted before w_i and after w_j in x. We denote the input sentence with these edit marks as x^{edit}. We then obtained 10-best outputs of the revision system $(y_1^{\text{edit}}, \cdots, y_{10}^{\text{edit}})$ for each x^{edit}. Here, these output sentences were generated through the diverse beam search with the same settings as the revision feature in Langsmith. We calculated the following score for each input sentence and its revisions:

$$r = |\{\, y_k^{\text{edit}} \mid x_{i:j} \in \text{ngram}(y_k^{\text{edit}}), 1 \leq k \leq 10\}|$$

where $x_{i:j}$ denotes the subsequence $(w_i, \cdots, w_j)$ in x. The function ngram$(\cdot)$ returns a set of all the n-grams of a given sequence. A lower r indicates that the subsequence specified with the edit marks are more frequently rephrased.

We also obtained a score r' for each x. r' was calculated using the input without the edit marks x and its 10-best outputs y_k. We compared r and r' for each x.

Results

We observed that r frequently[15] had lower values than r'. That is, a certain subsequence was more rephrased by the revision system when it had the edit marks than when it did not. These results validate our approach of controlling the revision focus, which is implemented in the REQUEST stage of the revision feature.

4.4 Conclusions

We have presented Langsmith, an academic writing assistance system. Langsmith provides a writing environment, in which human writers use several assistance features to improve the quality of texts. Our experiments suggest that our system is useful for inexperienced, non-native writers in revising English-language papers.

Publicly available paid version supports academic domains other than natural language processing. Other features now available include an example sentence search feature and a sentence comparison feature that determines which of two sentences is more fluent.

[15] We conducted the one-side sign test. The difference is significant with $p \leq 0.05$.

Chapter 5

How Non-native English speakers use AI-powered writing-support tools

5.1 Introduction

English has become the lingua franca of the world, bringing inevitable disadvantages to non-native English speakers (NNESs), especially those living in countries with low English proficiency, in numerous aspects of work and business. Academia is one such industry where English dominates. For NNES researchers, writing an academic manuscript in English can be a daunting task that requires the ability to write concisely, clearly, and fluently, without spelling or grammatical errors. Helping NNESs overcome such language barriers is an important step toward achieving diversity and inclusion in academia (Khelifa et al., 2022).

AI-powered writing-support tools play an important role in diminishing these language barriers. The typical current design of such tools involves machines suggesting potential revisions (e.g., correcting grammatical errors and improving the wording) to a user-written draft sentence. However, the users, regardless of their English abilities, must themselves determine whether such revisions are compatible with their writing goals. This raises several questions about NNESs in these situations, including why they will accept or reject AI-provided revisions, and how they develop trust in these AI-powered tools. Understanding how NNESs interact with AI-powered writing-support systems and develop trust/distrust in this process is an important step toward designing better human-AI collaborative writing systems. To date, the use of AI-powered writing tools has been studied extensively in the area of human-computer interaction (HCI) (Buschek et al., 2021; Clark et al., 2018; Coenen et al., 2021; Lee et al., 2022). Yet, little of this research has focused on NNESs.

In this study, we investigate NNESs behaviors and their mental models while using an AI-powered writing-support system. We focus on a rewriting tool, Langsmith (Chapter 4) [1], representing AI-powered writing-support systems. Similar to other writing-support systems, Langsmith suggests potential revisions in terms of grammaticality, fluency, and style once a draft sentence is an input by the user. First, we conducted a preliminary investigation of numerous available writing-assistance systems often employed by NNESs, alongside Langsmith, to gain important insights, in line with those recently reported by Liebling et al. (2021), who found that NNESs employed other resources, including search systems and dictionaries, to verify MT results. Our preliminary survey of NNES Langsmith users in Japan [2] revealed that the most used MT in conjunction with this rewriting tool. Based on this finding, we conducted user studies and interviews to gain a detailed understanding of NNESs Langsmith usage and thought processes when using Langsmith. We divided the NNESs participant into two groups, namely, those who did and did not simultaneously apply MT tools to ameliorate their writing, as the use of MT tools is likely to have an impact on how users assess the revisions recommended by Langsmith. We investigated differences in the decision-making between these two groups, about whether and how to adopt Langsmith's suggestions, and explored the factors influencing their trust in it.

Our findings suggest that NNESs using MT tend to face more difficulty making appropriate selections on their own when assessing rewriting suggestions. As a result, they rely on other resources, such as the score provided by the tool and back-translation. They are more likely to lose trust in the tool when discovering evident errors. Based on these and other findings, we suggested various avenues for improving AI-based NNES writing-support.

5.2 Related work

5.2.1 AI-powered writing-support systems

To overcome the language barriers to writing and publishing activity, researchers proposed and developed writing-support systems specifically for academic papers written in English by NNESs (Daudaravičius, 2015). Dale and Kilgarriff (2011) introduced a grammatical-error detection and correction system. Various systems have also been developed to rewrite texts into fluent academic English, as discussed in Chapter 2.

[1] http://langsmith.co.jp/

[2] Japan is one of the countries with low English proficiency according to the EF English proficiency index (https://www.ef.com/wwen/epi/regions/asia/japan/).

In addition to writing-support tools that emerged from pure research, numerous commercial tools have been developed and are already on the market. Dale and Viethen (2021) surveyed more than 50 of these commercial writing systems and categorized them into three types: pattern-matching style checkers, autocompletion tools, and rewriting tools. Pattern-matching style checkers generally identify errors with high precision based on large sets of manually produced rules. However, they are unable to detect and correct errors that are not included in their rules. In contrast, autocompletion and rewriting tools do not require manually generated rules, but instead tend to introduce more errors when predicting user input or proposing corrections. Rewriting tools suggest rewriting candidates that enhance style and fluency beyond traditional grammar and spelling corrections. Rewriting research has been an active area in the field of NLP. A variety of automatic rewriting techniques have been developed with various aims, including effective paraphrasing (Li et al., 2018), stylistic change (Jin et al., 2022b), improving fluency (Napoles et al., 2017), and simplification (Nisioi et al., 2017). The range of choice of commercial tools in this category, such as Wordtune[3] and Langsmith (Chapter 4), is likewise increasing. Many of these tools employ an interface that suggests multiple alternatives to human-written drafts, and the human author then selects the one that best fits his or her intentions.

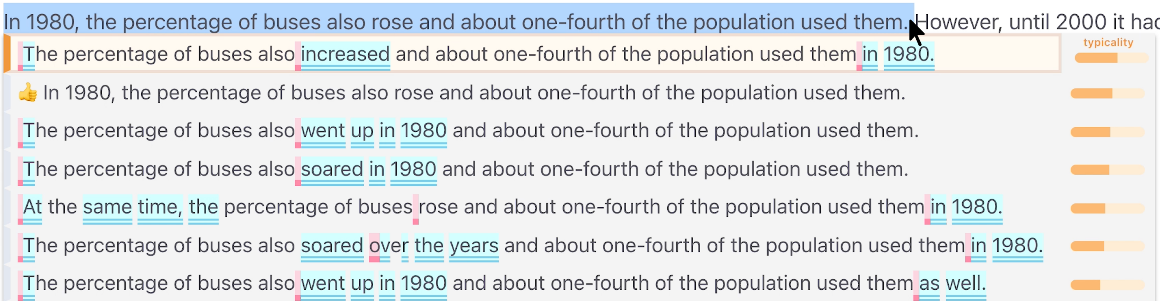
Numerous AI-powered writing-support systems display multiple suggestions in their user interfaces (Buschek et al., 2021; Coenen et al., 2021; Ito et al., 2020a; Lee et al., 2022). Chen and Tseng (2022) proposed a decision model for presenting multiple outputs from NLP models. They recommend the UI presenting multiple outputs when language generation models are applied, or when the outputs of NLP models are shown directly to the users. Generally, multiple suggestions are arranged in the order of their generation probability. Several UIs simultaneously display scores, such as the generation probability. For example, the AllenNLP language model demo site [4] shows the outputs of NLP models and their scores. Such scores are displayed in the hope that they will improve the users' performance in evaluating and adopting model outputs (Feng and Boyd-Graber, 2019).

5.2.2 Interaction between NNESs and AI-powered writing tools

Although the performance of writing-support systems is improving dramatically, their recommendations sometimes contain errors (He et al., 2021). Moreover, unlike pattern-matching systems and human editors, AI-powered writing tools are unable to provide users with reasons for their suggestions. Therefore, even if a probability score for each suggestion is presented, users must decide which suggestion to accept based on information extrinsic to

[3] https://www.wordtune.com/

[4] https://demo.allennlp.org/next-token-lm



In 1980, the percentage of buses also rose and about one-fourth of the population used them. However, until 2000 it hac

The percentage of buses also increased and about one-fourth of the population used them in 1980.

👍 In 1980, the percentage of buses also rose and about one-fourth of the population used them.

The percentage of buses also went up in 1980 and about one-fourth of the population used them.

The percentage of buses also soared in 1980 and about one-fourth of the population used them.

At the same time, the percentage of buses rose and about one-fourth of the population used them in 1980.

The percentage of buses also soared over the years and about one-fourth of the population used them in 1980.

The percentage of buses also went up in 1980 and about one-fourth of the population used them as well.

Figure 5.1 Screenshot of Langsmith. The typicality bar indicates how natural a sentence is, and is calculated by a language model. The blue and red highlights indicate where text has been added or removed, respectively.

the system: often, a pre-existing personal knowledge of English, which is inevitably lower among NNESs than among native speakers.

Studies indicated that NNESs have difficulty thinking and making judgments in English (Cohen and Brooks-Carson, 2001), and therefore tend to rely on MT when writing in English. According to Aranberri (2020); Lee (2020); Lee and Briggs (2021), MT aids the NNESs' English-writing process by allowing them to write in their native language and ultimately helps them write better English than on their own.

Despite these findings indicating that NNESs tend to use their native language during their English-writing process and increasingly rely on MTs, research on human interaction with AI-powered writing-support systems has mostly focused on native English speakers (Buschek et al., 2021; Clark et al., 2018; Coenen et al., 2021; Lee et al., 2022). Among few exceptions, Buschek et al. (2021) found that NNESs were more accepting of AI-powered writing tool suggestions than native English speakers. However, the reason behind this result remains unclear, indicating that there is a lack of understanding of how NNESs are affected by and use AI-powered writing tools. To address this gap in the existing research, our study conducted a behavioral study on the users of an AI-powered writing tool.

5.3 Research tool: Langsmith

We adopted Langsmith (Chapter 4), an AI-powered writing-support tool specifically tuned for academic English (Figure 5.1), as the focus of our research. The main users are NNES Japanese researchers and students.

