

**Controlled Natural Language Generation
for Morphologically Rich Languages: The Case of Arabic**

DISSERTATION

Submitted in Partial Fulfillment of

the Requirements for

the Degree of

DOCTOR OF PHILOSOPHY (Computer Science)

at the

NEW YORK UNIVERSITY
TANDON SCHOOL OF ENGINEERING

by

Bashar Alhafni

May 2025

**Controlled Natural Language Generation
for Morphologically Rich Languages: The Case of Arabic**

DISSERTATION

Submitted in Partial Fulfillment of
the Requirements for
the Degree of

DOCTOR OF PHILOSOPHY (Computer Science)

at the

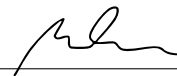
NEW YORK UNIVERSITY
TANDON SCHOOL OF ENGINEERING

by

Bashar Alhafni

May 2025

Approved:



Department Chair Signature

May 3, 2025

Date

Approved by the Guidance Committee:

Major: Computer Science



Ted Briscoe

Professor
MBZUAI

April 30, 2025

Date



Nizar Habash

Professor
NYU Abu Dhabi

April 30, 2025

Date

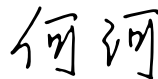


Kyunghyun Cho

Professor
NYU Courant Institute

April 29, 2025

Date



He He

Assistant Professor
NYU Courant Institute

May 1, 2025

Date



Mona Diab

Professor
CMU Language Technologies Institute

April 29, 2025

Date



Julia Stoyanovich

Associate Professor
NYU Tandon School of Engineering

April 29, 2025

Date

Microfilm or other copies of this dissertation are obtainable from

UMI Dissertation Publishing

ProQuest CSA

789 E. Eisenhower Parkway

P.O. Box 1346

Ann Arbor, MI 48106-1346

Vita

Bashar Alhafni was born in Damascus, Syria, in 1994. He earned a B.Sc. in Computer Science with a minor in Mathematics from the University of Bridgeport in 2017, followed by an M.Sc. from the University of Southern California in 2019. Since September 2020, he has been pursuing a Ph.D. in Computer Science at the New York University Tandon School of Engineering under the supervision of Prof. Nizar Habash. His doctoral studies are funded by the Global Ph.D. Student Fellowship at New York University Abu Dhabi. In summer 2022, he completed a research internship at Dataminr in New York City, focusing on timeline event extraction and summarization. The following summer, in 2023, he interned at Grammarly in San Francisco, working on personalized text generation and multilingual text editing. His research primarily focuses on natural language processing, with an emphasis on Arabic NLP and controlled natural language generation.

Acknowledgements

I am incredibly grateful for all the support I have received during my PhD. I know these acknowledgements are far too short and not enough. I would like to express my sincere gratitude to my advisor, Nizar Habash, for his trust and guidance. Nizar has been more than an advisor; he is a mentor, colleague, and friend. He taught me so much about research and life, and cared about my well-being. I also thank my PhD committee members: Profs. Ted Briscoe, Kyunghyun Cho, Mona Diab, He He, and Julia Stoyanovich for their valuable comments and feedback.

I am grateful to my colleagues at NYU Abu Dhabi, especially in the CAMEL Lab. I am proud to have been a CAMELeer and thank my labmates, past and present, for their friendship and collaboration. I am especially thankful to my coauthors Go Inoue, Injy Hamed, Christian Khairallah, Salam Khalifa, Ossama Obeid, Dima Taji, and Nasser Zalmout. I also acknowledge the NYU Abu Dhabi Global PhD Fellowship for generously supporting me during the first four years of my PhD, and Ted Briscoe and MBZUAI for supporting me during the final year. I am also thankful to my collaborators at Grammarly and Dataminr: Joel Tetrault, Vipul Raheja, Vivek Kulkarni, and Dhruv Kumar.

This dissertation would not have been possible without the unwavering support of many people in my life, especially my parents, whose sacrifices and encouragement enabled me to pursue a world-class education, despite not having the opportunity to attend college themselves. I also extend my deepest gratitude to my friends who shared this journey with me, especially Michele, Sara, Shaza, Salmane, Semih, Teodora, Reem, Farah, Valentin, Valentina, Yana, and Yazan. Abu Dhabi has been great because of you.

Bashar Alhafni

May 2025

الْعِلْمُ يَرْفَعُ بَيْتاً لَا عِمَادَ لَهُ وَالْجَهْلُ يَهْدِمُ بَيْتَ الْعِزِّ وَالْكَرَمِ

Knowledge raises a house without foundation

While ignorance destroys a house of glory and generosity.

To my parents, this is for you and because of you.

Thank you for believing in me.

ABSTRACT

**Controlled Natural Language Generation
for Morphologically Rich Languages: The Case of Arabic****by****Bashar Alhafni****Advisor: Prof. Nizar Habash****Submitted in Partial Fulfillment of the Requirements for
the Degree of Doctor of Philosophy (Computer Science)****May 2025**

Recent breakthroughs in natural language processing (NLP) have led to the development of natural language generation (NLG) systems, such as large language models (LLMs), that can produce fluent, human-like text. However, these models are predominantly pretrained on data that is mostly in English, limiting their performance on languages with less representation. This challenge is further amplified by the fact that languages differ in their morphological complexity. For example, languages like Arabic are morphologically rich, featuring a high number of inflections for a single base word. In contrast, languages like English are morphologically poorer, with fewer inflections and simpler word forms. Additionally, current NLG models are challenging to control and often over-generate, making them unreliable for tasks that require high precision or for generating outputs tailored to specific user preferences.

Arabic presents additional challenges for NLG. Unlike English, it is orthographically ambiguous, as it uses optional diacritics to indicate short vowels and consonantal doubling. Since these diacritics are typically omitted, readers must rely on contextual and templatic morphology clues to deduce meaning, making disambiguation a core challenge for Arabic NLP. Additionally, Arabic is diglossic, with Modern Standard Arabic (MSA) coexisting alongside Dialectal Arabic (DA). MSA serves as the standard form of the language and it is used in news, education, and official communication, yet it is not the native language of any Arabic speaker and often suffers from orthographic inconsistencies. In contrast, DA, which is primarily spoken, lacks standardized orthography. The rise of social media has led to a surge in written DA content, but without established spelling conventions, this data remains highly inconsistent and noisy. Furthermore, annotated resources for DA are scarce across dialects, further complicating the development of Arabic NLP.

This dissertation addresses these challenges by introducing controlled NLG approaches tailored to Arabic. We develop NLG models for three key Arabic NLG tasks that contribute to AI in social good and education: gender rewriting, grammatical error correction, and dialectal text normalization. Beyond introducing state-of-the-art models, our work contributes new datasets that enable new Arabic NLG tasks, expanding the resources available for Arabic NLP. We propose controllable modeling strategies that incorporate linguistic traits into NLG systems. While many linguistic traits shape how we communicate and use language, this dissertation focuses on three that directly impact writing in Arabic: grammatical gender, error patterns, and dialect. Although Arabic is our primary focus, the insights and techniques developed here extend to other morphologically rich languages, promoting the advancement of more inclusive and controllable NLG systems.

Table of Contents

Vita	iv
Acknowledgements	v
Abstract	vii
List of Figures	xv
List of Tables	xxiii
1 Introduction	1
1.1 Overview and Motivation	1
1.2 Dissertation Outline	5
1.3 Contributions	6
1.4 Publications	8
2 Arabic Linguistic Background	10
2.1 Arabic and its Dialects	10
2.2 Arabic Morphology	13
2.3 Arabic Orthography	14
3 Natural Language Generation Background	17
3.1 Natural Language Generation	17
3.1.1 A Short History of Natural Language Generation	18
3.1.2 Types of Natural Language Generation	21

	x
3.1.3 Neural Architectures for Natural Language Generation	22
3.2 Controlled Natural Language Generation	23
3.3 Language Generation Tasks in this Dissertation	25
4 Arabic Gender Rewriting	27
4.1 Introduction	28
4.2 Background and Related Work	30
4.3 The Arabic Parallel Gender Corpus	33
4.3.1 Corpus Selection	34
4.3.2 Corpus Annotation	35
4.3.3 Automatic Word-Level Annotations	37
4.3.4 Corpus Overview and Statistics	38
4.4 Approach	42
4.4.1 Joint Gender Rewriting	42
4.4.2 Multi-Step Gender Rewriting	43
4.5 Experimental Setup	46
4.5.1 Results	48
4.5.2 Error Analysis	50
4.5.3 Use Case: Post-Editing MT Output	50
4.6 The User-Aware Arabic Gender Rewriter	52
4.7 The Shared Task on Arabic Gender Rewriting	54
4.7.1 Data	54
4.7.2 Participants and Systems	54
4.7.3 Results	55
4.7.4 Error Analysis	56
4.8 Summary	57

	xi
5 Arabic Grammatical Error Detection and Correction	58
5.1 Introduction	59
5.2 Background and Related Work	60
5.3 Approach	63
5.3.1 Arabic Grammatical Error Detection	63
5.3.2 Arabic Grammatical Error Correction	68
5.4 Experimental Setup	70
5.4.1 Data	70
5.4.2 Experiments	71
5.4.3 Results	74
5.4.4 Error Analysis	79
5.5 Summary	80
6 Dialectal Text Normalization	81
6.1 Introduction	82
6.2 Background and Related Work	83
6.3 Approach	87
6.4 Experimental Setup	89
6.4.1 Data	89
6.4.2 Experiments	90
6.4.3 Results	92
6.4.4 Error Analysis	95
6.5 Summary	97
7 Text Editing	98
7.1 Introduction	99

	xii
7.2 Background and Related Work	100
7.3 Approach	102
7.3.1 Edit Extraction	102
7.3.2 Edit Representation	104
7.3.3 Edits Coverage	106
7.4 Experimental Setup	108
7.4.1 Data	108
7.4.2 Experiments	109
7.4.3 Results	110
7.4.4 Runtime Performance	114
7.4.5 Error Analysis	115
7.5 Summary	119
8 Summary and Conclusions	120
8.1 Arabic Gender Rewriting	120
8.2 Arabic Grammatical Error Detection and Correction	122
8.3 Dialectal Text Normalization	123
8.4 Text Editing	124
8.5 Future Work	125
A Chapter 4 Appendix	129
A.1 LLMs Prompts	129
A.2 LLMs Results	132
B Chapter 5 Appendix	133
B.1 LLMs Prompts	133
B.2 LLMs Results	136

	xiii
B.3 Error Type Statistics	137
C Chapter 6 Appendix	138
C.1 LLMs Prompts	138
C.2 LLMs Results	141
D Chapter 7 Appendix	142
D.1 Edit Tagging Results	142

List of Figures

2.1	The distribution of the different Arabic dialects over the Arab World and surrounding areas (Wikipedia, 2011).	11
4.1	The multi-step gender rewriting system. First person gendered words are in purple and second person gendered words are in red . The user target gender is 1M/2M. The input words <i>glad</i> (1F+B), <i>know you</i> (B+2F), and <i>ladies</i> (2F+B) are rewritten to their masculine forms.	43
4.2	The Arabic Gender Rewriter interface showing gender rewritten alternatives of three input sentences in four modes: (a) Target speaker ♀ gender rewrites, (b) Target speaker ♀ and target listener ♀ and ♂ gender rewrites, (c) Target speaker ♀ and ♂ and target listener ♀ gender rewrites, and (d) Target speaker ♀ and ♂ and target listener ♀ and ♂ gender rewrites. Speaker gendered words are in blue and listener gendered words are in orange	53