Langsmith offers two modes: autocompletion and rewriting. In the former, it predicts the rest of the sentence that a user is typing when the user presses the tab key. In turn, the latter transforms the user's writing into more fluent expressions. For each sentence that the user selects using the cursor, Langsmith suggests multiple rewritten options, as shown in Figure 5.1. If the user selects part of a sentence, Langsmith intensively suggests rewritten options for that part.[5]

In addition, the rewriting mode offers two features to assist users in their writing. When a sentence with a placeholder "()" inserted is selected, Langsmith fills in the placeholder as part of its rewriting process. Furthermore, Langsmith provides a "typicality score" for each rewrite candidate (orange bars shown in Figure 5.1). Each score is based on the generation probability of that sentence as calculated by the relevant neural language model. Thus, the higher the typicality score, the more common the sentence is in the language model's training data. This score is often used to assess sentence fluency in NLP (Kann et al., 2018; Yang et al., 2018). This typicality score feature has been introduced after the experiment of Chapter 4.

5.4 Preliminary study

Our preliminary study is aimed at investigating how Langsmith users typically use the tool to write academic papers, and what other tools they use in conjunction with it in their writing process. We started with a log analysis of Langsmith users to identify which of the two modes (autocompletion and rewriting) and corresponding features were most used. Then, we constructed a short survey to determine which tools besides Langsmith users employed to help them write academic papers in English. This survey was sent to individuals on the Langsmith user mailing list and also posted on Twitter. The 39 users that responded were not compensated. Each of these phases of is described in further detail in the following subsection.

5.4.1 Log analysis results

To determine the most employed modes and features of Langsmith in writing processes, we analyzed the users' session logs. Each such log includes the function name and number of times the function was called on the Langsmith website. We analyzed 6,860 sessions from April 26th to September 16th, 2021, and found that within that period, the rewriting mode was used around 50 times more frequently per session than the autocompletion mode (M = 9.80, SD = 18.72 vs. M = 0.19, SD = 1.70). We also found that the placeholder feature of the rewriting service was rarely used, i.e., only 0.1% of total rewriting time.

[5] Langsmith may rewrite areas outside of the selected range if the rewrite of the part requires another adjustment or if there are errors.

5.4.2 Survey results

In the survey, we asked about the tools the respondents usually used when writing papers in English and how they used Langsmith. We received responses from 39 Japanese adults (33 males and 6 females), whose average age was 32 (range: 22–54). They included thirteen faculty members, nine Master's students, nine Ph.D. students, one industry researcher, five public institution researchers, and two others.

Survey results indicated that 85% of the respondents used MT tools such as DeepL and Google Translate in conjunction with Langsmith. A total of 75% used grammatical error correction tools like Grammarly and Trinka. More than 70% of the respondents also reported using online/offline resources (e.g., Hyper Collocation and Power Thesaurus) to look up example sentences and alternative expressions. To the question of how they usually use Langsmith, exactly half of those surveyed answered that they used other editors to create English text and copied them into Langsmith as needed; under a third responded that they pasted MT output into Langsmith for editing; and five stated that they wrote sentences into Langsmith directly.

In summary, the log analysis and survey results of Langsmith users showed that they used the rewriting mode far more frequently than auto-complete mode. Furthermore, the users typically used the rewriting tool in combination with other tools — mostly MT and other rewriting tools — while writing academic English.

These results help us formulate specific research questions and construct the design for our main study. First, we used a customized version of Langsmith, where the autocompletion mode and placeholder feature were disabled. Second, we formed our research questions around how the use of other tools affects NNESs to assess the appropriateness/validity of the rewriting suggestions. Because numerous users employed MTs, we were particularly interested in understanding how this affects their use of rewriting tools and the writing process, and what factors influence trust in the tools.

5.5 Research questions

Our preliminary study and recent literature reviewed in Section 5.2 show that NNESs increasingly rely on MT tools. Thus, we pose the following research question:
***RQ1**: How do NNESs use MT alongside an AI-powered rewriting tool during their writing process?*

Based on the answers to RQ1, we investigate how NNESs use AI-powered rewriting tools. As discussed earlier, when an AI-powered rewriting tool recommends word replacements, grammatical corrections, etc., it is not always possible for NNESs to make in-

formed choices about whether these recommendations are worth adopting. Prior research has reported that NNESs use various strategies to verify MT output, such as using external resources (e.g., dictionaries, other MT software, and back-translation), or asking experts (Liebling et al., 2021). In the case of rewriting tools that provide their suggestions in English, however, it is not clear what strategies NNESs use to determine whether a suggestion should be adopted or rejected. Therefore, we pose the following question:
***RQ2**: How do NNESs decide whether or not to accept the suggestions of an AI-powered rewriting tool? What, if any, is the difference in the approach of NNESs who use MT and those who do not in assessing the suggestions?*

Finally, we are interested in understanding how users form mental models of rewriting tools during their writing process and, in particular, what kinds of suggestions make NNESs feel that the tool is reliable vs. unreliable. This is an important issue, because whether users continue to use a system or not depends on their perception of its usefulness (Davis, 1985). Therefore, the final question:
***RQ3**: What impressions are key to NNESs' development of trust (or distrust) in an AI-powered rewriting tool, and are there differences between those who use MT and those who do not regarding these impressions?*

5.6 Main study

To explore the research questions presented in Section 5.5, we designed the main study, in which NNESs performed English-writing tasks using Langsmith's rewriting mode, followed by an online interview. To ensure that the focus of the main study remains solely on rewriting, we disabled the autocompletion function and placeholder feature.

5.6.1 Procedure

A crowdsourcing platform, CrowdWorks[6] was used to recruit 24 NNESs, of whom 21 completed the writing tasks, as detailed in Section 5.6.3 below. After receiving a briefing on the study and signing consent forms, they were shown a video that explained how to use Langsmith. To familiarize themselves with this tool and the task format, they were asked to perform a brief (approximately 10-minute) practice writing task using Langsmith. Subsequently, they were asked to work on two writing tasks and make screen recordings during their writing process. The tasks were distributed to the participants via Google Docs, and we asked them to write their final answers on the same Google Docs files. The order of

[6] https://crowdworks.jp/

the two tasks was randomized among the participants. As our preliminary study indicates that NNESs use various tools for writing, we allowed the use of other tools besides Langsmith. However, the use of any such tools that could not be captured in screen recordings was forbidden.

Online interviews were conducted within a week after both main tasks were completed, and they were semi-structured. We randomly selected 15 participants and invited them for a follow-up interview, and 14 participated. The interview protocol was designed to encourage the interviewees to reflect on their writing process, including questions about their general practices when writing academic papers, use of writing-support tools, opinions and impressions of Langsmith, and how they assessed/selected the suggestions provided. All interviews were conducted in Japanese; they were audio-recorded, and lasted approximately one hour (range: 56-74 minutes). Compensation for participation in the main study was calculated based on the local pay rates for part-time work: participants who participated only in the writing task received 4,500 yen, and those who participated in both the writing task and the interview received 7,500 yen.

5.6.2 Writing tasks

We adapted all writing tasks from examples of the IELTS Academic Writing Task 1 posted on the website of iPassIELTS (an online IELTS course provider) with the company's permission. Each task consisted of a bar graph and a table, which the participants were asked to describe.[7] The original IELTS Academic Writing Task 1 requires examinees to write a minimum of 150 words, and the estimated time for completing it is 20 minutes.[8] However, when we initially asked two NNES members of our laboratory team to complete a sample task, we found that a 20-minute limit left them very little time to elaborate on their writing using Langsmith. Therefore, we provided a 30 min window for each writing task. However, we did not set a strict time limit, as our principal goal was not to assess the participants' English-writing ability, but rather to obtain a detailed picture of their English-writing process using Langsmith.

5.6.3 Participants

We recruited the participants for the main study via CrowdWorks, a Japanese crowdsourcing service. The participation criteria were: 1) experience in writing academic English within

[7] https://www.ipassielts.com/ielts_training/study_plans_single/leisure-time and https://www.ipassielts.com/ielts_training/study_plans_single/ielts_task1_hotel_occupancy

[8] https://www.ielts.org/for-test-takers/test-format

Table 5.1 Demographic of participants in main study.

Participant	Gender	Profession	Use of MT
P1∗	Male	Ph.D. student	✓
P2	Male	faculty member	✓
P3∗	Female	undergraduate student	
P4∗	Male	faculty member	
P5∗	Female	undergraduate student	
P6∗	Male	Ph.D. student	✓
P7	Female	public institution researcher	
P8∗	Male	public institution researcher	✓
P9	Male	master's student	✓
P10∗	Female	master's student	✓
P11∗	Male	master's student	✓
P12∗	Female	undergraduate student	✓
P13∗	Male	faculty member	
P14∗	Female	Ph.D. student	✓
P15	Male	master's student	✓
P16∗	Male	master's student	✓
P17	Female	Ph.D. student	✓
P18∗	Male	Ph.D. student	✓
P19	Male	industry researcher	✓
P20∗	Male	Ph.D. student	
P21	Female	physician	✓

Note: ∗ indicates those who participated in the interview. ✓indicates the participants using MT.

the previous three years, and/or 2) a plan to write a paper in English within the following year. Of the 24 Japanese students and researchers who were recruited [9], three failed to complete the study due to technological issues, resulting in a final pool of 21 participants (13 males, 8 females) with a mean age of 31.8 (range: 20-47). None of them had prior experience using Langsmith. Three were undergraduate students, four were Master's students, seven were Ph.D. students, one was an industry researcher, two were public institution researchers, three were faculty members, and one was a physician. Nine of the male and five of the female participants (i.e., those marked with asterisks in Table 5.1) participated in the post-task interviews.

During the study, 15 participants (71%) used MT, and 11 (52%) used web searches in addition to Langsmith. Four also used Grammarly, a grammar-error correction tool. Although Grammarly has a rewriting feature, this feature is only available to paid users, and all four of the participants in question used the free plan.[10] Notably, all participants had default access to the spelling- and grammar-correction functions of Google Docs.

5.6.4 Measurement and analysis

We collected screen recording data from 21 participants, and audio data from 14 interviews. To differentiate between the assessments of Langsmith's suggestions between those who used MT and those who did not, we divided our participants into two groups, namely the *MT* and *SELF* group. We analyzed differences in how their respective member sets assessed the suggestions made by Langsmith and formed impressions of it. [11].

Video recordings. We used video recordings to address RQ1 and RQ2. To identify the tools used by participants along with Langsmith (RQ1), we reviewed the video recordings and counted the number of participants using each tool. For MT, we categorized its usage into two types: forward-translation (Japanese → English) and back-translation (English → Japanese). Notably, back-translation is often used in human translation work, and back-translation by MT is also used to verify MT output (Liebling et al., 2021). To address RQ2, we observed the video recordings and identified which tools were used for assessing the Langsmith suggestions. Importantly, the use of MT was limited to back-translation, as Langsmith suggestions were listed in English.