- 5.1 An example showing the differences between the alignments of the M^2 scorer, a standard Levenshtein distance, ARETA, and our proposed algorithm. The edit operations are keep (**K**), replace (**R**), insert (**I**), delete (**D**), merge (**M**), and split (**S**). Dotted lines between the erroneous and corrected sentences represent gold alignment. The last three rows present different granularities of ARETA error types based on our alignment. The sentence in the figure can be translated as “*Social media must be used wisely, as it has both negative and positive effects*”. 67
- 7.1 An example showing the different edit representations: words, words (compressed), subwords, and subwords (compressed). The edit operations are keep (**K/K***), delete (**D/D***), merge before (**M**), replace (**R_[c]**), insert (**I_[c]**), and append (**A_[c]**). Solid lines indicate word alignments between the corrected and erroneous sentences, while dotted lines denote erroneous subword boundaries. The sentence in the figure can be translated as “*Health, especially mental health, must be taken care of*”. 103

List of Tables

4.1	Examples of the changes needed to generate gender alternative forms of gender-specific words in Arabic.	33
4.2	Examples from the APGC v2.0 including the original sentence, its gender label, its rewrite gender label, and its rewrite to the opposite grammatical gender where appropriate. First person gendered words are in purple and second person gendered words are in pink . The two-letter label specifies gender information of first person (first letter) and second person (second letter). M is Masculine; F is Feminine; and B is invariant.	36
4.3	Examples of word-level gender annotation. First person gendered words are in purple and second person gendered words are in pink	38
4.4	Sentence-level statistics of the original corpus (a) and the balanced corpus (b) with its five versions.	39
4.5	Word-level statistics of the original corpus (a) and the balanced corpus (b) with its five versions.	41
4.6	Multi-user gender rewriting results on the Dev set of APGC v2.0. Aug indicates using augmented data.	49
4.7	Gender rewriting results on the Test sets of APGC v2.0.	50
4.8	Error type statistics of our best augmented system's performance on APGC v2.0 Dev.	51

4.9	BLEU results on the post-edited Google Translate output of APGC v2.1	
	Test using our best augmented system.	51
4.10	List of the five teams who participated in the gender rewriting shared task.	55
4.11	Approaches and techniques used by the participants. Gender ID refers to gender identification. Special Preprocessing refers to any form of preprocessing done to modify the data (e.g., adding side-constraints, morphological processing, transliteration, etc.). Pretrained Models indicates the usage of pretrained models as part of the system.	55
4.12	Results on the Blind Test set. Numbers in parentheses are the ranks. . .	56
4.13	(a) The relative difference in the number of generated words for each team in comparison with the Blind Test reference. (b) The Pearson correlation of the shared task metrics in Table 4.12 with the <i>absolute</i> values of Word Δ	56
5.1	Evaluation of different alignment algorithms.	65
5.2	The statistics of the error types in the Train sets of QALB-2014, QALB-2015, and ZAEBUC. The error types are based on the extended ALC (Alfaifi et al., 2013) taxonomy as used by Belkebir and Habash (2021).	66
5.3	Corpus statistics of Arabic GEC datasets.	71
5.4	GED results on the Dev and Test sets in terms of macro precision, recall, $F_{0.5}$, and accuracy.	74
5.5	GEC results on the Dev sets of QALB-2014, QALB-2015, and ZAEBUC. B&B (2015) and W+ (2018) refer to Bougares and Bouamor (2015) and Watson et al. (2018a), respectively. The best overall results are in bold. Results of our best systems are underlined.	75

5.6	GED granularity results when used within the best GEC system on the Dev sets of QALB-2014, QALB-2015, and ZAEBUC. Results in grey indicate using gold GEC labels (i.e., Oracle). The best results are in bold.	76
5.7	GED granularity results when used within GEC on the Test sets of QALB-2014, QALB-2015, and ZAEBUC. B&B (2015) and S+ (2022) refer to Bougares and Bouamor (2015) and Solyman et al. (2022), respectively. The best overall results are in bold. Results of our best systems are underlined.	78
5.8	Specific error type performance of AraBART and our best system on the Dev sets of QALB-2014, QALB-2015, and ZAEBUC. Results are reported in terms of $F_{0.5}$. The best results are in bold.	79
6.1	An example sentence from the MADAR CODA Corpus in its raw and CODA parallel forms across five city dialects. The DA sentences are provided along with their transliterations in the HSB scheme (Habash et al., 2007). The sentence in the table can be translated as “ <i>We would like two hamburgers and two coffees. To go, please.</i> ”	87
6.2	The four different types of control tokens we use in our experiments.	88
6.3	The top 10 character edit transformations from raw to CODA in the entire MADAR CODA dataset across the five dialects. <SPC> indicates an explicit white space; whereas an empty cell indicates a <i>null</i> string.	90
6.4	Dev set results for multiple systems. Results in grey indicate using gold DID labels (i.e., Oracle). Best results are in bold. Best oracle results are underlined.	93
6.5	Results on the Test set.	94

6.6	Dialect-specific results of the best system (AraT5 + City) against the baseline (AraT5) on the Dev set.	94
6.7	Dialect-specific results of the best system (AraT5 + DA Phrase) against the baseline (AraT5) on the Test set.	95
6.8	Distribution of errors in the Dev set with one example per error type. . .	96
7.1	Edit statistics on QALB-2014, QALB-2015, and ZAEBUC. Input is the input unit representation (word or subword). Comp. indicates whether the edit is compressed. Subset specifies whether the edits capture all errors, punctuation-only errors (Pnx), or non-punctuation errors (NoPnx). Edits represents the total number of unique edits in the training set of each dataset. OOV% is the percentage of out-of-vocabulary edits (non-unique) in the Dev set of each dataset.	106
7.2	Edit statistics on MADAR CODA and APGCv2.0. Input is the input unit representation (word or subword). Comp. indicates whether the edit is compressed. Edits represents the total number of unique edits in the training set of each dataset. OOV% is the percentage of out-of-vocabulary edits (non-unique) in the Dev set of each dataset.	107
7.3	MSA GEC results on the Dev sets of QALB-2014, QALB-2015, and ZAEBUC. Best non-ensemble results are underlined. The best overall results are in bold.	110
7.4	CODAfication and gender rewriting results on the Dev sets of MADAR CODA and APGC v2.0. Best non-ensemble results are underlined, best overall results are in bold.	111

7.5	MSA GEC results on the Test sets of QALB-2014, QALB-2015 (L1), QALB-2015 (L2), and ZAEBUC. Best non-ensemble results are underlined. The best overall results are in bold.	113
7.6	CODAfication and gender rewriting results on MADAR CODA and APGC v2.0 Test sets. Best non-ensemble results are underlined, best overall results are in bold.	114
7.7	Number of parameters (Params.), initialization time (Init.), and runtime for different models on the Dev set of QALB-2014. Init. and runtime are in seconds and averaged over 10 runs on a single A100 GPU using a batch size of 32.	115
7.8	Error type performance ($F_{0.5}$) on the Dev sets of QALB-2014, QALB-2015, and ZAEBUC for the best baseline Seq2Seq model, our best SWEET system, and the best ensemble. Best non-ensemble scores are underlined, best overall scores are in bold.	116
7.9	Distribution of errors over a sample of 100 cases from the MADAR CODA Dev set for the best Seq2Seq model, the best SWEET model, and the best ensemble. Percentages in parentheses indicate the reduction in errors relative to the Seq2Seq model.	117
7.10	Distribution of erroneous sentences from the APGC v2.0 Dev set across the four target corpora for the Multi-Step model, the best SWEET model, and the top ensemble. The second half of the table reports the distribution of Fixed , Shared , and newly introduced Errors by the SWEET and ensemble models.	117

7.11	Error type distribution for a sample of 100 Dev sentences from APGC v2.0, comparing the Multi-Step model, the best-performing SWEET model, and the top ensemble system.	118
A.1	0-shot prompts used to evaluate LLMs performance on gender rewriting. Prompt Lang is the prompt language either in English (EN) or Arabic (AR).	129
A.2	5-shot prompts used to evaluate LLMs performance on gender rewriting. Prompt Lang is the prompt language either in English (EN) or Arabic (AR).	130
A.3	5-shot prompts with gender identification used to evaluate LLMs performance on gender rewriting. Prompt Lang is the prompt language either in English (EN) or Arabic (AR).	131
A.4	LLMs results on the Dev sets of APGCv2.0. P-Lang is the prompt language either in English (EN) or Arabic (AR). Best $F_{0.5}$ results for each LLM are underlined.	132
B.1	0-shot prompts used to evaluate LLMs performance on GEC. Prompt Lang is the prompt language either in English (EN) or Arabic (AR). . .	133
B.2	5-shot prompts used to evaluate LLMs performance on GEC. Prompt Lang is the prompt language either in English (EN) or Arabic (AR). . .	134
B.3	5-shot prompts with binary GED used to evaluate LLMs performance on GEC. Prompt Lang is the prompt language either in English (EN) or Arabic (AR).	135
B.4	LLMs results on MSA and DA GEC on the Dev sets of QALB-2014, QALB-2015, and ZAEBUC. P-Lang is the prompt language either in English (EN) or Arabic (AR). Best average $F_{0.5}$ results for each LLM are underlined. Best overall results are in bold.	136

B.5	The statistics of the different GED granularity error types we model across the three datasets. The description of the labels in the 13-Class and 43-Class categories are in Table 5.2. For the 2-Class labels, E refers to erroneous words and C refers to correct words.	137
C.1	0-shot prompts used to evaluate LLMs performance on CODAfication. Prompt Lang is the prompt language either in English (EN) or Arabic (AR).	138
C.2	5-shot prompts used to evaluate LLMs performance on CODAfication. Prompt Lang is the prompt language either in English (EN) or Arabic (AR).	139
C.3	5-shot prompts with specified dialect used to evaluate LLMs performance on CODAfication. Prompt Lang is the prompt language either in English (EN) or Arabic (AR).	140
C.4	LLMs results on the Dev set of MADAR CODA. P-Lang is the prompt language either in English (EN) or Arabic (AR). Best $F_{0.5}$ results for each LLM are underlined.	141
D.1	CODAfication and gender rewriting results on the Dev sets of MADAR CODA and APGC v2.0. Input is the input unit representation (word or subword). Comp. indicates whether the edit is compressed. Pruning experiments were conducted using the top two models (AraBERTv02 and CAMeLBERT). Best results are in bold.	142

D.2 MSA GEC results on the Dev sets of QALB-2014, QALB-2015, and ZAEBUC. **Input** is the input unit representation (word or subword). **Comp.** indicates whether the edit is compressed. **Subset** specifies whether the edits capture all errors, punctuation-only errors (Pnx), or non-punctuation errors (NoPnx). NoPnx models are evaluated after removing punctuation, while Pnx models are evaluated on a version of the Dev set where all non-punctuation errors are corrected. Pruning experiments were conducted using the top two models (AraBERTv02 and CAMELBERT), while punctuation segregation experiments used the best model (AraBERTv02). Best **All** results are in bold. Best **NoPnx** and **Pnx** results are underlined. 143

Chapter 1

Introduction

In this chapter, we present an overview of the dissertation, starting with the main motivations behind this work and the challenges in Arabic natural language generation. Next, we outline the structure of the dissertation, summarizing the content of each chapter. We then highlight the main research contributions before concluding with a list of publications directly related to this work.

1.1 Overview and Motivation

Enabling computers to automatically generate natural text is what initially started the development of the natural language processing (NLP) field as we know it today. Interest in natural language generation (NLG) dates back to the 1950s. This was evident in the development of the first machine translation (MT) system at the beginning of the Cold War ([Hutchins, 2004](#)), as well as in Alan Turing's proposal for a new criterion to measure intelligence, which involved a computer's ability to mimic human written conversation to the point where it cannot be distinguished from a human ([Turing, 1950](#)). By the 1960s, NLG research expanded into areas such as paraphrasing ([Klein, 1965](#)),

discourse generation (Klein and Simmons, 1963), and dialogue generation (Weizenbaum, 1966). However, NLG proved to be a challenging problem, and early attempts relied heavily on rule-based systems, which had limited success in generating coherent and flexible text from structured data (McKeown, 1985; Shieber et al., 1989; Smadja and McKeown, 1994; Beesley, 1996). With the advancements in statistical machine learning, NLG approaches shifted from rule-based to data-driven models (Radev et al., 2002; Koehn et al., 2003; Charniak et al., 2003). The rise of deep learning and increased computational power further transformed NLG, enabling more sophisticated models for text generation (Hochreiter and Schmidhuber, 1997; Cho et al., 2014; Bahdanau et al., 2015; Vaswani et al., 2017). In particular, large-scale self-supervised pretraining (Peters et al., 2018; Devlin et al., 2019; Yang et al., 2019a) significantly improved NLG performance, leading to models like BART (Lewis et al., 2020) and T5 (Raffel et al., 2020). More recently, large language models (LLMs) have demonstrated the ability to frame almost any NLP task as a generation problem, achieving strong zero- and few-shot performance through prompting (Kojima et al., 2023; Wei et al., 2023; Touvron et al., 2023; Almazrouei et al., 2023; OpenAI et al., 2024). Despite these advancements, current NLG approaches still face key limitations that hinder their broader applicability, particularly for morphologically rich languages like Arabic. This dissertation addresses two key challenges: **English-Centricity** and **Limited Controllability**.

English-Centric NLG: Current state-of-the-art NLG models are typically deep encoder-decoder or decoder-only neural networks based on the transformer architecture (Vaswani et al., 2017). These models are pretrained on massive amounts of unannotated text, primarily sourced from the web, to achieve strong performance. For example, LLaMA-3 (Grattafiori et al., 2024) was trained on 15 trillion tokens of web data. However, En-

English dominates web content. According to the latest statistics from Common Crawl—a non-profit organization that regularly captures snapshots of the web—English accounts for 46% of all textual content, while most other languages are limited to around 6% (Rana, 2010). Arabic, despite being spoken by over 400 million people and serving as the official language of more than 20 countries, makes up only about 0.5% of web data. This severe underrepresentation presents a significant challenge for Arabic NLG, as models trained primarily on English-centric data struggle to capture Arabic morphological richness and diglossic nature. While multilingual pretraining efforts (e.g., GPT-3 (Brown et al., 2020), BLOOMZ (Muennighoff et al., 2023)) aim to expand linguistic coverage, underrepresented languages still perform worse than high-resource ones (Joshi et al., 2020; Pfeiffer et al., 2020; Patra et al., 2023; Asai et al., 2024). This highlights the need for targeted research on Arabic NLG to create more inclusive NLP technologies that capture their linguistic characteristic and cultural impact.

Limited Controllability in NLG: In conditional NLG, the goal is to generate an output sequence $Y = y_1, y_2, \dots, y_n$ given an input sequence $X = x_1, x_2, \dots, x_m$. This process is typically modeled through the following autoregressive distribution:

$$P(Y|X) = \prod_{t=1}^n P(y_t | y_1, \dots, y_{t-1}, X)$$

Different NLG tasks require varying degrees of control over the generated text. For example, grammatical error correction demands a high level of precision, as the output should closely resemble the input, with only minimal modifications. In contrast, tasks like MT allow for more flexibility in phrasing while preserving meaning. However, most state-of-the-art NLG models, particularly LLMs, struggle with controlled generation (Sun et al., 2023). They often introduce unnecessary changes by adding, removing, or

paraphrasing content, even when minimal edits are required (Fang et al., 2023; Davis et al., 2024). This lack of fine-grained control not only degrades user experience but also limits the applicability of these models in tasks that require high fidelity to the input. Another critical aspect of controllability in NLG is ensuring that generated outputs align with user-specific requirements. Most standardized NLG tasks assume a single, objectively correct output, but in many real-world applications, the correct output depends on the user’s context, preferences, or identity. The inability of existing models to incorporate such user-specific constraints makes them inflexible and often inappropriate. A case in point is the “*I-am-a-doctor and you-are-a-nurse*” MT problem in many gender-marking languages such as Arabic, where single-output user-unaware MT often results in “*I am a [male] doctor and you are a [female] nurse*”, which is inappropriate for female doctors and male nurses, respectively. A controllable NLG system should allow users to specify preferences, ensuring that the generated output better reflects their identity and expectations (Sun et al., 2019; Blodgett et al., 2020).

These challenges make NLG a compelling area of research. In this dissertation, we address these limitations by developing controlled NLG approaches specifically designed for Arabic. Our work focuses on three key tasks: **gender rewriting**, **grammatical error correction**, and **dialectal text normalization**. While many traits shape how individuals write, we focus on those that have a direct impact on Arabic writing, such as grammatical gender, error types, and dialect. For each task, we propose controllable modeling approaches that incorporate these traits to enhance the controllability of NLG systems. Each chapter introduces the task, discusses its challenges, and presents our contributions. Although our primary focus is Arabic, the models and insights developed in this dissertation have broader applicability to other morphologically rich languages.

1.2 Dissertation Outline

The rest of the dissertation is divided into the following chapters:

Second Chapter: Arabic Linguistic Background In the second chapter, we provide a linguistic overview of the Arabic language and its dialects, focusing on its morphological richness and orthographic ambiguity. We discuss the challenges that Arabic NLG systems encounter when processing diverse and noisy text, highlighting the complexities introduced by grammatical variation and dialectal differences.

Third Chapter: NLG Background The third chapter provides a background on NLG, covering its evolution from rule-based methods to neural models, along with key applications and architectures. It then introduces controlled NLG and its techniques before summarizing the NLG tasks explored in this dissertation.

Fourth Chapter: Arabic Gender Rewriting In chapter four, we introduce the task of Arabic gender rewriting and present a new dataset that enables its study. We develop multiple gender rewriting models and demonstrate how gender rewriting can effectively reduce bias in machine translation systems. Lastly, we share key findings and lessons learned from a shared task we organized on Arabic gender rewriting.

Fifth Chapter: Grammatical Error Correction This chapter presents the first results on Arabic grammatical error correction using pretrained sequence-to-sequence models. We also introduce the task of multi-class Arabic grammatical error detection and establish its first benchmark results. We show that conditioning models on error patterns enhances grammatical error correction performance across three datasets from different genres.

Sixth Chapter: Dialectal Text Normalization In this chapter, We introduce the task of dialectal Arabic text normalization and present the first results using pretrained sequence-to-sequence models. Furthermore, we demonstrate that conditioning these models on the input’s dialect significantly improves performance. We present results on five Arabic dialects: Beirut, Cairo, Doha, Rabat, and Tunis.

Seventh Chapter: Text Editing In the seventh chapter, we introduce the first Arabic text editing model that frames NLG as a sequence tagging task. We demonstrate the applicability of this approach on gender rewriting, grammatical error correction, and dialectal text normalization. We show that this model outperforms autoregressive systems, offering significantly faster inference times and making it more suitable for real-world, practical NLG applications.

Eighth Chapter: Summary and Conclusions We conclude with this chapter, which provides a high-level summary of the contributions in this dissertation and the main general conclusions. We also discuss future research directions.

1.3 Contributions

The main contributions of this dissertation are:

1. We address the limitations of English-centricity and limited controllability in NLG for morphologically rich languages by developing controlled NLG approaches specifically designed for Arabic.
2. We focus on three key tasks: gender rewriting, grammatical error correction, and dialectal text normalization. We introduce controlled NLG approaches that

incorporate linguistic traits shaping Arabic writing, including grammatical gender, error types, and dialect.

3. We introduce the novel task of Arabic gender rewriting and present a new dataset to support its study. We develop multiple gender rewriting models and demonstrate how gender rewriting can effectively reduce bias in machine translation systems.
4. We present the first results on Arabic grammatical error correction using pretrained sequence-to-sequence models. We also introduce the task of multi-class Arabic grammatical error detection and establish its first benchmark results. We show that conditioning models on error patterns enhances grammatical error correction performance across three datasets from different genres.
5. We present the first results on dialectal Arabic text normalization using pretrained sequence-to-sequence models. We further demonstrate that conditioning these models on the input’s dialect significantly improves performance.
6. We introduce the first Arabic text editing model that frames NLG as a sequence tagging task. We demonstrate the applicability of this approach on gender rewriting, grammatical error correction, and dialectal text normalization. Our results show that this model outperforms autoregressive systems, offering significantly faster inference times and making it more suitable for real-world, practical NLG applications.

1.4 Publications

Most of the work in this dissertation was previously discussed in the following peer-reviewed conference articles. In the chapters that follow, we elaborate and extend the work presented in these papers:

1. **Bashar Alhafni**, Sarah Al-Towaity, Ziyad Fawzy, Fatema Nassar, Fadhil Eryani, Houda Bouamor, and Nizar Habash. 2024. [Exploiting dialect identification in automatic dialectal text normalization](#). In *Proceedings of The Second Arabic Natural Language Processing Conference*, pages 42–54, Bangkok, Thailand. Association for Computational Linguistics.
2. **Bashar Alhafni**, Go Inoue, Christian Khairallah, and Nizar Habash. 2023. [Advancements in Arabic grammatical error detection and correction: An empirical investigation](#). In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 6430–6448, Singapore. Association for Computational Linguistics.
3. **Bashar Alhafni**, Ossama Obeid, and Nizar Habash. 2023. [The user-aware Arabic gender rewriter](#). In *Proceedings of the First Workshop on Gender-Inclusive Translation Technologies*, pages 3–11, Tampere, Finland. European Association for Machine Translation.
4. **Bashar Alhafni**, Nizar Habash, and Houda Bouamor. 2022. [User-centric gender rewriting](#). In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 618–631, Seattle, United States. Association for Computational Linguistics.

5. **Bashar Alhafni**, Nizar Habash, and Houda Bouamor. 2022. [The Arabic parallel gender corpus 2.0: Extensions and analyses](#). In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 1870–1884, Marseille, France. European Language Resources Association.
6. **Bashar Alhafni**, Nizar Habash, Houda Bouamor, Ossama Obeid, Sultan Alrowili, Daliyah AlZeer, Kawla Mohamad Shnqiti, Ahmed Elbakry, Muhammad ElNokrashy, Mohamed Gabr, Abderrahmane Issam, Abdelrahim Qaddoumi, Vijay Shanker, and Mahmoud Zyate. 2022. [The shared task on gender rewriting](#). In *Proceedings of the Seventh Arabic Natural Language Processing Workshop (WANLP)*, pages 98–107, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.
7. **Bashar Alhafni**, Nizar Habash, and Houda Bouamor. 2020. [Gender-aware reinflection using linguistically enhanced neural models](#). In *Proceedings of the Second Workshop on Gender Bias in Natural Language Processing*, pages 139–150, Barcelona, Spain (Online). Association for Computational Linguistics.

The following preprint is also discussed:

8. **Bashar Alhafni** and Nizar Habash. 2025. [Enhancing Text Editing for Grammatical Error Correction: Arabic as a Case Study](#). arXiv preprint arXiv:2503.00985

Chapter 2

Arabic Linguistic Background

In this chapter, we present key linguistic background on Arabic relevant to this dissertation. Rather than providing a comprehensive overview of the language, our goal is to highlight essential linguistic background that contextualize and motivate our work. We begin with an overview of Arabic and its dialects, followed by a discussion of its rich morphological system. We then discuss Arabic orthography, and the implications of the lack of standardized orthography in dialectal Arabic.

2.1 Arabic and its Dialects

Arabic, a Semitic language, exists along a spectrum of linguistic forms, including Classical Arabic (CA), Modern Standard Arabic (MSA), and Dialectal Arabic (DA). CA is primarily used in religious and classical literary texts, while MSA serves as the standardized form of the language, used in formal media, official communication, culture, and education. In contrast, DA consists of various spoken dialects across the Arab world that lack standardized orthographies but have become increasingly common in written communication, particularly on social media.

Despite representing the formal form of Arabic, MSA is not spoken in daily exchanges, and it is not considered the native language of Arabic speakers. Instead, DA is the primary medium of daily communication. A typical Arabic speaker acquires native proficiency in one of the Arabic dialects and learns to read and write in MSA through formal education (Mohit et al., 2014a). As a result, when writing in MSA, speakers frequently incorporate elements from their dialects, leading to code-mixing at the phonological, morphological, and lexical levels (Habash et al., 2008).