[9] All participants' first language was Japanese.

[10] Plans and feature details of Grammarly: https://www.grammarly.com/plans (accessed on January 9, 2022)

[11] An analysis of the participants' writings is presented in Appendix .2

Interviews. The recorded interviews were transcribed using an automatic-transcription tool. The transcripts were then reviewed and corrected by the first author. We identified themes in the transcripts using an inductive approach (Corbin and Strauss, 2014). Two authors separately analyzed one-third of the transcripts and sorted them into meaningful categories, while identifying relationships between the themes and looking for salient themes. The same two authors then iteratively checked and elaborated on their codes until agreement and saturation were reached. Then, the first author coded the rest of the transcripts.

5.7 Findings

We present our findings organized around our three research questions. The participants' ID numbers include an "-MT" or "-SELF" suffix according to whether that person used MT during the writing process.

5.7.1 NNESs' writing methods (RQ1)

This section describes the *SELF* and *MT* groups' respective writing methods — how they used Langsmith and MT in their writing process (RQ1) — based on 1) our observations of the participants' screen recordings, and 2) the interview data.

Usage of the MT during writing

MT was used by 15 of the 21 participants during writing tasks. Out of the 15 participants, six drafted the full text in Japanese and forward-translated it into English. Seven others drafted the text in Japanese for some parts and in English for others, and forward-translated the Japanese parts into English. Back-translation, on the other hand, was used by 10 participants, eight of whom also used forward-translation. In most cases, these participants would back-translate their English text into Japanese after refining it with Langsmith.

Writing methods

***SELF* group.** All participants in the *SELF* group created the draft themselves entirely in English and revised it with Langsmith. They also occasionally used web searches to look up words. As P4-SELF noted, "I don't think about the text in Japanese when writing scientific papers."

***MT* group.** In contrast, all participants in the *MT* group revealed in their interviews that they used the MT technology regularly, i.e., not only for the writing tasks of this study. As P18-MT explained:

> *"I use machine translation almost all the time, except for expressions that come up many times or that I use often. So I think I rely on the machine translation 80% of the time when I write in English."*

Many participants in the *MT* group created the drafts with forward-translation, and then revised them by repeating Langsmith and back-translation. They also occasionally edited the English text by themselves after back-translation. However, two (P1-MT and P6-MT) never edited the English text by themselves. Instead, the two participants rewrote the source Japanese text and repeated the process of forward-translation, Langsmith, and back-translation until a back-translation result (in Japanese) that they were comfortable with appeared. These participants seemed to follow a similar practice (i.e., repeat forward and backward translations) in their daily English academic writing:

> *"I usually draft some Japanese text, translate it into English using DeepL, and then translate it back into Japanese, again using DeepL. I check to see whether the Japanese is properly translated, and if it is not, I look at what is wrong and correct the sentences one by one."* (P1-MT)

Summary of findings (RQ1)

NNES participants who did not use MT during the writing process rarely used tools other than Langsmith to revise their English text. However, those who used MT in the writing process often wrote sentences in their first language (Japanese) and translated them into English. After refining the English with Langsmith, they further back-translated it and evaluated its validity. These results show that NNESs who use MT in conjunction with Langsmith to write English texts make creative use of MT's forward- and back-translations in combination, before and after using the rewriting tool.

5.7.2 NNESs' assessment of rewriting suggestions (RQ2)

To explore NNESs' decision-making on whether/how they adopt Langsmith's suggestions (RQ2), we first examined video recordings to assess what other tools were used to verify Langsmith's output. Then, we conducted a thematic analysis of NNESs' interview quotes and identified their assessment strategies.

Table 5.2 Tools used to check suggestions, identified from the video recordings

	MT(Back-trans.)	WebSearch	Gram.Checker	Nothing
MT (15)	67% (10)	20% (3)	13% (2)	27% (4)
SELF (6)	-	33% (2)	17% (1)	50% (3)

Note: The values in parentheses are headcounts.

Table 5.3 Assessment strategies of suggestion identified via thematic analysis

	Typicality	Back-trans.	WebSearch	Own proficiency
MT (9)	100% (9)	67% (6)	23% (2)	34% (3)
SELF (5)	60% (3)	-	40% (2)	40% (2)

Note: The values in parentheses are headcounts.

Usage of other tools to check Langsmith output

Table 5.2 lists the tools used by the participants to verify Langsmith's output. This indicates that the *MT* group members tend to back-translate Langsmith's output, while *SELF* group members were more likely to determine the output for themselves.

Assessment strategies

From the interview analysis, we identified four strategies by which the sampled NNESs assessed Langsmith's writing suggestions. These were by consulting 1) Langsmith's typicality score, 2) back-translation, 3) web search results, and 4) their own English knowledge. According to the interviewees, the search engines were used to check the meaning or usage of unclear words. Table 5.3 lists the percentages of interviewees from *MT* and *SELF* groups who claimed to adopt each of these assessment strategies. Overall, members of both groups seemed to rely on the typicality scores provided by Langsmith. However, members of the *MT* group seemed to rely more on system-generated suggestions (by their typicality scores provided by Langsmith or back-translations) than members of the *SELF* group. Specific insights gained from the interviews with each of the *SELF* and *MT* groups are given in the following.

***SELF* group.** Members of the *SELF* group generally appeared to decide whether or not to accept Langsmith's suggestions based on their own preferences. Although some referred to the system-provided typicality score, they did not seem to regard it as particularly important. As P13-SELF put it, "I noticed there was a chart on the right side [i.e., typicality score], but

I didn't pay much attention to it." They referred to the typicality scores only when they were not sure about their own decisions. As P3-SELF commented:

> *"I could usually narrow it down to about one or two sentences, because even if I got a lot of sentences, some of them were a bit different from what I really wanted to say. If I was unsure about two or one so, I would either read them myself to check, or if not sure myself, refer to the credibility section on the right-hand side."*

Both advantages and disadvantages of using the typicality score to assess Langsmith's suggestions were reported by members of the *SELF* group. Especially those who *ipso facto* did not use MT, considered it a positive aspect of the system that it facilitated conversion from English to English. As P3-SELF stated, "I think it's more efficient to keep it in English when revising something written in English." However, P20-SELF reported the negative effects of using a typicality score, reducing self-elaboration:

> *"In a way, it seemed as if the level of my revision was becoming increasingly shallower. At first, I was comparing not only the typicality of the text with my original text, but by the end of the project, I felt like I was unconsciously or without thinking looking for text with a high typicality score."*

***MT* group.** Members of the *MT* group tended to regard Langsmith's typicality scores as more important than their *SELF* group counterparts. Some *MT* members simply adopted the system proposal with the highest typicality score without checking it. For example, P1-MT stated that he was unsure of his English proficiency and thought that the machine judgment was better than his own. Some others expressed a belief that Langsmith's suggestions with smaller typicality scores were completely unworthy of consideration. As P18-MT explained, "I only looked at the top three or four. I didn't look at the bottom of the suggestions because their typicality bars were so small that I didn't think I needed to look at them." However, this is not to suggest that *MT* group members completely trusted Langsmith's typicality scores. Rather, they relied on them due to a perceived lack of any other means of assessment.

> *"I can't judge whether it sounds fluent or not because I'm not a native speaker. I had my doubts about whether the sentence was really fluent, but I selected it."* (P16-MT)

Another characteristic strategy adopted by *MT* group members was back-translation using MT. All participants who stated that they ever used back-translation also back-translated

Table 5.4 Comparison of trust and distrust factors in *MT* and *SELF*

	factors	MT (9)	SELF (5)
trust	quality of suggestions	44% (4)	100% (5)
	variety of suggestions	89% (8)	100% (5)
distrust	obvious errors in suggestions	89% (8)	20% (1)

Note: The values in parentheses are headcounts.

their task text after using Langsmith. Mostly, they said they did this to ensure that Langsmith's output did not contain evident errors or evidently missing information, as it was more efficient for them to check it in Japanese. As P10-MT noted, "I tried it in DeepL first, and if something looked strange, I checked the English text myself."

Some participants also guessed the quality of English sentences provided by Langsmith from the quality of back-translated Japanese. They believed that if their *MT* could translate system-produced English into error-free Japanese, the English itself was also error-free.

Summary of findings (RQ2)

The sampled NNESs' use of a variety of tools for determining Langsmith's output was consistent with the findings of previous research, which indicates that NNESs rely on other resources to check MT results Liebling et al. (2021). Furthermore, *MT* group members tended to assign more importance to Langsmith's typicality scores, and were more likely to use back-translation to make judgments, as compared to *SELF* group members.

5.7.3 NNESs' development of trust in the rewriting tool (RQ3)

RQ3 addressed the factors contributing to NNESs' trust and distrust of Langsmith. To answer this, we conducted a thematic analysis of NNESs' interview quotes and identified the factors shaping their trust and distrust.

Trust and distrust factors

Our coding of the interview data identified two factors contributing to establishing trust, namely, the "quality of suggestions" and "variety of suggestions," as well as one distrust factor, namely "evident errors in suggestions." Table 5.4 shows how the incidence of each of these factors differed across the two groups.

Many interviewees from both groups mentioned that a positive and trust-building aspect of Langsmith was the great variety of suggestions it provided. Further, all interviewees from the *SELF* group pointed to the quality of the system's proposals as a factor that boosted their trust in it. In contrast, many *MT* group members reported finding errors in Langsmith's suggestions, which led to a decrease of their trust in it. These and other findings are discussed in more detail below.

***SELF* group.** All interviewees from the *SELF* group regarded Langsmith's suggestions as high-quality, and stated that their writing was ameliorated by adopting them. In terms of the diversity of such suggestions, they also commented that they had learned new English expressions from the system; some expressed appreciation for the fact that they could select and mix system-generated expressions according to their own preferences. As P5-SELF explained,

> *"I combined the first one with the third one and so on. I also kept what I wanted to keep in my writing, but which reflected Langsmith's suggestions that I thought were better. I liked the fact that I could combine multiple suggestions."* (P5-SELF)

Interestingly, a few participants in the *SELF* group expressed distrust of Langsmith, even when it contained evident errors or made low-quality suggestions. As P4-SELF puts it:

> *"My experience was that when I typed in longer sentences, I had the impression that Langsmith returned suggestions that didn't make much sense, so I figured that my sentence was too long to make sense in the first place."* (P4-SELF)

These quotes indicate that participants, who were able to produce English sentences on their own, were often able to evaluate and take advantage of the various suggestions generated by Langsmith.

***MT* group.** Many interviewees in the *MT* group likewise indicated that Langsmith's diverse recommendations introduced them to new English expressions. However, some commented that it was difficult to select appropriate suggestions and/or to modify them as needed:

> *"If there was a big change, I wondered whether the sentence is weird, which gives me a chance to think about where to fix it. But since I am unable to fix it, I'm unsure of whether I'm making the right choice."* (P12-MT)

Thus, as described in Section 5.7.2, *MT* group participants turned to typicality scores to help them select among Langsmith's various suggestions. Moreover, in contrast to the interviewees from the *SELF* group, some said they appreciated multiple suggestions as a backup in case they did not prefer the top suggestion. As P16-MT put it, "If there was only one suggestion and I was told that it was the best one, I could not do anything about it, even if it looked weird. Langsmith gave me about seven suggestions, so I could look at them from the top down, and if a suggestion seemed wrong, I could choose the next one. I think it was a good feature." Many interviewees in the *MT* group also pointed out clear discrepancies in Langsmith's suggestions to the text they had prepared. This point was rarely made by their *SELF* group counterparts. In particular, some of those who used back-translation commented that they could not immediately identify errors when examining Langsmith's suggestions. However, they did notice them via back-translation, i.e., when they appeared in Japanese.

> *"At first, I fixed each sentence with Langsmith. I thought, 'Oh, that's good work,' and then put it into DeepL and translated it into Japanese. But it turned out that something was missing."* (P1-MT)

As described in Section 5.7.2, participants in the *MT* group tended to be unable to assess suggestions on their own in English, and had to rely on typicality scores. This suggests that they had lesser ability to recognize errors in the suggestions than *SELF* group members. Additionally, when finding errors, some of the *MT* group did not correct them themselves, but relied again on MT to do so.

Summary of findings (RQ3)

The participating Japanese NNESs appreciated that the focal AI-powered rewriting tool made high-quality and varied suggestions. However, those who used MT tended to lose confidence in Langsmith as soon as they found clear output errors. This mostly happened after they back-translated suggested English sentences into Japanese using MT. Participants in the *MT* group tended to evaluate Langsmith's output based on factors other than their English proficiency, resulting in a high error-detection load as well as a loss of trust in the tool when errors were found.

5.8 Discussion and design implications

5.8.1 NNESs' writing strategies

Existing research makes conflicting arguments and suggestions about NNES's use of MT in academic writing. While some studies recommend that NNESs not draft in their native languages nor use MT when writing papers (Wallwork, 2016; Wallwork and Southern, 2020), recent studies (Aranberri, 2020; Lee, 2020; Lee and Briggs, 2021; Tsai, 2020) claim that the use of MT can improve NNESs' writing and also provides educational benefits (Lee, 2021).

Our findings show that many of our participants created their initial drafts in their native language. They further seem to imply that the use of MT is widespread and highly trusted by Japanese NNESs, even in the context of academic writing. We have drawn a few implications from this finding, which we outline below.

First, most prior studies on the interaction with AI-powered writing tools (Buschek et al., 2021; Lee et al., 2022) based their analyses on tool logs, which may not be an appropriate design for understanding the writing behavior of NNESs, given the variety of their tool-use behaviors that we observed. For example, our findings show that the text input of writing tools is not necessarily drafted by the users, but may be the output of other tools (e.g., MT). Therefore, we encourage future developers and evaluators of writing-support systems for NNESs to adopt research designs that comprehensively observe the writing process.

Second, a characteristic usage of MT, namely back-translation, was observed in our study. Back-translation has actually been hailed as an effective strategy for MT-output assessment (Liebling et al., 2021; Miyabe and Yoshino, 2009). For example, recent work on MT for outbound translations — i.e., ones in which the translation is into a language unknown to the MT user — reported that back-translation increased user confidence (Zouhar et al., 2021).

However, the such use of MT may not always be effective. In fact, a majority of MT research has focused on improving translation robustness, even when NNESs make some errors in their original text (Anastasopoulos et al., 2019). This means that MT (including back-translation) can often correctly translate texts containing errors. Such automatic correction of the original text suggests that use of MT is inappropriate to evaluate the correctness of the original text (Wallwork and Southern, 2020).

Using back-translation by MT to assess source text quality may require improvements in aspects other than robustness. For example, if there are errors in the source English text, MT may need to warn the users instead of attempting to translate it into Japanese. Further, to help NNESs correctly assess the appropriateness of English text, technologies of text quality evaluation such as Langsmith's typicality score are important. In NLP, text quality evaluation continues to be an active research area (Zhu and Bhat, 2020), and further development is encouraged.

5.8.2 Feedback for NNESs from AI-powered rewriting tools

Numerous participants in *MT* group selected system-generated English sentences based on their typicality scores, and some ignored sentences that were assigned low scores. One member in the *SELF* group also reported subconscious pressure to act according to typicality scores, which made his own elaboration more shallow. Although these numerical indicators provide a useful basis for NNESs' judgments, they also appear to have become an obstacle to NNESs checking the AI-powered tool's suggestions for themselves. This finding encourages further research on the explanations and feedback for NNESs. For example, providing feedback in natural language on the reasons for corrections (Nagata et al., 2021) or actual examples of phrases and expressions (Kaneko et al., 2022) might be worthwhile. Further, it is important to investigate the kinds of explanations NNESs find most useful.

5.8.3 NNESs' mental models of the AI-powered rewriting tool

The fact that the NNESs who used MT tend to lose trust in the focal AI-powered writing tool when they find errors in its suggestions implies that this subgroup may demand higher precision from such tools, as compared to broadly similar individuals who did not use MT. Furthermore, many of our participating MT-using NNESs indicated that they were not confident or good at English. This dynamic was also reflected in their usual writing strategies. P1-MT, for example, edited his original Japanese draft rather than the English version when he found errors or unintended expressions in the latter. Similarly, P12-MT stated that in the past, she had written papers in Japanese, had them translated into English by an expert human translator, and then asked an English proofreader to revise them before submission. This suggests that NNESs who have (or perceive themselves as having) low English proficiency may avoid writing English as much as possible. Such tendency suggests that they may form mental models of AI-powered rewriting tools differently from the mental models of NNESs, who do not use MT in their writing process. This is a promising avenue for future research, which we hope will be stimulated by the present study.

5.8.4 Diversity of suggestions

Many participants mentioned its diverse output as a positive aspect of Langsmith. In particular, many participants in the *SELF* group combined some of the multiple suggestions from

Langsmith. This tool uses an algorithm called diverse beam search (Vijayakumar et al., 2018), which achieves diverse outputs by adding noise to the probability distribution of the model's outputs. Due to the nature of the algorithm, high noise strength produces a variety of outputs but also increases the probability of errors. Several studies have been conducted on other algorithms to achieve diverse outputs, but many of them have also yielded slower or degraded performance (Ippolito et al., 2019; Luo and Shakhnarovich, 2020). According to our findings, participants in the *MT* group faced difficulty in assessing suggestions, indicating that diversity and quality must be balanced to support the NNES.

Evaluating diversity is not technically easy, and few NLP tasks have added diversity to their evaluation criteria (Tevet and Berant, 2021). Our findings highlight the need for further research on the improvement of algorithms and evaluation of models when such diversity is a goal. Further behavioral research in HCI must examine NNESs trade-offs among diversity, quality, and speed.

5.8.5 NLP-powered integrated writing assistance

When working with Langsmith to produce academic English writing, participating Japanese NNESs relied on several other tools and frequently switched between various applications and websites. In particular, many participants in the *MT* group switched back and forth between MT and Langsmith several times. It is reasonable to expect that such switching can lead to increases in the workload (Pilzer et al., 2020), and could make the assessment of AI-powered tool's suggestions more demanding. To improve usability, an integrated writing environment for NNESs must be developed. To achieve this, we believe that it is important to unify protocols across NLP models and editors (Hagiwara et al., 2019), and promote the development of interfaces that allow efficient access to NLP functions (Yang et al., 2019).

5.8.6 Limitations

Our study has several limitations that should be the focus of future research. First, the study setup differed from most independent academic writing. Specifically, the participants in our main study had previously never used Langsmith, and their usual writing practices may be different from the one observed in our study.

Furthermore, the participants were not allowed to use paper dictionaries and any other tools that could not appear on-screen (including secondary digital devices). This may have created further deviation from their common writing practice. Therefore, we hope to conduct future research over a longer period and in more realistic settings.

Second, we analyzed NNESs' behavior based on their writing process, specifically by dividing them into two groups according to whether they used MT. This examined the impact of MT use on the rewriting process. In the future, however, it will be important to analyze data grouped by English proficiency. We found several differences between the two groups. The use or non-use of MT may be related to their English proficiency, and thus all of the above differences may be attributed to their English proficiency or confidence in English. Indeed, prior research has found that NNESs with low English proficiency tend to translate from their native language rather than writing directly in a second language (Cohen and Brooks-Carson, 2001) and employ MT more frequently than those with higher English proficiency (Tsai, 2020). Furthermore, we targeted only Japanese researchers and students, and a more comprehensive study of NNESs would necessarily involve participants with a wider array of linguistic, cultural, and occupational backgrounds.

Finally, this is a case study based on Langsmith alone. While we believe that our findings will guide the future development of AI-powered writing-support systems for NNESs, research focused on other NLP models and interfaces could contradict our findings. Therefore, we emphasize the need for ongoing research as this technology develops.

5.9 Conclusion

We examined how one AI-powered writing-support tool, Langsmith, was used and perceived by NNES Japanese researchers in the context of English-language paper writing. We first investigated what other tools are used for this purpose and found that many participants supplemented their use of rewriting tools with MT. Further, we conducted user studies and interviews with these researchers to understand how they assessed Langsmith's rewriting suggestions and what factors prompted them to trust or distrust these suggestions.

Our results suggest that the NNESs who used MT tended to rely on sources of information other than their personal English proficiency to evaluate the English output of an AI-powered writing-support tool. Probably as a consequence, we observed that they tended to discover errors in the rewriting tool's suggestions at a relatively late stage in their writing tasks, and that their trust in the tool plummeted upon noticing evident errors.

In summary, the results of this study imply that interaction between NNESs and the focal AI-powered writing tool may be restricted by the language barrier. While AI-powered writing tools may help them become aware of new words and expressions, reliable verification processes remain time-consuming and expensive. We hope that our results will motivate both the HCI and NLP research communities to take up the challenge of developing writing-support tools specifically for NNESs.