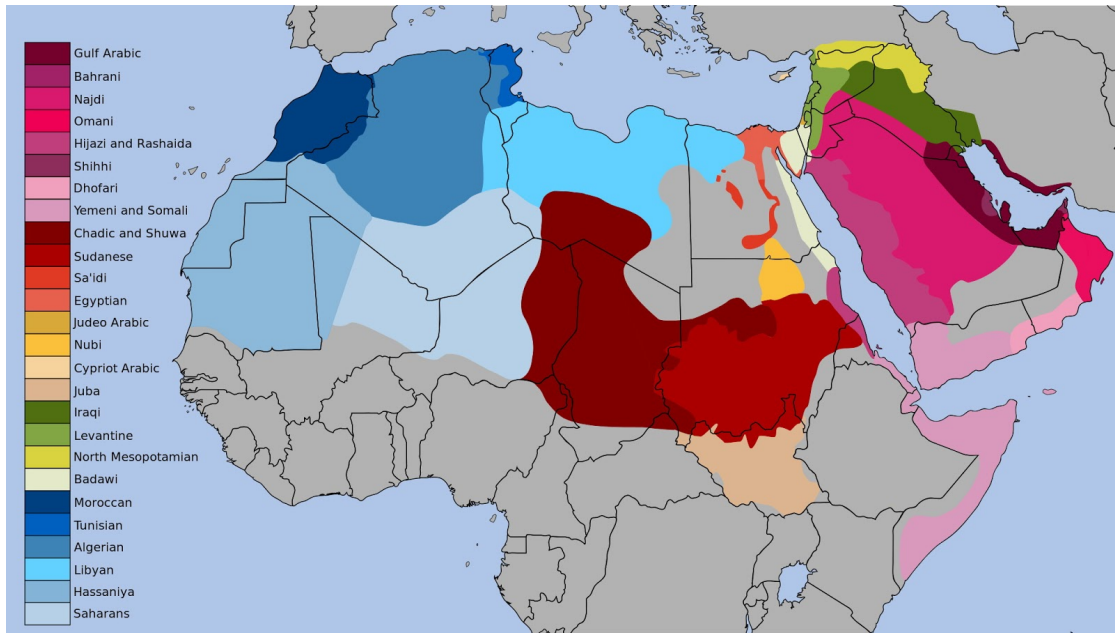


Figure 2.1: The distribution of the different Arabic dialects over the Arab World and surrounding areas (Wikipedia, 2011).

Arabic dialects vary widely, with geographical location being the primary distinguishing factor. Figure 2.1 presents the distribution of dialects across the Arab world. The major dialectal groups are typically classified as follows (Habash, 2010):

- **Egyptian:** Spoken in Egypt and Sudan, including Nile Valley dialects.
- **Levantine:** Includes dialects of Lebanon, Syria, Jordan, and Palestine.
- **Gulf Arabic:** Encompasses dialects spoken in Kuwait, the UAE, Bahrain, and

Qatar. Saudi Arabia is generally included, though it has significant subdialectal variation, and Omani Arabic is sometimes grouped here.

- **North African (Maghrebi):** Covers the dialects of Morocco, Algeria, Tunisia, and Mauritania, with Libyan Arabic sometimes included.
- **Iraqi Arabic:** Shares features with both Levantine and Gulf dialects.
- **Yemeni Arabic:** Often classified as a distinct dialect group.

Additionally, within each of these regional categories, substantial variation exists at the village, town, and city levels, further enriching the linguistic diversity of the Arabic-speaking world.

Despite their similarities, DA and MSA have many differences that prevent MSA tools from being effectively utilized for dialectal text. Arabic dialects diverge from MSA and from each other at the phonological, lexical, and morphological levels (Watson, 2007). Phonologically, for instance, the MSA alveolar affricate ج /j/ is realized differently across dialects: as /g/ in Egyptian, as /ʒ/ in Levantine, and as /y/ in Gulf Arabic. Consequently, the word جميل *jmyl*¹, meaning ‘handsome’, is pronounced as /jamīl/ in MSA, /gamīl/ in Egyptian, /ʒamīl/ in Levantine, and /yamīl/ in Gulf Arabic. Morphologically, DA also differs considerably from MSA. For example, the MSA future proclitic /sa+/ (spelled +س s+) appears in Egyptian Arabic as /Ha+/ (+ح) or /ha+/ (+ه), with both variants occurring without a fixed pattern. Lexically, the differences are even more pronounced. The following are a few examples: in Egyptian *بس bas* ‘only’, *طريضة tarabayzaḥ* ‘table’, *مرات mirAt* ‘wife [of]’, and *دول dawl* ‘these’, correspond to the MSA words *فقط faqaT*, *طاولة TAwilah*, *زوجة zawjaḥ*, and *هؤلاء hawġlĀ*, respectively (Habash et al., 2012a).

Although MSA follows a well-defined standard orthography, Arabic dialects lack such standardization, leading to significant variation in how DA text is written (§2.3).

¹All Arabic transliterations follow the Habash-Soudi-Buckwalter (HSB) transliteration scheme (Habash et al., 2007).

These inconsistencies introduce noise, increase sparsity, and amplify ambiguity in written DA content. Additionally, since most written DA appears in user-generated content on social media, it is further influenced by slang, informal spelling, and frequent errors, making it even more inconsistent and challenging for Arabic NLP systems.

Complicating matters further, annotated DA data remains scarce, making it difficult to develop robust computational models. In this dissertation, we address these challenges by presenting NLG models for both MSA and DA, aiming to improve text generation across different forms of Arabic.

2.2 Arabic Morphology

Morphology studies the internal structure of words, focusing on how their components interact and shape their semantic and syntactic behaviors. Arabic has a rich morphological system that inflects for different combinations of morphological features such as gender, number, person, case, state, aspect, mood and voice, in addition to various attachable clitics such as prepositions, particles, and pronouns ([Habash, 2010](#)). This results in a high number of word inflections, significantly expanding the vocabulary space. For instance, while the total number of words in a large MSA corpus is 20% lower than in its English parallel (a morphologically poor language), the number of unique word types in MSA is nearly double that of English ([Kholly and Habash, 2010](#)).

In addition to its morphological richness, Arabic (including both MSA and DA) exhibits a high degree of ambiguity, primarily due to the multiple possible interpretations of the same words. This ambiguity is further exacerbated by Arabic’s optional diacritization system. One consequence of this high ambiguity is that, on average, an Arabic word has approximately 12 different out-of-context morphological analyses ([Habash, 2010](#)). These

two challenges, morphological richness and ambiguity, are central to why Arabic poses difficulties for NLP: the extensive range of word forms increases data sparsity, while the high level of ambiguity complicates disambiguation.

To address these challenges, various morphological modeling approaches have been developed. One of the earliest solutions is the use of morphological analyzers, also known as morphological dictionaries, which enumerate all possible inflected forms of words in the language (Buckwalter, 2002; Graff et al., 2009). A well-designed morphological analyzer should comprehensively cover all inflected forms of a given lemma (richness) and return all possible analyses of a surface word (ambiguity), with the most appropriate analysis selected through morphological disambiguation. Beyond dictionaries, machine learning-based morphological taggers and disambiguators (Pasha et al., 2014b; Abdelali et al., 2016) have been developed to automate this process, followed by more advanced neural morphological models (Zalmout and Habash, 2019; Zalmout, 2020; Inoue et al., 2022).

In this dissertation, we leverage various morphological analysis and disambiguation resources to introduce control in our NLG models. These resources range from traditional morphological analyzers to more advanced neural-based morphological taggers and disambiguators, enabling more precise and context-aware text generation.

2.3 Arabic Orthography

Orthography deals with the written form of a language, specifying how its sounds are mapped to a particular script. In the case of Arabic, MSA serves as the primary written form, with a well-defined orthographic system. However, despite its standardization, written MSA suffers many orthographic inconsistencies even in professionally written

news articles (Buckwalter, 2004a). These inconsistencies are particularly relevant to grammatical error correction, which we explore in greater detail in Chapter 5.

Beyond these inconsistencies, Arabic orthography allows the use of optional diacritics. Diacritics in Arabic primarily convey phonological information that complements the consonant-based Abjad script. While Arabic has many diacritics, the basic set used in most MSA contexts consists of nine symbols: vowel diacritics (Fatha َ, Damma ُ, Kasra ِ) indicate short vowels; nunation diacritics (ّ, ٌ, ٍ) indicate a short vowel followed by /n/; the gemination diacritic, Shadda ّ, indicates doubling of the consonant letter it follows; the Sukun ْ (silence) diacritic indicates that no vowel is present; and finally, the special elongation diacritic ِ (aka Dagger Alif) indicates a long /ā/ vowel. In MSA written text, diacritics are usually omitted, leaving readers to infer the meaning of certain words based on the context (Habash, 2010). This leads to ambiguity, as in the case of the verb كُنت *knt*, which can be read as *kuntu* ‘I was’, *kunta* ‘You [masculine] were’, or *kunti* ‘You [feminine] were’. While this allows the same text to be interpreted differently based on the context, which can be beneficial in text generation, it poses major challenges for disambiguation and detection systems, which must account for multiple possible readings of the same word or phrase.

Furthermore, orthography is much more challenging in DA compared to MSA. While MSA has a well-defined standard orthography, Arabic dialects do not. When speakers write in DA, they often do so in a way that reflects either the phonology or the etymology of the words. As a result, apart from unintentional typographical errors, no spelling of a dialectal word can be deemed truly “incorrect”. This phenomenon, known as *spontaneous orthography* (Eskander et al., 2013). For instance, the word for ‘small [feminine singular]’ in the Beirut dialect, /zʁi:ri/, can be written in a range of spontaneous Arabic spellings, some of which highlighting its phonology and others its etymological connections to

MSA صغيرة $S\gamma yr\hbar$ /s^ʕavira[t]/. These include: زغيري $z\gamma yry$, زغيره $z\gamma yrh$, زغيرة $z\gamma yr\hbar$, صغيري $S\gamma yry$, صغيره $S\gamma yrh$, and صغيرة $S\gamma yr\hbar$. This makes DA particularly challenging for many NLP tasks. To address the lack of standardized orthography for DA, (Habash et al., 2012a) proposed CODA, a Conventional Orthography for Dialectal Arabic. Applying CODA to DA text reduces sparsity and noise, which can lead to better modeling. We discuss CODA in more detail in Chapter 6. In this dissertation, we introduce DA text normalization models that normalizes DA into the CODA convention.

Chapter 3

Natural Language Generation

Background

Automatic text generation has been a longstanding goal in computer science, evolving from early rule-based systems to modern neural approaches. This chapter provides an overview of NLG methodologies, covering rule-based, statistical, and neural models, along with applications and architectures used in text generation. We then introduce controlled NLG, discussing different techniques for guiding text generation. Finally, we summarize the NLG tasks explored in this dissertation.

3.1 Natural Language Generation

Natural language generation (NLG) refers to any task involving the production of text, such as translation, summarization, or dialogue generation. NLG is inherently complex, requiring decisions about what content to include, how to structure it, and how to express it coherently. This section provides an overview of key NLG approaches.

3.1.1 A Short History of Natural Language Generation

Rule-based NLG: Early NLG systems relied on template- and grammar-based approaches, which were highly structured and consisted of multiple components responsible for different stages of text generation. These included content determination to select relevant information, document structuring to arrange it coherently, aggregation to merge similar sentences, lexical choice to ensure precise and natural wording, and realization to generate the final text (Reiter and Dale, 1997). Such approaches were often rigid, requiring extensive manual effort to design templates and domain-specific rules.