Chapter 6

Conclusion and Future Work

In this thesis, we aimed to assist NNESs or academic writing in English. We proposed a new rewriting task to convert drafts into more fluent text, created an evaluation dataset, and investigated how to construct pseudo-training data to build the rewriting model (Chapter 3). Then, we built a writing support system, Langsmith, which incorporates the rewriting model, and released it to the public (Chapter 4). Finally, we investigated the effectiveness and usage of Langsmith (Chapter 5). As described in Chapter 2.2, although various writing assistance tools are now available to the public, Langsmith was one of the earliest tools released among them.

Although we demonstrated that Langsmith helped NNES improve their writing in the Chapter 4 experiment, subsequent qualitative analysis revealed that NNESs struggle to assess Langsmith's suggestions and reflect them in their original document. In addition, we found that many NNESs use machine translation to produce their drafts. Machine translation systems have also improved dramatically in performance over the past few years, and numerous machine translation applications (e.g., Google Translation and DeepL) are now available. Our findings in Chapter 5 suggest that NNESs may be able to start their revisions from a more fluent English draft than we assumed when we designed the rewriting task in Chapter 3.

6.1 Future work

Based on our findings and recent works on NLP and HCI, we discuss promising directions for NNESs writing support.

Expansion of the datasets for writing supports: Although a number of datasets on various revisions have been proposed in recent years, the rewriting task remains a low-resource setting because drafts are not publicly available, and there are multiple possible candidates for reasonable revisions, as discussed in Chapter 3. Recently created datasets have used approaches such as extracting differences from multiple versions submitted to paper archives (Du et al., 2022; Jiang et al., 2022), having professional editors edit manuscripts to create differences (Mita et al., 2022), and creating drafts with pseudo-noise similar to our method (Dong et al., 2021). Although various datasets are now available, and we have access to a wide variety of editing data at the time of writing, there is no doubt that many processes have not been addressed previously. Our ultimate goal is to support the entire writing process for non-native English speakers. Data collection and developing protocols and frameworks are important issues supporting the entire process.

User log analysis and personalization: There may be a gap between the input we assume into the system and the actual sentence the user inputs. In addition, the input that should be assumed may depend on the user's abilities. Indeed, our study revealed that some users draft their own drafts and others use machine translation to draft their drafts. Machines and humans could make errors differently. In addition, assistance will be required at different stages of the writing process (Sarrafzadeh et al., 2021). In general, when data is not included in the training data, the performance of the model may be adversely affected. A more in-depth and continuous analysis of Langsmith's user logs could reveal more about how non-native English speakers write and what kind of assistance they seek. Analyzing user behavior is important not only for Langsmith but also for research on writing assistants, including rewriting task design, methods for creating rewriting task datasets, and designing other writing support systems.

Human-in-the-loop approach to building and evaluating writing support systems: As discussed above, building a dataset for rewriting is not easy, and personalization according to the user's ability and writing style is an important issue. To solve these challenges, the Reinforcement Learning from Human Feedback (RLHF) approach (Stiennon et al., 2020; Ziegler et al., 2019) may be one effective direction. RLHF is a framework that human evaluates the model's output against the real input of the user, and the model is trained based on that feedback. In this approach, the model can be trained using real user inputs. We believe using RLHF for continuous system improvement is worthwhile because Langsmith has already acquired a certain number of users. For the success of the RLHF, improvements in reinforcement learning algorithms are important, as well as practices regarding how to evaluate NLG models by humans and collect efficient and appropriate human feedback.

Reference-free automatic text evaluation: In Chapter 3, we created an evaluation dataset and evaluated the model's performance using reference-based evaluation metrics. Reference-based evaluation metrics are a standard evaluation method in many NLG tasks (Sai et al., 2022). However, reference-based evaluation metrics typically have the disadvantage that is creating reference text is costly. Therefore, a new evaluation dataset must be created whenever a model in a different domain is created. A reference-free evaluation metric would reduce the cost of the evaluation. Various reference-free evaluation metrics have been proposed recently, and their performance has been improved (Rei et al., 2021). Furthermore, reference-free evaluation metrics can also be used for filtering and re-ranking against the output of NLG systems (Fernandes et al., 2022), such as Langsmith's typicality score (Chapter 5). A reference-free evaluation metrics can be helpful not only for the evaluation of NLG models but also for helping NNESs asses system suggestions.

Beyond the surface-level suggestions: Support for information and logical structure is important for better writing (King, 2012). No one would dispute the importance of such assistance, but how to implement it computationally is still unclear and is a longstanding issue in NLP (Dou et al., 2022). Langsmith also provides only superficial support. As a first step, we believe it is essential to develop a method for automatically evaluating the logical structure of sentences by classifying or scoring good sentences in terms of logical structure and other factors. If automatic evaluation with high accuracy becomes possible, it may lead to constructing more writing support models using reinforcement learning and data augmentation approaches.

6.2 NLP and HCI Collaboration

Finally, we emphasize the importance of cross-disciplinary research spanning natural language processing and human-computer interaction to promote research and social diffusion of applications using natural language processing, especially natural language generation.

Human reactions and processes change when the environment changes due to new systems and other factors. For example, research had shown that human writing processes also changed when they changed from handwriting to typewriting (Haas, 1989; Lutz, 1987). AI writing assistance could also change the process and style of human writing. Therefore, a better understanding of the human writing process is important for more effective AI writing assistance. We also believe that case studies must be accumulated to understand the human process. Our research will be one of the most important case studies for building better writing support systems.

Collaboration between NLP and HCI is important not only for writing support systems but also for various NLG systems (Heuer and Buschek, 2021). With the development of deep learning technology, NLG systems are able to generate very fluent sentences, which have the potential to be applied to a variety of applications. Furthermore, the recent emergence of large-scale language models has made it possible to handle various tasks simply by adjusting the prompts. These developments will likely lead to the development of many more NLG applications. In developing applications, it is important to consider how users will use them and the development of their interfaces. Furthermore, collaboration with HCI is important in terms of extrinsic evaluation, which examine the usefulness of the NLG model to users in a real-life scenario. We hope more collaboration between NLP and HCI will lead to more research on various applications, including writing support tools.

References

(2008–2022). GROBID. https://github.com/kermitt2/grobid.

Al-Thanyyan, S. S. and Azmi, A. M. (2021). Automated Text Simplification: A Survey. *ACM Comput. Surv.*, 54(2).

Anastasopoulos, A., Lui, A., Nguyen, T. Q., and Chiang, D. (2019). Neural Machine Translation of Text from Non-Native Speakers. 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)*, pages 3070–3080, Minneapolis, Minnesota. Association for Computational Linguistics.

Aranberri, N. (2020). With or without you? Effects of using machine translation to write flash fiction in the foreign language. In *Proceedings of the 22nd Annual Conference of the European Association for Machine Translation*, pages 165–174, Lisboa, Portugal. European Association for Machine Translation.

Boisson, J., Kao, T.-H., Wu, J.-C., Yen, T.-H., and Chang, J. S. (2013). Linggle: a Web-scale Linguistic Search Engine for Words in Context. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 139–144, Sofia, Bulgaria. Association for Computational Linguistics.

Bowman, S. R., Angeli, G., Potts, C., and Manning, C. D. (2015). A Large Annotated Corpus for Learning Natural Language Inference. In *Proceedings of EMNLP*, pages 632–642.

Brill, E. and Moore, R. C. (2000). An Improved Error Model for Noisy Channel Spelling Correction. In *Proceedings of ACL*, pages 286–293.

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., and Amodei, D. (2020). Language Models are Few-Shot Learners. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H., editors, *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Bryant, C., Felice, M., and Briscoe, T. (2017). Automatic Annotation and Evaluation of Error Types for Grammatical Error Correction. In *Proceedings of ACL*, pages 793–805.

Buchman, M., Moore, R., Stern, L., and Feist, B. (2000). *Power Writing: Writing with Purpose*, volume 4.

Buschek, D., Zürn, M., and Eiband, M. (2021). The Impact of Multiple Parallel Phrase Suggestions on Email Input and Composition Behaviour of Native and Non-Native English Writers. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, CHI '21, page 1–13, New York, NY, USA. Association for Computing Machinery.

Chen, E. and Tseng, Y.-H. (2022). A Decision Model for Designing NLP Applications. In *Companion Proceedings of the Web Conference 2022*, WWW '22, page 1206–1210, New York, NY, USA. Association for Computing Machinery.

Chen, M. X., Lee, B. N., Bansal, G., Cao, Y., Zhang, S., Lu, J., Tsay, J., Wang, Y., Dai, A. M., Chen, Z., Sohn, T., and Wu, Y. (2019). Gmail smart compose: Real-time assisted writing. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Miningg (KDD'19)*, page 2287–2295.

Clark, E., Ross, A. S., Tan, C., Ji, Y., and Smith, N. A. (2018). Creative Writing with a Machine in the Loop: Case Studies on Slogans and Stories. In *23rd International Conference on Intelligent User Interfaces*, IUI '18, page 329–340, New York, NY, USA. Association for Computing Machinery.

Coenen, A., Davis, L., Ippolito, D., Reif, E., and Yuan, A. (2021). Wordcraft: a Human-AI Collaborative Editor for Story Writing.

Cohen, A. D. and Brooks-Carson, A. (2001). Research on Direct versus Translated Writing: Students' Strategies and Their Results. *The Modern Language Journal*, 85(2):169–188.

Corbin, J. and Strauss, A. (2014). *Basics of Qualitative Research: Techniques and Procedures for Developing Grounded Theory*. SAGE Publications.

Dahlmeier, D., Ng, H. T., and Wu, S. M. (2013). Building a Large Annotated Corpus of Learner English: The NUS Corpus of Learner English. In *Proceedings of the Eighth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 22–31, Atlanta, Georgia. Association for Computational Linguistics.

Dale, R. and Kilgarriff, A. (2011). Helping our own: The HOO 2011 pilot shared task. In *Proceedings of the 13th European Workshop on Natural Language Generation*, pages 242–249, Nancy, France. Association for Computational Linguistics.

Dale, R. and Viethen, J. (2021). The automated writing assistance landscape in 2021. *Natural Language Engineering*, 27(4):511–518.

Dang, H., Benharrak, K., Lehmann, F., and Buschek, D. (2022). Beyond Text Generation: Supporting Writers with Continuous Automatic Text Summaries. In *Proceedings of the 35th Annual ACM Symposium on User Interface Software and Technology*, UIST '22, New York, NY, USA. Association for Computing Machinery.

Daudaravičius, V. (2015). Automated Evaluation of Scientific Writing: AESW Shared Task Proposal. In *Proceedings of the Tenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 56–63, Denver, Colorado. Association for Computational Linguistics.

Davis, F. D. (1985). *A technology acceptance model for empirically testing new end-user information systems: Theory and results*. PhD thesis, Massachusetts Institute of Technology.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2019). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of NAACL-HLT*, pages 4171–4186.

Dong, Q., Wan, X., and Cao, Y. (2021). ParaSCI: A Large Scientific Paraphrase Dataset for Longer Paraphrase Generation. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 424–434, Online. Association for Computational Linguistics.

Dou, Y., Forbes, M., Koncel-Kedziorski, R., Smith, N. A., and Choi, Y. (2022). Is GPT-3 Text Indistinguishable from Human Text? Scarecrow: A Framework for Scrutinizing Machine Text. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7250–7274, Dublin, Ireland. Association for Computational Linguistics.

Du, W., Raheja, V., Kumar, D., Kim, Z. M., Lopez, M., and Kang, D. (2022). Understanding Iterative Revision from Human-Written Text. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3573–3590, Dublin, Ireland. Association for Computational Linguistics.

Dwivedi-Yu, J., Schick, T., Jiang, Z., Lomeli, M., Lewis, P., Izacard, G., Grave, E., Riedel, S., and Petroni, F. (2022). EditEval: An Instruction-Based Benchmark for Text Improvements. arXiv.

Edunov, S., Ott, M., Auli, M., and Grangier, D. (2018). Understanding Back-Translation at Scale. In *Proceedings of EMNLP*, pages 489–500.

Fedus, W., Goodfellow, I., and Dai, A. M. (2018). MaskGAN: Better Text Generation via Filling in the _. In *International Conference on Learning Representations*.

Felice, M., Bryant, C., and Briscoe, T. (2016). Automatic Extraction of Learner Errors in ESL Sentences Using Linguistically Enhanced Alignments. In *Proceedings of COLING*, pages 825–835.

Feng, S. and Boyd-Graber, J. (2019). What Can AI Do for Me? Evaluating Machine Learning Interpretations in Cooperative Play. In *Proceedings of the 24th International Conference on Intelligent User Interfaces*, IUI '19, page 229–239, New York, NY, USA. Association for Computing Machinery.

Fernandes, P., Farinhas, A., Rei, R., De Souza, J., Ogayo, P., Neubig, G., and Martins, A. (2022). Quality-Aware Decoding for Neural Machine Translation. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1396–1412, Seattle, United States. Association for Computational Linguistics.

Flesch, R. (1948). A new readability yardstick. *Journal of Applied Psychology*, 32(3):221 – 233.

Flowerdew, J. (2007). The non-Anglophone scholar on the periphery of scholarly publication. *AILA review*, 20(1):14–27.

Grangier, D. and Auli, M. (2018). QuickEdit: Editing text & translations by crossing words out. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 272–282, New Orleans, Louisiana. Association for Computational Linguistics.

Haas, C. (1989). Does the Medium Make a Difference? Two Studies of Writing With Pen and Paper and With Computers. *Human–Computer Interaction*, 4(2):149–169.

Hagiwara, M., Ito, T., Kuribayashi, T., Suzuki, J., and Inui, K. (2019). TEASPN: Framework and Protocol for Integrated Writing Assistance Environments. 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): System Demonstrations*, pages 229–234, Hong Kong, China. Association for Computational Linguistics.

He, J., Peng, B., Liao, Y., Liu, Q., and Xiong, D. (2021). TGEA: An Error-Annotated Dataset and Benchmark Tasks for TextGeneration from Pretrained Language Models. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 6012–6025, Online. Association for Computational Linguistics.

Heafield, K. (2011). KenLM: Faster and Smaller Language Model Queries. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, pages 187–197, Edinburgh, Scotland. Association for Computational Linguistics.

Heuer, H. and Buschek, D. (2021). Methods for the Design and Evaluation of HCI+NLP Systems. In *Proceedings of the First Workshop on Bridging Human–Computer Interaction and Natural Language Processing*, pages 28–33, Online. Association for Computational Linguistics.

Holtzman, A., Buys, J., Du, L., Forbes, M., and Choi, Y. (2020). The Curious Case of Neural Text Degeneration. In *Proceedings of the 8th International Conference on Learning Representations (ICLR 2020).*

Huang, J. C. (2010). Publishing and learning writing for publication in English: Perspectives of NNES PhD students in science. *Journal of English for Academic Purposes*, 9(1):33–44.

Ippolito, D., Kriz, R., Sedoc, J., Kustikova, M., and Callison-Burch, C. (2019). Comparison of Diverse Decoding Methods from Conditional Language Models. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3752–3762, Florence, Italy. Association for Computational Linguistics.

Ito, T., Kuribayashi, T., Hidaka, M., Suzuki, J., and Inui, K. (2020a). Langsmith: An Interactive Academic Text Revision System. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 216–226, Online. Association for Computational Linguistics.

Ito, T., Kuribayashi, T., Kobayashi, H., Brassard, A., Hagiwara, M., Suzuki, J., and Inui, K. (2019). Diamonds in the Rough: Generating Fluent Sentences from Early-Stage Drafts for Academic Writing Assistance. In *Proceedings of the 12th International Conference on Natural Language Generation*, pages 40–53, Tokyo, Japan. Association for Computational Linguistics.

Ito, T., Kuribayashi, T., Kobayashi, H., Brassard, A., Hagiwara, M., Suzuki, J., and Inui, K. (2020b). Assisting authors to convert raw products into polished prose. *Journal of Cognitive Science*, 21(1):103–140.

Jiang, C., Xu, W., and Stevens, S. (2022). arXivEdits: Understanding the Human Revision Process in Scientific Writing. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Jin, D., Jin, Z., Hu, Z., Vechtomova, O., and Mihalcea, R. (2022a). Deep Learning for Text Style Transfer: A Survey. *Computational Linguistics*, 48(1):155–205.

Jin, D., Jin, Z., Hu, Z., Vechtomova, O., and Mihalcea, R. (2022b). Deep Learning for Text Style Transfer: A Survey. *Computational Linguistics*, 48(1):155–205.

Kaneko, M., Takase, S., Niwa, A., and Okazaki, N. (2022). Interpretability for language learners using example-based grammatical error correction. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7176–7187, Dublin, Ireland. Association for Computational Linguistics.

Kann, K., Rothe, S., and Filippova, K. (2018). Sentence-Level Fluency Evaluation: References Help, But Can Be Spared! In *Proceedings of the 22nd Conference on Computational Natural Language Learning*, pages 313–323, Brussels, Belgium. Association for Computational Linguistics.

Kessler, J. (2017). Scattertext: a Browser-Based Tool for Visualizing how Corpora Differ. In *Proceedings of ACL 2017, System Demonstrations*, pages 85–90, Vancouver, Canada. Association for Computational Linguistics.

Khelifa, R., Amano, T., and Nuñez, M. A. (2022). A solution for breaking the language barrier. *Trends in Ecology & Evolution*, 37(2):109–112.

King, C. L. (2012). Reverse Outlining: A Method for Effective Revision of Document Structure. *IEEE Transactions on Professional Communication*, 55(3):254–261.

Kingma, D. P. and Ba, J. (2015). Adam: A Method for Stochastic Optimization. In *Proceedings of ICLR*.

Koehn, P. (2004). Statistical significance tests for machine translation evaluation. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing (EMNLP 2004)*, pages 388–395.

Lee, J. and Webster, J. (2012). A Corpus of Textual Revisions in Second Language Writing. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (ACL 2012)*, pages 248–252.

Lee, M., Liang, P., and Yang, Q. (2022). CoAuthor: Designing a Human-AI Collaborative Writing Dataset for Exploring Language Model Capabilities. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*, CHI '22, New York, NY, USA. Association for Computing Machinery.

Lee, S.-M. (2020). The impact of using machine translation on EFL students' writing. *Computer Assisted Language Learning*, 33(3):157–175.

Lee, S.-M. (2021). The effectiveness of machine translation in foreign language education: a systematic review and meta-analysis. *Computer Assisted Language Learning*, 0(0):1–23.

Lee, S.-M. and Briggs, N. (2021). Effects of using machine translation to mediate the revision process of Korean university students' academic writing. *ReCALL*, 33(1):18–33.

Li, Z., Jiang, X., Shang, L., and Li, H. (2018). Paraphrase Generation with Deep Reinforcement Learning. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3865–3878, Brussels, Belgium. Association for Computational Linguistics.

Lichtarge, J., Alberti, C., Kumar, S., Shazeer, N., Parmar, N., and Tong, S. (2019). Corpora Generation for Grammatical Error Correction. In *Proceedings of NAACL-HLT*, pages 3291–3301.

Liebling, D. J., Heller, K., Mitchell, M., Díaz, M., Lahav, M., Salehi, N., Robertson, S., Bengio, S., Gebru, T., and Deng, W., editors (2021). *Three Directions for the Design of Human-Centered Machine Translation.*

Lin, C.-Y. (2004). ROUGE: A Package for Automatic Evaluation of Summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Logeswaran, L., Lee, H., and Bengio, S. (2018). Content preserving text generation with attribute controls. In Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 31, pages 5103–5113. Curran Associates, Inc.

Luo, R. and Shakhnarovich, G. (2020). Analysis of diversity-accuracy tradeoff in image captioning.

Lutz, J. A. (1987). A Study of Professional and Experienced Writers Revising and Editing at the Computer and with Pen and Paper. *Research in the Teaching of English*, 21(4):398–421.

Mita, M., Sakaguchi, K., Hagiwara, M., Mizumoto, T., Suzuki, J., and Inui, K. (2022). Towards Automated Document Revision: Grammatical Error Correction, Fluency Edits, and Beyond.

Miyabe, M. and Yoshino, T. (2009). Accuracy Evaluation of Sentences Translated to Intermediate Language in Back Translation. In *Proceedings of the 3rd International Universal Communication Symposium*, IUCS '09, page 30–35, New York, NY, USA. Association for Computing Machinery.

Mizumoto, T., Komachi, M., Nagata, M., and Matsumoto, Y. (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*, pages 147–155, Chiang Mai, Thailand. Asian Federation of Natural Language Processing.

Moosavi, N. S., Rücklé, A., Roth, D., and Gurevych, I. (2021). SciGen: a Dataset for Reasoning-Aware Text Generation from Scientific Tables. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*.

Nagata, R., Hagiwara, M., Hanawa, K., Mita, M., Chernodub, A., and Nahorna, O. (2021). Shared Task on Feedback Comment Generation for Language Learners. In *Proceedings of the 14th International Conference on Natural Language Generation*, pages 320–324, Aberdeen, Scotland, UK. Association for Computational Linguistics.

Napoles, C., Sakaguchi, K., and Tetreault, J. (2016). There's No Comparison: Reference-less Evaluation Metrics in Grammatical Error Correction. In *Proceedings of EMNLP*, pages 2109–2115.