Statistical NLG: To overcome the limitations of rule-based systems, NLG shifted toward statistical methods, enabling more flexible, data-driven text generation. This transition introduced statistical language modeling, which assigns probabilities to sequences of words. Formally, given a sequence of words w_1, w_2, \dots, w_n , a language model estimates the likelihood $P(w_1, w_2, \dots, w_n)$. An ideal model assigns high probability to natural-sounding text and low probability to incoherent sequences. Most language models assume that the probability of a word is dependent only on the words preceding it. By applying the *chain rule of probability*, the likelihood of a sentence can be decomposed as follows:

$$P(w_1, w_2, \dots, w_n) = P(w_1)P(w_2|w_1)P(w_3|w_1, w_2)\dots P(w_n|w_1, \dots, w_{n-1})$$

A widely used form of statistical language modeling is the n -gram model. Instead of trying to estimate the probability of a word given all preceding words, n -gram models make the Markov assumption that the probability of a word depends only on the previous

$n-1$ words. For example, a 3-gram language model (trigram) estimates:

$$P(w_i|w_1, ..w_{i-1}) \approx P(w_i|w_{i-2}, w_{i-1})$$

An n -gram model is constructed from a corpus of text by simply counting how many times each word in the text is preceded by each possible n -gram. This makes n -gram models easier to train than grammar-based approaches, which require manually labeled training data. Due to their simplicity and efficiency, we use n -gram models in this dissertation for various NLG tasks.

However, n -gram language models have several limitations. First, they suffer from data sparsity, assigning zero probability to unseen (n -gram, word) pairs, which requires smoothing techniques. Second, their computational cost also grows exponentially with n , making long-range dependencies difficult to capture. In practice, most models use n between 1 and 5, which is insufficient for maintaining coherence. Third, they struggle with out-of-vocabulary (OOV) words, as they can only generate words encountered during training. Neural language models, described next, address many of these limitations.

Neural NLG: Neural network-based language models replace traditional statistical models with a learned function (the neural network) that predicts the likelihood of a word sequence. Unlike n -gram models, neural models can assign nonzero probabilities to unseen sequences and capture longer dependencies. State-of-the-art neural language models can process sequences thousands of words long.

One of the key advances in neural language modeling was the transition from operating on sequences of discrete words to operating on sequences of continuous vector representations. Instead of treating words as independent symbols, modern neural models represent each word w_t as a dense embedding y_t . Early neural language models used pretrained

word embeddings, such as word2vec (Mikolov et al., 2013) and GloVe (Pennington et al., 2014), where embeddings were learned separately from the language model. In contrast, modern neural language models jointly optimize embeddings along with the rest of the network, treating the embedding matrix as a trainable parameter.

At each time step t , a neural language model predicts the next word based on the preceding words, similar to an n -gram model but without fixed context limitations. Given a sequence of words w_1, w_2, \dots, w_n , the model estimates a probability distribution over all possible next words:

$$P(w_t|w_1, \dots, w_{t-1}) = \frac{\exp(f_\theta(w_1, \dots, w_{t-1})_i)}{\sum_j \exp(f_\theta(w_1, \dots, w_{t-1})_j)}$$

where f_θ is a neural network parameterized by θ , which takes the previous words as input and generates a score for each word in the vocabulary. These scores are then normalized using the softmax function, producing a probability distribution. The model selects the next word using a decoding strategy, such as greedy decoding or beam search.

In many language modeling applications, the model is conditioned on an external input in addition to the preceding words. This means it can generate text not only by continuing a sequence but also by incorporating information from a prompt, structured data, or another input source. Given an input sentence $X = x_1, x_2, \dots, x_m$ and a target sentence $Y = y_1, y_2, \dots, y_n$, a conditional neural language model estimates:

$$P(Y|X) = \prod_{t=1}^n P(y_t|y_1, \dots, y_{t-1}, X)$$

This modeling paradigm, also known as an encoder-decoder or sequence-to-sequence (Seq2Seq), is widely used in machine translation, text summarization, and dialogue generation. We adopt Seq2Seq models for all NLG tasks in this dissertation.

3.1.2 Types of Natural Language Generation

NLG can be categorized into several types based on the nature of their inputs. Following [Li et al. \(2021\)](#), we group NLG applications into the following categories:

- **Unconditional Generation:** This type of generation does not rely on any explicit input beyond an initial prompt or starting token. As discussed earlier, unconditional language models estimate the probability of a word sequence without external conditioning. Common applications include story generation ([Ma et al., 2024](#)) and open-ended text continuation ([Radford et al., 2019](#)).
- **Conditional Generation:** In conditional generation, the model generates text based on a given input sequence. As covered earlier, this setup is widely used in machine translation ([Sutskever et al., 2014](#)), summarization ([Rush et al., 2015](#)), and dialogue generation ([Li et al., 2016](#)).
- **Attributed-based Generation:** Here, the model is conditioned on both an input sequence and a set of explicit attributes, such as sentiment, style, or author traits. Applications include style transfer ([Keskar et al., 2019](#); [Krause et al., 2021](#)) and personalized text generation ([Alhafni et al., 2024c](#); [Zhang et al., 2024](#)).
- **Data-to-Text Generation:** This type of generation converts structured data such as tables ([Parikh et al., 2020](#)), knowledge graphs ([Wiseman et al., 2017](#)), or database entries ([Novikova et al., 2017](#)) into natural language text.
- **Multimedia-to-Text Generation:** In multimedia-to-text generation, the model generates text based on non-textual inputs such as images, videos, or audio. Applications include image captioning ([Chen et al., 2015](#)), video captioning ([Long et al., 2018](#)), and speech recognition ([Yadav and Sitaram, 2022](#)).

This dissertation focuses on unconditional, conditional, and attribute-based NLG models.

3.1.3 Neural Architectures for Natural Language Generation

Given the sequential nature of language, early neural text generation models primarily relied on recurrent neural networks (RNNs), such as long short-term memory (LSTM) networks (Hochreiter and Schmidhuber, 1997) and gated recurrent units (GRUs) (Cho et al., 2014). Although less common, convolutional neural networks (CNNs) have also been explored for text generation (Kalchbrenner et al., 2017). Conditional generation models typically adopt an encoder-decoder or Seq2Seq architecture (Sutskever et al., 2014), where an encoder processes the input sequence and a decoder generates the output. The introduction of the attention mechanism between the encoder and decoder significantly improved performance in machine translation (Bahdanau et al., 2015) and was soon adopted across various NLG tasks.

More recently, self-attention-based architectures, particularly the Transformer (Vaswani et al., 2017), have become the dominant paradigm for state-of-the-art NLG. Transformers offer several advantages over their recurrent predecessors, the most notable being parallelization—operations are applied to all tokens in a sequence simultaneously, rather than sequentially as in RNNs. This drastically reduces training time and makes computation independent of sequence length. Additionally, Transformers are far better at capturing long-range dependencies, enabling them to establish relationships between words and phrases that may be far apart in a text. These advantages have driven the scaling of Transformers and their widespread adoption in large language models (LLMs), which are now at the forefront of NLG research and applications.

Most neural text generation models are trained using backpropagation to maximize the log-likelihood of predicting the next word given the preceding words: $P(w_t | w_1, \dots, w_{t-1})$. A major breakthrough in NLG has been self-supervised pretraining, where models learn

from large amounts of raw text without requiring labeled data. This approach has led to significant improvements across various NLP tasks (Peters et al., 2018; Devlin et al., 2019; Liu et al., 2019; Yang et al., 2019b) and is now the de facto learning paradigm: models are first pretrained on massive text corpora and then fine-tuned on specific downstream tasks. Modern pretrained models are predominantly based on the Transformer architecture. For text generation, pretraining strategies typically fall into two categories: decoder-only language models, such as GPTs (Radford et al., 2019; Brown et al., 2020; OpenAI et al., 2024), which predict text autoregressively, and encoder-decoder (SeqSeq) models, such as BART (Lewis et al., 2020; Liu et al., 2020c) and T5 (Raffel et al., 2020; Xue et al., 2021), which learn to reconstruct corrupted text sequences. In this dissertation, we leverage pretrained language models for all NLG tasks we study.

3.2 Controlled Natural Language Generation

Controlled NLG is the task of generating text that adheres to a given attribute or condition while maintaining fluency and coherence. The controlled attribute can take various forms depending on the application, ranging from text characteristics (such as sentiment, topic, or keywords) to author-related traits (such as writing style, gender, or age). This approach, often referred to as attribute-based generation, can be formalized as follows:

$$P(Y|X, A) = \prod_{t=1}^n P(y_t|y_1, \dots, y_{t-1}, X, A)$$

where X is the input sequence, Y is the generated output, and A represents the conditioning attribute. This paradigm also extends to unconditional generation, where text is generated solely based on the specified attribute, modeled as: $P(Y|A)$.

Controlled NLG methods fall into content control (hard control) and attribute control

(soft control) (Liang et al., 2024). Content control explicitly constrains text structure and word choice, enforcing strict linguistic rules (Hua and Wang, 2020; He, 2021; Yang and Klein, 2021; Lin and Riedl, 2021; Lu et al., 2022; Liu et al., 2022; Chai et al., 2022; Juseon-Do et al., 2024; Jie et al., 2024). In contrast, attribute control operates at a higher level, influencing broad characteristics like sentiment (Keskar et al., 2019; Krause et al., 2021), style (Trotta et al., 2022; Alhafni et al., 2024c; Zhang et al., 2024), or topic rather than dictating specific words (Dathathri et al., 2020; Chan et al., 2022). Hard control enforces predefined constraints, while soft control allows for greater flexibility, making it particularly useful for stylistic adaptation and personalized generation.

Control can be integrated into text generation either during training or at inference. Training-time approaches include retraining, where models are trained from scratch on attribute-specific data (Keskar et al., 2019; He, 2021); fine-tuning (Zhang and Song, 2022; Zhou et al., 2023b), which adapts pretrained models by incorporating control attributes; and reinforcement learning (Upadhyay et al., 2022; Dai et al., 2024), which optimizes model outputs using reward signals. At inference time, control mechanisms are applied in real-time through prompt engineering (Lester et al., 2021; Li and Liang, 2021; Liu et al., 2023), which guides output by modifying input prompts; latent space manipulation (Subramani et al., 2022; Liu et al., 2024; Turner et al., 2024), which adjusts internal model activations to influence text attributes; and decoding-time interventions (Dathathri et al., 2020; Krause et al., 2021; Yang and Klein, 2021) which modify probability distributions or impose constraints during generation.

In this dissertation, we explore attribute-controlled NLG by conditioning models on linguistic attributes that influence Arabic writing, including grammatical gender, dialect, and error patterns. These techniques enable fine-grained control in the NLG tasks we explore, enhancing adaptability to diverse user preferences and linguistic contexts.

3.3 Language Generation Tasks in this Dissertation

This dissertation explores three language generation tasks. A brief summary of each task is provided here:

Arabic Gender Rewriting: We define gender rewriting as the task of generating alternative versions of a sentence based on a given target user’s grammatical gender. This task is particularly relevant for gender-marking, morphologically rich languages like Arabic, where ensuring inclusivity in NLG applications requires generating outputs that align with a user’s identity. To our knowledge, this is a novel task, with no prior work in Arabic or any other language. In Chapter 4, we introduce the task, review related work, and describe the datasets and models developed for it. The research presented in this chapter is based on findings from [Alhafni et al. \(2020, 2022a,b,c, 2023b\)](#).

Grammatical Error Correction: Grammatical Error Correction (GEC) aims to correct spelling and grammatical errors, making it valuable for educational applications and writing assistance tools. In this dissertation, we focus on GEC for MSA and introduce Arabic Grammatical Error Detection (GED) as a distinct task. Our findings demonstrate that conditioning GEC models on GED predictions significantly enhances performance. Chapter 5 presents the task, reviews related work, and details our contributions to Arabic GEC. The research presented in this chapter is based on findings from [Alhafni et al. \(2023a\)](#).