Napoles, C., Sakaguchi, K., and Tetreault, J. (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*, pages 229–234, Valencia, Spain. Association for Computational Linguistics.

Ng, H. T., Wu, S. M., Briscoe, T., Hadiwinoto, C., Susanto, R. H., and Bryant, C. (2014). The CoNLL-2014 Shared Task on Grammatical Error Correction. In *Proceedings of CoNLL:Shared Task*, pages 1–14.

Nisioi, S., Štajner, S., Ponzetto, S. P., and Dinu, L. P. (2017). Exploring Neural Text Simplification Models. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 85–91, Vancouver, Canada. Association for Computational Linguistics.

Ott, M., Edunov, S., Baevski, A., Fan, A., Gross, S., Ng, N., Grangier, D., and Auli, M. (2019). fairseq: A Fast, Extensible Toolkit for Sequence Modeling. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: System Demonstrations (NAACL 2019)*, pages 48–53.

Papineni, K., Roukos, S., Ward, T., and Zhu, W.-J. (2002). Bleu: a Method for Automatic Evaluation of Machine Translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Pilzer, J., Rosenast, R., Meyer, A. N., Huang, E. M., and Fritz, T. (2020). Supporting Software Developers' Focused Work on Window-Based Desktops. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, page 1–13, New York, NY, USA. Association for Computing Machinery.

Politzer-Ahles, S., Girolamo, T., and Ghali, S. (2020). Preliminary evidence of linguistic bias in academic reviewing. *Journal of English for Academic Purposes*, 47:100895.

Prabhumoye, S., Tsvetkov, Y., Salakhutdinov, R., and Black, A. W. (2018). Style Transfer Through Back-Translation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 866–876, Melbourne, Australia. Association for Computational Linguistics.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., and Sutskever, I. (2019). Language Models are Unsupervised Multitask Learners.

Ramírez-Castañeda, V. (2020). Disadvantages in preparing and publishing scientific papers caused by the dominance of the English language in science: The case of Colombian researchers in biological sciences. *PloS one*, 15(9):e0238372.

Rei, R., Farinha, A. C., Zerva, C., van Stigt, D., Stewart, C., Ramos, P., Glushkova, T., Martins, A. F. T., and Lavie, A. (2021). Are References Really Needed? Unbabel-IST 2021 Submission for the Metrics Shared Task. In *Proceedings of the Sixth Conference on Machine Translation*, pages 1030–1040, Online. Association for Computational Linguistics.

Sai, A. B., Mohankumar, A. K., and Khapra, M. M. (2022). A survey of evaluation metrics used for nlg systems. *ACM Comput. Surv.*, 55(2).

Sakaguchi, K., Napoles, C., Post, M., and Tetreault, J. (2016). Reassessing the goals of grammatical error correction: Fluency instead of grammaticality. *Transactions of the Association for Computational Linguistics*, 4:169–182.

Sarrafzadeh, B., Jauhar, S. K., Gamon, M., Lank, E., and White, R. W. (2021). Characterizing Stage-Aware Writing Assistance for Collaborative Document Authoring. *Proc. ACM Hum.-Comput. Interact.*, 4(CSCW3).

Schick, T., Yu, J. A., Jiang, Z., Petroni, F., Lewis, P., Izacard, G., You, Q., Nalmpantis, C., Grave, E., and Riedel, S. (2023). PEER: A Collaborative Language Model. In *The Eleventh International Conference on Learning Representations*.

Sellam, T., Das, D., and Parikh, A. (2020). BLEURT: Learning robust metrics for text generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL 2020)*, pages 7881–7892.

Seow, A. (2002). The writing process and process writing. *Methodology in language teaching: An anthology of current practice*, pages 315–320.

Sharma, S., El Asri, L., Schulz, H., and Zumer, J. (2017). Relevance of Unsupervised Metrics in Task-Oriented Dialogue for Evaluating Natural Language Generation. *arXiv preprint arXiv:1706.09799*.

Soyer, H., Topić, G., Stenetorp, P., and Aizawa, A. (2015). CroVeWA: Crosslingual Vector-Based Writing Assistance. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations*, pages 91–95, Denver, Colorado. Association for Computational Linguistics.

Stiennon, N., Ouyang, L., Wu, J., Ziegler, D., Lowe, R., Voss, C., Radford, A., Amodei, D., and Christiano, P. F. (2020). Learning to summarize with human feedback. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H., editors, *Advances in Neural Information Processing Systems*, volume 33, pages 3008–3021. Curran Associates, Inc.

Sun, S., Zhao, W., Manjunatha, V., Jain, R., Morariu, V., Dernoncourt, F., Srinivasan, B. V., and Iyyer, M. (2021). IGA: An Intent-Guided Authoring Assistant. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5972–5985, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Susser, B. (1994). Process approaches in ESL/EFL writing instruction. *Journal of Second Language Writing*, 3(1):31–47.

Tan, C. and Lee, L. (2014). A Corpus of Sentence-level Revisions in Academic Writing: A Step towards Understanding Statement Strength in Communication. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 403–408, Baltimore, Maryland. Association for Computational Linguistics.

Tevet, G. and Berant, J. (2021). Evaluating the Evaluation of Diversity in Natural Language Generation. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 326–346, Online. Association for Computational Linguistics.

Tsai, S.-C. (2020). Chinese students' perceptions of using Google Translate as a translingual CALL tool in EFL writing. *Computer Assisted Language Learning*, 0(0):1–23.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L. u., and Polosukhin, I. (2017). Attention is All you Need. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 30, page 6000−6010. Curran Associates, Inc.

Vijayakumar, A., Cogswell, M., Selvaraju, R., Sun, Q., Lee, S., Crandall, D., and Batra, D. (2018). Diverse Beam Search for Improved Description of Complex Scenes. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, pages 7371–7379.

Wallwork, A. (2016). *English for writing research papers*. Springer.

Wallwork, A. and Southern, A. (2020). *100 Tips to Avoid Mistakes in Academic Writing and Presenting*. Springer.

Wang, Q., Huang, L., Jiang, Z., Knight, K., Ji, H., Bansal, M., and Luan, Y. (2019). PaperRobot: Incremental Draft Generation of Scientific Ideas. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1980–1991, Florence, Italy. Association for Computational Linguistics.

Wang, Q., Xiong, Y., Zhang, Y., Zhang, J., and Zhu, Y. (2021). AutoCite: Multi-Modal Representation Fusion for Contextual Citation Generation. WSDM '21, page 788−796, New York, NY, USA. Association for Computing Machinery.

Wang, Q., Zeng, Q., Huang, L., Knight, K., Ji, H., and Rajani, N. F. (2020). ReviewRobot: Explainable Paper Review Generation based on Knowledge Synthesis. In *Proceedings of the 13th International Conference on Natural Language Generation*, pages 384–397, Dublin, Ireland. Association for Computational Linguistics.

Wieting, J. and Gimpel, K. (2018). ParaNMT-50M: Pushing the Limits of Paraphrastic Sentence Embeddings with Millions of Machine Translations. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 451–462, Melbourne, Australia. Association for Computational Linguistics.

Williams, A., Nangia, N., and Bowman, S. (2018). A Broad-Coverage Challenge Corpus for Sentence Understanding through Inference. In *Proceedings of NAACL-HLT*, pages 1112–1122.

Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., and Brew, J. (2019). HuggingFace's Transformers: State-of-the-art Natural Language Processing. *arXiv preprint arXiv:1910.03771*.

Wu, J.-C., Chang, Y.-C., Mitamura, T., and Chang, J. S. (2010). Automatic Collocation Suggestion in Academic Writing. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics Conference Short Papers (ACL 2010)*, pages 115–119.

Xie, Z., Genthial, G., Xie, S., Ng, A., and Jurafsky, D. (2018). Noising and Denoising Natural Language: Diverse Backtranslation for Grammar Correction. In *Proceedings of NAACL-HLT*, pages 619–628.

Yang, Q., Cranshaw, J., Amershi, S., Iqbal, S. T., and Teevan, J. (2019). Sketching NLP: A Case Study of Exploring the Right Things To Design with Language Intelligence. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, CHI '19, page 1–12, New York, NY, USA. Association for Computing Machinery.

Yang, Z., Hu, Z., Dyer, C., Xing, E. P., and Berg-Kirkpatrick, T. (2018). Unsupervised Text Style Transfer using Language Models as Discriminators. In Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 31, page 7298–7309. Curran Associates, Inc.

Yimam, S. M., Venkatesh, G., Lee, J., and Biemann, C. (2020). Automatic compilation of resources for academic writing and evaluating with informal word identification and paraphrasing system. In *Proceedings of the 12th Language Resources and Evaluation Conference (LREC 2020)*, pages 5896–5904.

Yuan, W., Liu, P., and Neubig, G. (2022). Can We Automate Scientific Reviewing? *J. Artif. Int. Res.*, 75.

Yuan, Z. and Briscoe, T. (2016). Grammatical error correction using neural machine translation. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 380–386, San Diego, California. Association for Computational Linguistics.

Zhang, F., Hashemi, H. B., Hwa, R., and Litman, D. (2017). A Corpus of Annotated Revisions for Studying Argumentative Writing. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1568–1578, Vancouver, Canada. Association for Computational Linguistics.

Zhang, T., Kishore, V., Wu, F., Weinberger, K. Q., and Artzi, Y. (2020). BERTScore: Evaluating Text Generation with BERT. In *International Conference on Learning Representations*.

Zhao, W., Wang, L., Shen, K., Jia, R., and Liu, J. (2019). Improving Grammatical Error Correction via Pre-Training a Copy-Augmented Architecture with Unlabeled Data. 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)*, pages 156–165, Minneapolis, Minnesota. Association for Computational Linguistics.

Zhou, J. and Bhat, S. (2021). Paraphrase Generation: A Survey of the State of the Art. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5075–5086, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Zhu, W. and Bhat, S. (2020). GRUEN for Evaluating Linguistic Quality of Generated Text. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 94–108, Online. Association for Computational Linguistics.

Zhu, W., Hu, Z., and Xing, E. (2019). Text Infilling. *arXiv preprint arXiv:1901.00158*.

Ziegler, D. M., Stiennon, N., Wu, J., Brown, T. B., Radford, A., Amodei, D., Christiano, P., and Irving, G. (2019). Fine-Tuning Language Models from Human Preferences. *arXiv preprint arXiv:1909.08593*.

Zouhar, V., Novák, M., Žilinec, M., Bojar, O., Obregón, M., Hill, R. L., Blain, F., Fomicheva, M., Specia, L., and Yankovskaya, L. (2021). Backtranslation Feedback Improves User Confidence in MT, Not Quality. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 151–161, Online. Association for Computational Linguistics.