Dialectal Text Normalization: Text normalization involves mapping non-canonical text to a standardized form, which is particularly crucial for Dialectal Arabic (DA) due to its lack of standardized orthography (§2). Normalization plays a dual role: as an

upstream task, it mitigates data sparsity and variation to facilitate downstream NLP applications; as a downstream task, it enhances text readability. In this dissertation, we explore CODAfication, the task of normalizing DA text into the CODA convention. Our results show that conditioning NLG models on the writer’s dialect improves performance. Chapter 6 introduces CODA and the CODAfication task, detailing our contributions in this space. The research presented in this chapter is based on findings from [Alhafni et al. \(2024a\)](#).

Chapter 4

Arabic Gender Rewriting

This chapter introduces the task of gender rewriting and presents the Arabic Parallel Gender Corpus, a novel dataset designed for gender identification and rewriting in contexts where one or two target users (first-person “I” and/or second-person “You”) have independent grammatical gender preferences. We develop and evaluate various gender rewriting systems, including a joint model and a multi-step approach, both of which combine the strengths of rule-based and neural methods. Additionally, we benchmark open-source and commercial LLMs to assess their performance on gender rewriting. Our models establish a strong benchmark on the newly introduced corpus, demonstrating their effectiveness. Furthermore, we showcase a practical application of our gender rewriting systems by post-editing machine translation outputs, ensuring they align with users’ grammatical gender preferences and reducing gender bias when translating from English to Arabic. Additionally, we introduce a web-based gender rewriting system that allows users to interact with our models seamlessly. Finally, we share insights from organizing a shared task on Arabic gender rewriting, highlighting key challenges and lessons learned.

4.1 Introduction

The remarkable progress in NLP has raised expectations about user experience, particularly regarding gender identity representation. Gender stereotypes, both negative and positive, are embedded in most of the world’s languages (Maass and Arcuri, 1996; Menegatti and Rubini, 2017) and are further propagated and amplified by NLP systems (Sun et al., 2019). This not only degrades user experiences but also contributes to representational harms (Blodgett et al., 2020). While human-generated data used to train NLP systems is often considered the primary source of bias, merely balancing or debiasing training data does not necessarily mitigate these biases. This is because most NLP systems are designed to generate a single text output without accounting for user-specific grammatical gender preferences. To address this, NLP systems should integrate users’ grammatical gender preferences whenever available to ensure accurate and inclusive text generation.

Ensuring user-aware gender representation becomes even more challenging in multi-user contexts, where different users (first, second, and third persons) have independent grammatical gender preferences. One example of this phenomenon is the machine translation of the sentence *I am a doctor and you are a nurse*. While English uses gender neutral terms leading to ambiguous gender references for the first and second persons (*I/doctor* and *you/nurse*), some morphologically rich languages use gender-specific terms for these two expressions. For instance, in Arabic, a gender-unaware single-output machine translation from English often results in *أنا طبيب وأنت ممرضة* *Âna Tbyb wÂnt mmrDh* ‘I am a [male] doctor and you are a [female] nurse’, which is inappropriate for female doctors and male nurses, respectively. Alternatively, user-aware personalized NLP systems should be designed to produce outputs that are as gender-specific as the user

information they have access to. Users information could be either explicitly embedded as part of the input (e.g., ‘she is a doctor and he is a nurse’) or provided externally by the users themselves. However, contextual complexity and the lack of gender-annotated resources in morphologically rich languages make this task particularly challenging.

In this chapter, we define the task of gender rewriting as generating alternatives of a given Arabic sentence to match different target user gender contexts: a *male speaker* with a *male listener*, a *female speaker* with a *male listener*, a *male speaker* with a *female listener*, and a *female speaker* with a *female listener*. The user-specified gender preferences are treated as part of the input to guide the rewriting process. This requires changing the grammatical gender (masculine or feminine) of certain words referring to the users (speaker/first person and listener/second person). Formally, given an input X that combines both the input Arabic sentence and the target gender preferences, and a sequence of word-level gender labels G corresponding to the input Arabic sentence, the goal is to generate a rewritten version Y that matches the user specified gender preferences:

$$P(Y|X, G) = \prod_{t=1}^n P(y_t|y_1, \dots, y_{t-1}, X, G)$$

To facilitate this task, we introduce the Arabic Parallel Gender Corpus, a new dataset designed for gender identification and rewriting. We develop and benchmark various gender rewriting systems on this corpus and demonstrate their effectiveness. Furthermore, we show how these systems can be leveraged to mitigate gender bias in English-to-Arabic machine translation. Finally, we introduce a web-based application that enables users to interact with our gender rewriting models and share insights gained from organizing a shared task on gender rewriting.

4.2 Background and Related Work

Several approaches have been proposed to mitigate gender bias in various NLP tasks (Stanczak and Augenstein, 2021) including machine translation (Rabinovich et al., 2017; Elaraby et al., 2018; Vanmassenhove et al., 2018; Escudé Font and Costa-jussà, 2019; Stanovsky et al., 2019; Costa-jussà and de Jorge, 2020; Gonen and Webster, 2020; Saunders and Byrne, 2020; Bentivogli et al., 2020; Saunders et al., 2020; Stafanovičs et al., 2020; Saunders et al., 2021; Savoldi et al., 2021; Ciora et al., 2021; Savoldi et al., 2023, 2024; Sánchez et al., 2024; Garg et al., 2024), dialogue systems (Cercas Curry et al., 2020; Dinan et al., 2020a; Liu et al., 2020a,b; Sheng et al., 2021b,a), language modeling (Lu et al., 2018; Bordia and Bowman, 2019; Sheng et al., 2019; Vig et al., 2020; Kaneko et al., 2022; Kotek et al., 2023; Levy et al., 2024), co-reference resolution (Rudinger et al., 2018; Zhao et al., 2018a; Cao and Daumé, 2021), and named entity recognition (Mehrabi et al., 2019; Cimitan et al., 2024).

Approaches to mitigating gender bias include debiasing word embeddings (contextualized or non-contextualized) before using them in downstream tasks (Bolukbasi et al., 2016; Zhao et al., 2018b; Gonen and Goldberg, 2019; Manzini et al., 2019; Zhao et al., 2020; Lauscher et al., 2020; Katsarou et al., 2022), classifying gender bias along multiple dimensions (Dinan et al., 2020b), adding additional information to the input to enable models to capture gender information correctly (Vanmassenhove et al., 2018; Moryossef et al., 2019; Stafanovičs et al., 2020; Saunders et al., 2020; Basta et al., 2020), or training models on gender-balanced corpora created through counterfactual data augmentation techniques (Madaan et al., 2018; Park et al., 2018; Lu et al., 2018; Maudslay et al., 2019; Zmigrod et al., 2019; Emami et al., 2019; Costa-jussà and de Jorge, 2020; Bartl et al., 2020; de Vassimon Manela et al., 2021; Sen et al., 2021). In terms of rewriting, Van-

[massenhove et al. \(2021\)](#) and [Sun et al. \(2021\)](#) presented rule-based and neural rewriting models to generate gender-neutral sentences. [Garg et al. \(2024\)](#) used LLMs to machine translation outputs to generate all possible gender alternatives.

Most existing work focuses on English and may not generalize well to morphologically rich languages. Nevertheless, there have been recent studies that explored the gender bias problem in languages other than English. [Zhao et al. \(2020\)](#) studied gender bias which is exhibited by multilingual embeddings in four languages (English, German, French, and Spanish) and demonstrated that such bias can impact cross-lingual transfer learning tasks. [Zmigrod et al. \(2019\)](#) used a counterfactual data augmentation approach and developed a generative model to convert between masculine and feminine sentences in four languages (French, Hebrew, Italian, and Spanish). [Vanmassenhove and Monti \(2021\)](#) introduced an English-Italian dataset where the English sentences are gender annotated at the word-level and paired with multiple gender alternative Italian translations when needed. [Bentivogli et al. \(2020\)](#) proposed MuST-SHE, a multilingual benchmark for analyzing gender bias in machine translation and speech translation, containing English-Spanish/French/Italian spoken data from TED talks. [Nozza et al. \(2021\)](#) introduced a benchmark dataset of manually crafted sentence templates to assess gender-stereotyped completions across six languages: English, Italian, French, Portuguese, Romanian, and Spanish. [Muller et al. \(2023\)](#) developed an automated pipeline to quantify gender representation bias in large-scale multilingual datasets for machine translation across 55 languages. [Savoldi et al. \(2025\)](#) introduced mGeNTe, a multilingual extension of the GeNTe corpus ([Piergentili et al., 2023](#)), which provides manually curated parallel data for gender neutralization in English-Italian, English-Spanish, and English-German.

For Arabic, [Habash et al. \(2019\)](#) introduced the first version of the Arabic Parallel Gender Corpus, a gender-annotated parallel dataset of first-person singular sentences.

Each sentence was labeled according to the speaker’s grammatical gender as feminine, masculine, or ambiguous, with corresponding gendered rewrites provided for the masculine and feminine sentences. They also developed a two-step gender rewriting system: a feature-based classifier for gender identification and a character-level seq2seq model for rewriting. Their system was applied to machine translation post-editing to produce gender-specific translations.

Our work extends the Arabic Parallel Gender Corpus by incorporating contexts that involve both first and second grammatical persons, covering singular, dual, and plural constructions, and expanding the dataset to six times its original size. In this chapter, we detail the creation process of this extended corpus and introduce our gender rewriting systems.

Arabic Grammatical Gender

Arabic has a rich and complex morphological system (§2) that inflects for various morphological features, including gender, number, person, case, state, aspect, mood, and voice, as well as several attachable clitics such as prepositions, particles, and pronouns. Arabic nouns, adjectives, and verbs inflect for gender: masculine (M) and feminine (F), and for number: singular (S), dual (D) and plural (P). Changing the grammatical gender of Arabic words involves either changing the *base word*, altering *pronominal enclitics* that are attached to the base word, or a combination of both. A *base word* in Arabic refers to the stem along with its attachable affixes (prefixes, suffixes, circumfixes). Changing the *base word* gender requires either a suffix change, a pattern change, or a lexical change as shown in Table 4.1. Arabic also has clitics that attach to the stem after affixes. A clitic is a morpheme that has the syntactic characteristics of a word but shows evidence of being phonologically bound to another word. In this respect, a clitic is distinctly different

Paired Gender Alternatives		Rewrite Type
أمير <i>Âmyr</i> (NOUN.MS) prince	أميرة <i>Âmyrh</i> (NOUN.FS) princess	Suffix Change (a)
أحمر <i>ÂHmr</i> (ADJ.MS) red	حمراء <i>HmrA'</i> (ADJ.FS) red	Pattern Change (b)
أخ <i>Âx</i> (NOUN.MS) brother	أخت <i>Âxt</i> (NOUN.FS) sister	Lexical Change (c)
أميركم <i>Âmyr+km</i> (NOUN.MS+PRON.2MP) your (MP) prince	أميرتكن <i>Âmyr+kn</i> (NOUN.MS+PRON.2FP) your (FP) prince	Enclitic Change (d)
أمراء <i>ÂmrA'</i> (NOUN.MP) princes	أميرات <i>ÂmyrAt</i> (NOUN.FP) princesses	Pattern Change + Suffix Change (e)
أميركم <i>Âmyr+km</i> (NOUN.MS+PRON.2MP) your (MP) prince	أميرتكن <i>Âmyrt+kn</i> (NOUN.FS+PRON.2FP) your (FP) princess	Suffix Change + Enclitic Change (f)
أمراءهم <i>ÂmrA'+hm</i> (NOUN.MP+PRON.3MP) their (MP) princes	أميراتهن <i>ÂmyrAt+hn</i> (NOUN.FP+PRON.3FP) their (FP) princesses	Pattern Change + Suffix Change + Enclitic Change (g)

Table 4.1: Examples of the changes needed to generate gender alternative forms of gender-specific words in Arabic.

from an affix, which is phonologically and syntactically part of the word. Proclitics are clitics that precede the word (like a prefix), whereas enclitics are clitics that follow the word (like a suffix). *Pronominal enclitics* are pronouns that cliticize to previous words (Table 4.1(d)). It is worth noting that multiple affixes and clitics can appear in a single word in Arabic and changing the grammatical gender of such words requires changing the genders of both the base word and its clitics (Table 4.1(f-g)).

4.3 The Arabic Parallel Gender Corpus

The first version of the Arabic Parallel Gender Corpus (APGC v1.0) was introduced by [Habash et al. \(2019\)](#). It consists of gender-annotated and rewritten first-person singular Arabic sentences, sourced from a subset of the English-Arabic OpenSubtitles 2018 dataset ([Lison and Tiedemann, 2016](#)). Each sentence is labeled based on the grammatical gender of its singular speaker as F (feminine), M (masculine), or B (invariant/ambiguous).

For the M and F sentences, they introduced their parallel opposite gender forms. In this dissertation, we expand APGC v1.0 and introduce APGC v2.0 by adding second person targets as well as increasing the total number of sentences over 6.5 times, reaching over 590K words. The new corpus consists of multiple parallel components: four combinations of first- and second-person grammatical gender (masculine and feminine), English source sentences, and English-to-Arabic machine translation outputs. We describe the selection process and annotation guidelines for APGC v2.0 in (§4.3.1, §4.3.2, §4.3.3) and provide an overview and analysis of the corpus in (§4.3.4).

4.3.1 Corpus Selection

As in [Habash et al. \(2019\)](#), we selected the original set of sentences from the English-Arabic OpenSubtitles 2018 dataset ([Lison and Tiedemann, 2016](#)), which includes 29.8 million English-Arabic sentence pairs. We chose OpenSubtitles because it has parallel sentences in English and because it is full of conversational (first and second person) texts in MSA. We extracted all the pairs that include first or second person pronouns on the English side: *I, me, my, mine, myself*, and *you, your, yours, yourself*. This selection process identified 13.4 million pairs: 2.8 million (21.1%) include first and second person pronouns, 5.7 million (42.5%) include only first person pronouns, and 4.9 million (36.4%) include only second person pronouns.

Out of this set, we randomly selected 52,000 English-Arabic pairs to be manually annotated, while maintaining the original first and second person sentences proportions: 10,972 (21.1%) pairs contain first and second person pronouns on the English side, 22,100 (42.5%) pairs contain only first person pronouns on the English side, 18,928 (36.4%) pairs contain only second person pronouns on the English side. To be consistent with APGC v1.0’s preprocessing, we ran the Arabic sentences through MADAMIRA ([Pasha](#)

[et al., 2014b](#)) to do white-space-and-punctuation tokenization and UTF-8 cleaning.

In addition to the above, we re-annotated all of the 11,240 sentences from APGC v1.0 to include second person references and match our extended guidelines completely. In total, this resulted in 63,240 English-Arabic sentence pairs for the next annotation step.

4.3.2 Corpus Annotation

Four professional linguists (three females and one male), all of whom are native speakers of Arabic, were hired through a linguistic annotation firm, to complete the task.

Gender Identification First, the annotators were asked to identify the genders of the first and second person references in each sentence, then assign to each sentence a two-letter label, where each letter refers to the gender of the first and second person references, respectively. Each letter in the label can have one of three values: F (feminine), M (masculine), or B (invariant/ambiguous). Therefore, each sentence will get a label from one of the nine different label combinations: B, 1FB, 1MB, B2F, B2M, 1M2M, 1F2M, 1M2F, or 1F2F. Additionally, the annotators were asked to identify the dual and plural gendered references. If present, the sub-label corresponding to the gender of the first or second person reference would get an extra mark: “!” (e.g., BF!, M!B!, etc.).

Gender Rewriting In the case of an F or M sub-label, the annotators were asked to copy the sentence and modify it to obtain the opposite gender forms. The modifications are strictly limited to morphological reinflections and word substitutions as was done in [Habash et al. \(2019\)](#). Therefore, the total number of words is maintained along with a perfect alignment between each sentence and its parallel opposite gender forms. For example, the sentence in Table 4.2(c) includes a first person gender reference and is

English	Arabic	Label	Rewrite Label	Rewrite	
I wanna thank you	أريد أن أشكرك	B			(a)
I'm so happy for you	أنا سعيد من أجلك	1FB	1MB	أنا سعيد من أجلك	(c)
We were coming to see you	نحن قادمون لرؤيتك	1FB	1MB	نحن قادمون لرؤيتك	(d)
Because I'm your big brother	لأنني أخوك الكبير	1MB	1FB	لأنني أختك الكبيرة	(e)
We're ready	نحن مستعدون	1MB	1FB	نحن مستعدات	(f)
I know, babe	أعلم ذلك يا عزيزتي	B2F	B2M	أعلم ذلك يا عزيزي	(g)
I respect you [plural]	أنا أحترمكم	B2F	B2M	أنا أحترمكم	(h)
I'm right here dad	أنا هنا يا أبي	B2M	B2F	أنا هنا يا أمي	(i)
I love you [plural] so much	أحبكم كثيرا	B2M	B2F	أحبكن كثيرا	(j)
Baby, I'm so scarasberry right now	أنا خائفة للغاية يا عزيزي	1F2M	1M2M	أنا خائف للغاية يا عزيزي	(n)
			1F2F	أنا خائفة للغاية يا عزيزتي	(o)
			1M2F	أنا خائف للغاية يا عزيزتي	(p)
I'm glad you made it home, mom	أنا سعيد بعودتك يا أماه	1M2F	1F2F	أنا سعيدة بعودتك يا أماه	(q)
			1M2M	أنا سعيد بعودتك يا أبتاه	(r)
			1F2M	أنا سعيدة بعودتك يا أبتاه	(s)

Table 4.2: Examples from the APGC v2.0 including the original sentence, its gender label, its rewrite gender label, and its rewrite to the opposite grammatical gender where appropriate. First person gendered words are in purple and second person gendered words are in pink. The two-letter label specifies gender information of first person (first letter) and second person (second letter). M is Masculine; F is Feminine; and B is invariant.

labeled by the annotators as 1FB, and therefore, the annotators would introduce its gender cognate 1MB. If the sentence includes both first and second person gender references (1M2M, 1F2M, 1M2F, or 1F2F), the annotators would then introduce all its possible gender cognates, as in Table 4.2(n-s) for instance.

In the vast majority of cases, the opposite gender forms of most words end up sharing the same lemma (reinflection), e.g., والد *wAld* ‘parent/father [M]’ and والدة *wAldh* ‘parent/mother [F]’. However, there are cases where gender-specific words have to be mapped to different lemmas, resulting in a lexical change. For instance, أبي *Âby* ‘my dad’ and أمي *Âmy* ‘my mom’ (Table 4.2(i)), or أخوك *Âxwk* ‘your brother’ and أختك *Âxtk*

‘your sister’ (Table 4.2(e)). Furthermore, the annotators were instructed to avoid any heterocentric assumptions during the annotation. For example, the sentence أنت زوجي *Ânt zwjy* ‘you are my husband’ is labeled as B2M (ambiguous first person, masculine second person) and not 1F2M (feminine first person, masculine second person). The annotators were also instructed to treat all proper names as gender-ambiguous (B), even when they have strong gender-specific associations, and as such are not rewritten. Finally, the annotators were asked to flag bad translations and malformed sentences.

4.3.3 Automatic Word-Level Annotations

Since the annotators were only allowed to perform grammatical inflections and word substitutions, all sentences and their parallels are perfectly aligned at the word level. This allowed us to obtain word-level gender annotations automatically as a byproduct. Since gender information could be expressed at different parts of Arabic words (§4.2), we mark the genders of both the base words and their pronominal enclitics. To do this, we look at the original sentence and all of its parallel forms. If the word is the same across all the parallel versions of a sentence, then we label it as B. Otherwise, we check if the word ends with a gender marking pronominal enclitic, we label the gender of the enclitic based on predefined rules as 1F, 1M, 2F, or 2M. If the gendered word does not end with a gender-marking enclitic, then we label the enclitic as B. Once the enclitic is labeled, we compare the base form of the word across its parallel forms. If the base form is the same, we label it as B. Otherwise, we assign the base form the same label as its sentence-level gender label. This results in 25 possible word-level gender labels (e.g., B+1F, 1F+2M).

For example, in Table 4.3(d-g), the word أنا *Âna* ‘I’ is the same across all four parallel versions of the sentence and thus labeled as B. In contrast, the words سعيدة *sçydh* ‘happy [F]’ and سعيد *sçyd* ‘happy [M]’ change across the parallel versions. By

English	Arabic	Label	
I want to talk to you	أريد أن أتحدث معك B B B B	B	(a)
I am going to my office	أنا ذاهبة لكتبي B 1F+B B	FB	(b)
	أنا ذاهب لكتبي B 1M+B B	MB	(c)
I am glad to know you [plural]	أنا سعيدة بمعرفتكم B+2M 1F+B B	FM	(d)
	أنا سعيد بمعرفتكم B+2M 1F+B B	MM	(e)
	أنا سعيد بمعرفتكن B+2F 1M+B B	MF	(f)
	أنا سعيدة بمعرفتكن B+2F 1F+B B	FF	(g)

Table 4.3: Examples of word-level gender annotation. First person gendered words are in purple and second person gendered words are in pink.

looking at the sentence-level labels of the four parallel forms and since they do not end with enclitics, we can deduce that the word سعيدة $s\epsilon ydh$ is first-person feminine and label it 1F+B, and that the word سعيد $s\epsilon yd$ is first-person masculine and labeled it 1M+B. Similarly, we determine that the words بمعرفتكم $bm\epsilon rftkm$ ‘know you [plural] [M]’ and بمعرفتكن $bm\epsilon rftkn$ ‘know you [plural] [F]’ are second-person masculine and second-person feminine and only differ in terms of their enclitics, and therefore, would be labeled as B+2M and B+2F, respectively.

4.3.4 Corpus Overview and Statistics

Original Corpus After the annotation, 8.2% of the sentences (5,205) were eliminated due to malformed Arabic and annotation errors. This resulted in 58,035 sentences (423,254 words), constituting our *Original Corpus*. Table 4.4(a) includes the statistics

(a)						(b)					
Original Corpus						Balanced Corpus					
Sentences	Label	Rewrite Label				Input	Target 1M/2M	Target 1F/2M	Target 1M/2F	Target 1F/2F	Sentences
36,980 63.7%	B					B	B	B	B	B	36,980 46.0%
1,123 1.9%	1FB	B+1M				1FB	1MB	1FB	1MB	1FB	3063 3.8%
1,940 3.3%	1MB	B+1F				1MB	1MB	1FB	1MB	1FB	3063 3.8%
5,210 9.0%	B2F	B+2M				B2F	B2M	B2M	B2F	B2F	17374 21.6%
12,162 21.0%	B2M	B+2F				B2M	B2M	B2M	B2F	B2F	17374 21.6%
68 0.1%	1F2F	1M2F	1F2M	1M2M	1F2F	1M2M	1F2M	1M2F	1F2F	618 0.8%	
135 0.2%	1F2M	1M2M	1F2F	1M2F	1F2M	1M2M	1F2M	1M2F	1F2F	618 0.8%	
117 0.2%	1M2F	1F2F	1M2M	1F2M	1M2F	1M2M	1F2M	1M2F	1F2F	618 0.8%	
298 0.5%	1M2M	1F2M	1M2F	1F2F	1M2M	1M2M	1F2M	1M2F	1F2F	618 0.8%	
58,03											80,326

Table 4.4: Sentence-level statistics of the original corpus (a) and the balanced corpus (b) with its five versions.

about the *Original Corpus*. Out of all sentences, 36,980 (63.7%) are labeled as B. There are 17,374 (30%) sentences that include only second-person gendered references (BF and BM). This is five times more than sentences with only first-person gendered references (1FB and 1MB), which accounts for 5.3% (3,063 sentences) of all sentences. Moreover, the number of sentences including first or second person masculine references is more than the ones including feminine references (12,164 B2M vs 5,210 B2F, and 1,940 1MB vs 1,123 1FB). There are 618 (1.1%) sentences that have both first and second gendered references. All of the sentences that have first or second (or both) person gendered references are rewritten to introduce their opposite gender forms. This resulted in 21,055 manually added sentences (162,055 words). The word-level statistics of our *Original Corpus* are shown in Table 4.5(a). Among the newly added sentences, about 17% (27,596) of the words are gender-specific, constituting around 6.5% of all the words.