Zweig, G., Platt, J. C., Meek, C., Burges, C. J., Yessenalina, A., and Liu, Q. (2012). Computational Approaches to Sentence Completion. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 601–610, Jeju Island, Korea. Association for Computational Linguistics.

List of Publications

Journal Papers (Refereed)

1. Takumi Ito, Tatsuki Kuribayashi, Hayato Kobayashi, Ana Brassard, Masato Hagiwara, Jun Suzuki, Kentaro Inui. Assisting Authors to Convert Raw Products into Polished Prose. Journal of Cognitive Science, Vol.21, No.1, pp.99-135, 2020.

International Conference/Workshop Papers (Refereed)

1. Atsushi Shirafuji, Takumi Ito, Makoto Morishita, Yuki Nakamura, Yusuke Oda, Jun Suzuki, Yutaka Watanobe. Prompt Sensitivity of Language Model for Solving Programming Problems. In Proceedings of The 21st International Conference on Intelligent Software Methodologies, Tools, and Techniques (SOMET), pp. –, September 2022.

2. Tatsuki Kuribayashi, Yohei Oseki, Takumi Ito, Ryo Yoshida, Masayuki Asahara, Kentaro Inui. Lower Perplexity is Not Always Human-Like. In Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (ACL), pp. 5203–5217, August 2021.

3. Takumi Ito, Tatsuki Kuribayashi, Masatoshi Hidaka, Jun Suzuki, Kentaro Inui. Langsmith: An Interactive Academic Text Revision System. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP): System Demonstrations, pp. 216–226, October 2020.

4. Tatsuki Kuribayashi, Takumi Ito, Jun Suzuki, Kentaro Inui. Language Models as an Alternative Evaluator of Word Order Hypotheses: A Case Study in Japanese. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL), pp. 488–504, July 2020.

5. Shun Kiyono, Takumi Ito, Ryuto Konno, Makoto Morishita, Jun Suzuki. Tohoku-AIP-NTT at WMT 2020 News Translation Task. In Proceedings of the Fifth Conference on Machine Translation (WMT), pp. 145–155, November 2020.

6. Takumi Ito, Tatsuki Kuribayashi, Hayato Kobayashi, Ana Brassard, Masato Hagiwara, Jun Suzuki and Kentaro Inui. Diamonds in the Rough: Generating Fluent Sentences from Early-stage Drafts for Academic Writing Assistance. In Proceedings of the 12th International Conference on Natural Language Generation (INLG), pp. 40–53, October–November 2019.

7. Masato Hagiwara, Takumi Ito, Tatsuki Kuribayashi, Jun Suzuki and Kentaro Inui. TEASPN: Framework and Protocol for Integrated Writing Assistance Environments. In Proceedings of 2019 Conference on Empirical Methods in Natural Language Processing and 9th International Joint Conference on Natural Language Processing (EMNLP): System Demonstrations, pp. 229–234, November 2019.

Awards

Asia-Pacific Association for Machine Translation (AAMT) 第 16 回長尾賞

Other Publications (Not refereed)

1. 栗林樹生, 大関洋平, 伊藤拓海, 吉田遼，浅原正幸, 乾健太郎. 予測の正確な言語モデルがヒトらしいとは限らない. 言語処理学会第 27 回年次大会, pp. 267–272, March 2021. (言語処理学会第 27 回年次大会委員特別賞)

2. 栗林樹生, 大関洋平, 伊藤拓海, 吉田遼，浅原正幸, 乾健太郎. 日本語の読みやすさに対する情報量に基づいた統一的な解釈. 言語処理学会第 27 回年次大会, pp. 723-728, 2021.

3. 伊藤拓海, 栗林樹生, 日高雅俊, 鈴木潤, 乾健太郎. Langsmith: 人とシステムの協働による論文執筆. 言語処理学会第 27 回年次大会, pp. 1834–1839, March 2021.

4. 栗林樹生, 伊藤拓海, 鈴木潤, 乾健太郎. 日本語語順分析に言語モデルを用いることの妥当性について. 言語処理学会第 26 回年次大会, pp. 493–496, March 2020.

5. 伊藤拓海, 栗林樹生, 萩原正人, 鈴木潤, 乾健太郎. 英語論文執筆のための統合ライティング支援環境. 第 14 回 NLP 若手の会シンポジウム, August 2019.

6. 栗林樹生, 伊藤拓海, 内山香, 鈴木潤, 乾健太郎. 言語モデルを用いた日本語の語順評価と基本語順の分析. 言語処理学会第 25 回年次大会, pp.1053–1056, March 2019.

7. 伊藤拓海, 栗林樹生, 小林隼人, 鈴木潤, 乾健太郎. ライティング支援を想定した情報補完型生成. 言語処理学会第 25 回年次大会, pp.970-973, March 2019.

8. 伊藤拓海, 山口健史, 田然, 松田耕史, 岡崎直観, 乾健太郎. 自治体 FAQ の比較マイニング. 言語処理学会第 24 回年次大会, pp.536-539, March 2018.

9. 伊藤拓海, 鈴木正敏, 田然, 山口健史, 岡崎直観, 乾健太郎. 自治体 QA サービスのための FAQ の自治体間の横断的解析. 第 12 回 NLP 若手の会シンポジウム, September 2017.

Appendix

.1 Examples from the Smith dataset and generated sentences by baseline models

Table 1 shows examples from the SMITH dataset and the output of the baseline models. “Reference” is a sentence extracted from papers, “Draft” is written by a crowdworker and is the input for the baseline models.

Table 1 Further examples of draft, reference, and the baseline models’ output.

Draft	By this setting , the persona is acquired from a test set popl about both turker anad model .
H-ND	By this setting , the persona is acquired from a test set both about popl anad anad model .
ED-ND	In this setting , persona is obtained from the test set popl about both Turker and model .
GEC	By this setting , the persona is acquired from a test set pool about both turkey and models .
Reference	In this setting , for both the Turker and the model , the personas come from the test set pool .
Draft	In addition to results of study until now , we add two baseline to vindicate effectiveness on our flame work .
H-ND	In addition to the results of this study , we now add two baseline methods to vindicate effectiveness on our work .
ED-ND	In addition to the results of the study until now , we add two baselines to visualize the effectiveness of our framework .

GEC	In addition to the results of study until now , we added two baseline to vindicate effectiveness on our flame work .
Reference	In addition to results of previous work , we add two baselines to demonstrate the effectiveness of our framework .
Draft	Yhe input and output <∗> are one - hot encoding of the center word and the context word , <∗> .
H-ND	The input and output are one - hot encoding of the center word and the context word , respectively .
ED-ND	The input and output layers are one - hot encoding of the center word and the context word , respectively .
GEC	Yhe input and output are one - hot encoding of the center word and the context word , .
Reference	The input and output layers are centre word and context word one - hot encodings , respectively .
Draft	I registered the vocabulary sizes of encorder and decorder as 150 K and 50 K each other .
H-ND	I registered the vocabulary sizes of decorder and encorder as 150 K and each other .
ED-ND	We registered the vocabulary sizes of the encoder and decoder as 150 K and 50 K respectively .
GEC	I registered the vocabulary sizes of encoder and recorder as 150 K and 50 K for each other .
Reference	In this experiment , we set the vocabulary size on the encoder and decoder sides to 150 K and 50 K , respectively .
Draft	They add the new class image generated by generator and classfy them .
H-ND	They add the new image class generated by the generator and classfy them .
ED-ND	They add a new class of images generated by the generator and classify them .
GEC	They add a new class image generated by generator and classify them .

Reference	They add a new class of images that are generated by the generator and classify them .
Draft	The chart 3 shows performance of multi input correction against sub groups with different number of witnesses .
H-ND	Table 3 shows the performance of multi - chart correction against different input groups with different number of witnesses .
ED-ND	Figure 3 shows the performance of multiple input correction against subgraphs with different number of witnesses .
GEC	chart 3 shows performance of multi input correction against sub groups with different number of witnesses .
Reference	Figure 3 presents the performance of multi - input correction on subgroups with different number of witnesses .
Draft	It is vindicated that InferSent accomplishes the most <*> result regarding SentEval task .
H-ND	It is vindicated that InferSent accomplishes the most relevant result regarding the SentEval task .
ED-ND	It is vindicated that InferSent accomplishes the most important result regarding the SentEval task .
GEC	It is vindicated that InferSent accomplishes the most results regarding SentEval task .
Reference	InferSent has been shown to achieve state - of - the - art results on the SentEval tasks .
Draft	Our proposal model can get both long - term dependence and local information well .
H-ND	Our proposal can get both long - term and local information as well .
ED-ND	Our proposed model can capture both long - term dependencies and local information well .
GEC	Our proposal model can get both long - term dependence and local information well .
Reference	Our proposed model can both capture long - term dependencies and local information well .

Table 2 Comparison of writing performance between *MT* and *SELF*. TA=Task achievement; CC=Coherence and cohesion; LR=Lexical resource; GRA=Grammatical range and accuracy. Total is the sum of those criteria. Each value is the mean score of the participant's writings, and the value in parentheses is the standard deviation. The values in parentheses for groups are headcounts.

Group	TA	CC	LR	GRA	Total
MT (15)	2.5 (1.1)	6.2 (1.1)	6.7 (1.3)	7.5 (0.9)	22.9 (3.7)
SELF (6)	2.7 (0.5)	6.2 (1.7)	7.0 (1.5)	7.5 (0.6)	23.3 (4.0)

.2 Writing Performance of participants in the main study of Chpater 5

We asked iPassIELTS to rate the participants' writing based on the indicators usually employed when providing course feedback.[1] Each participant's tasks was evaluated in four aspects: "Task achievement," "Coherence and cohesion," "Lexical resource," and "Grammatical range and accuracy." Each aspect consists of two scoring items, and all items were rated on the same four-point scale, i.e., 1 = satisfactory, 2 = good, 3 = very good, and 4 = excellent.

The average number of words produced in the two tasks was almost identical: 161 words (SD = 13.6, range: 131-186) for the bar-graph task and 165 words (SD = 17.1, range: 125-209) for the table task. Two participants, P10-MT and P12-MT, took more than 30 minutes on both of their tasks, yet failed to reach the 150 word goal on either. The participant's score for each of the four dimensions of writing performance was the sum of the relevant iPassIELTS-assigned item scores across both writing tasks. As indicated by the scores presented in Table 2, there were minor differences between the *MT* and *SELF* groups in any writing-quality dimension. Note that the texts were written while using tools such as Langsmith and machine translation, and do not represent the pure English proficiency of the participants.

[1] Sample feedback: https://www.ipassielts.com/images/uploads/Nuri_GDP_growth_web.pdf