Balanced Corpus Similarly to Habash et al. (2019), to ensure equal gender representation in our dataset, we force balance the corpus by adding the manually rewritten sentences to the *Original Corpus* and using their original forms as their rewritten forms. This constitutes our *Balanced Corpus*. The sentence-level statistics of the *Balanced*

Corpus are presented in Table 4.4(b). This corpus has 80,326 sentences in total. Out of all sentences, 46% (36,980) are marked as B, whereas sentences with gendered references constituted 54% (43,346 sentences). We introduce five versions of the *Balanced Corpus*: **Input**, **Target 1M/2M**, **Target 1F/2M**, **Target 1M/2F**, and **Target 1F/2F**. Each of these target corpora is used to model the different target user contexts we are modeling for this task. The balanced Input corpus, includes all the sentences from the *Original Corpus* in addition to their rewritten forms. The Target 1M/2M corpus is the masculine-only corpus and it includes sentences that are either invariant/ambiguous or have a first or second person (or both) masculine references. Therefore, it only contains B, 1MB, B2M, and 1M2M sentences. The Target 1M/2F corpus is the masculine-feminine corpus and it contains sentences that are either invariant/ambiguous or have first person masculine references, second person feminine references, or first person masculine and second person feminine references (i.e., B, 1MB, B2F, and 1M2F sentences). The Target 1F/2M corpus is the feminine-masculine corpus and it contains B, 1FB, B2M, and 1F2M sentences. Finally, the Target 1F/2F corpus is the feminine-only corpus and it contains B, 1FB, B2F, and 1F2F sentences. All five corpora have the same number of sentences, words, and gendered-specific words. The word-level statistics of the *Balanced Corpus* are shown in Table 4.5(b).

Corpus Splits To aid reproducibility when using APGC v2.0 for various research experiments, we provide train, development, and test splits for all five balanced corpora. Following Habash et al. (2019), all five corpora were divided randomly as follows: training (Train: 70% or 57,603 sentences), development (Dev: 10% or 6,647 sentences) and testing (Test: 20% or 16,076 sentences). We made sure that the splits are balanced and all parallel versions of the sentences are in the same split.

(a)				(b)						
		Original Corpus		Balanced Corpus						
Words		Label	Rewrite Label	Input	Target 1M/2M	Target 1F/2M	Target 1M/2F	Target 1F/2F	Words	
395,658	93.5%	B+B		B+B	B+B	B+B	B+B	B+B	538,733	90.3%
21	0.0%	B+1F	B+1M	B+1F	B+1M	B+1F	B+1M	B+1F	43	0.0%
14	0.0%	B+1M	B+1F	B+1M	B+1M	B+1F	B+1M	B+1F	43	0.0%
157	0.0%	B+2F	B+2M	B+2F	B+2M	B+2M	B+2F	B+2F	1,419	0.2%
1,201	0.3%	B+2M	B+2F	B+2M	B+2M	B+2M	B+2F	B+2F	1,419	0.2%
1,488	0.4%	1F+B	1M+B	1F+B	1M+B	1F+B	1M+B	1F+B	4,870	0.8%
2,698	0.6%	1M+B	1F+B	1M+B	1M+B	1F+B	1M+B	1F+B	4,870	0.8%
6,685	1.6%	2F+B	2M+B	2F+B	2M+B	2M+B	2F+B	2F+B	22,655	3.8%
15,297	3.6%	2M+B	2F+B	2M+B	2M+B	2M+B	2F+B	2F+B	22,655	3.8%
3	0.0%	1F+1F	1M+1M	1F+1F	1M+1M	1F+1F	1M+1M	1F+1F	10	0.0%
3	0.0%	1M+1M	1F+1F	1M+1M	1M+1M	1F+1F	1M+1M	1F+1F	10	0.0%
0	0.0%	1F+2F	1M+2M 1F+2M 1M+2F	1F+2F	1M+2M	1F+2M	1M+2F	1F+2F	1	0.0%
1	0.0%	1M+2M	1F+2M 1M+2F 1F+2F	1M+2M	1M+2M	1F+2M	1M+2F	1F+2F	1	0.0%
0	0.0%	1F+2M	1M+2M 1M+2F 1F+2F	1F+2M	1M+2M	1F+2M	1M+2F	1F+2F	1	0.0%
0	0.0%	1M+2F	1M+2M 1F+2M 1F+2F	1M+2F	1M+2M	1F+2M	1M+2F	1F+2F	1	0.0%
0	0.0%	2F+1F	2M+1M 2M+1F 2F+1M	2F+1F	2M+1M	2M+1F	2F+1M	2F+1F	1	0.0%
0	0.0%	2F+1M	2M+1M 2M+1F 2F+1F	2F+1M	2M+1M	2M+1F	2F+1M	2F+1F	1	0.0%
0	0.0%	2M+1F	2M+1F 2F+1M 2F+1F	2M+1F	2M+1M	2M+1F	2F+1M	2F+1F	1	0.0%
1	0.0%	2M+1M	2M+1F 2F+1M 2F+1F	2M+1M	2M+1M	2M+1F	2F+1M	2F+1F	1	0.0%
4	0.0%	2F+2F	2M+2M	2F+2F	2M+2M	2M+2M	2F+2F	2F+2F	32	0.0%
23	0.0%	2M+2M	2F+2F	2M+2M	2M+2M	2M+2M	2F+2F	2F+2F	32	0.0%
423,254									596,799	

Table 4.5: Word-level statistics of the original corpus (a) and the balanced corpus (b) with its five versions.

Machine Translation Outputs The efforts to develop APGC v1.0 and APGC v2.0 were motivated by the observation of common gender bias in user-unaware NLP systems targeting morphologically rich languages, specifically Arabic in our case. As part of our dataset, we generated machine translation outputs by translating the English portion of the *Input Balanced Corpus* into Arabic using the Google Translate API. We selected Google Translate due to its widespread use, though our approach can be applied to any machine translation system. While Google Translate has made notable efforts to mitigate gender bias—such as generating multiple gendered translations for certain language pairs (Johnson, 2020)—Arabic is not yet among them. To support research on bias mitigation and corrective post-editing, we include Google Translate’s outputs in our corpus release.

4.4 Approach

In this section, we present the gender rewriting models explored in this dissertation. We experiment with two main approaches: joint Seq2Seq models, which perform sentence-level rewriting in a single pass without an explicit identification step (Alhafni et al., 2020), and multi-step word-level models, which decompose the process into separate identification and rewriting stages for finer control (Alhafni et al., 2022b). Below, we describe both models and evaluate their performance on APGC v2.0.

4.4.1 Joint Gender Rewriting

Our joint gender rewriting model is a character-level Seq2Seq model. The encoder consists of a two-layer bidirectional GRU (Cho et al., 2014), while the decoder is a two-layer GRU with additive attention (Bahdanau et al., 2014) over the encoder’s hidden states. Unlike the multi-step approach, this model rewrites sentences without an explicit word-level gender identification step, directly generating the target-gendered output in a single pass. To incorporate the user target gender, we employ side constraints (Sennrich et al., 2016a). Specifically, we prepend a special token representing the target gender to the input sentence (e.g., ‘<1M/2F> Input Sentence’). This token is treated like any other in the vocabulary, allowing the encoder to learn a representation for it, which the decoder then attends to when generating the output sequence.

Additionally, we explore enriching character representations with word-level morphological features. We extract functional gender features from the CALIMA_{Star} Arabic morphological analyzer (Taji et al., 2018b), which is part of CAMEL Tools (Obeid et al., 2020). These features indicate whether a word is masculine or feminine and whether its analysis includes spelling back-off, and they are represented as a four-dimensional one-

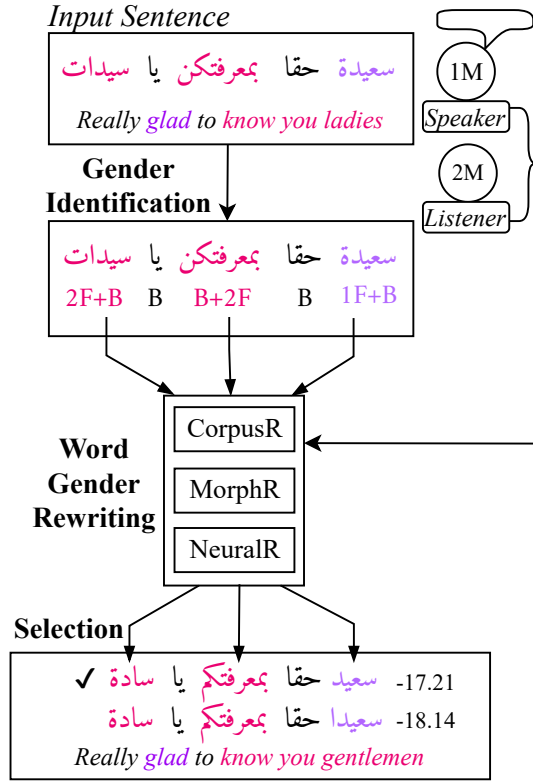


Figure 4.1: The multi-step gender rewriting system. First person gendered words are in purple and second person gendered words are in red. The user target gender is 1M/2M. The input words *glad* (1F+B), *know you* (B+2F), and *ladies* (2F+B) are rewritten to their masculine forms.

hot vector. We append these morphological features to character-level representations, enriching each character embedding with word-level information before feeding it into the encoder. At inference time, we use beam search to generate the output sequence.

4.4.2 Multi-Step Gender Rewriting

Seq2Seq models, while effective for many text generation tasks, often suffer from hallucinations and lack fine-grained control over the output (Ji et al., 2023). They also require large amounts of training data to generalize well, which can be a challenge in specialized tasks like gender rewriting. To address these limitations, we explore a

controlled word-level multi-step approach that combines the strengths of rule-based and neural models. Our system consists of three components: *Gender Identification*, *Out-of-Context Word Gender Rewriting*, and *In-Context Ranking and Selection*. Figure 4.1 presents an overview of our mutli-step gender rewriting model.

Gender Identification: We first identify the word-level gender label (base word + pronominal enclitic) for each word in the input sentence. We build a word-level classifier by leveraging a Transformer-based pretrained language model. There are many Arabic monolingual BERT models available such as AraBERT (Antoun et al., 2020), ARBERT (Abdul-Mageed et al., 2021a), and QARIB (Abdelali et al., 2021). However, we chose to use CAMELBERT MSA (Inoue et al., 2021) as it was pretrained on the largest MSA dataset to date. Following the work of Devlin et al. (2019), we fine-tune CAMELBERT MSA using Hugging Face’s transformers (Wolf et al., 2020) by adding a fully-connected linear layer with a softmax on top of its architecture.

Out-of-context Word Gender Rewriting: Given the desired *user target gender* as an input and the identified gender label for each word in the input sentence, we decide if a word-level gender rewrite is needed based on the compatibility between the provided user target gender and the predicted word-level gender labels. We implement three word-level gender alternative generation models: *Corpus-based Rewriter*, *Morphological Rewriter*, and *Neural Rewriter*:

- **Corpus-based Rewriter (CorpusR):** We build a simple word-level lookup rewriting model by exploiting the fully aligned words in the APGC. We implement this model as a bigram maximum likelihood estimator: given an input word with its bigram surrounding context (w_i, w_{i-1}) , a gender alternative target word (y_i) , and

a desired word-level target gender (g), the CorpusR model is built by computing $P(y_i|w_i, w_{i-1}, g)$ over the training examples. During inference, we generate all possible gender alternatives for the given input word (w_i). If the bigram context (w_i, w_{i-1}) was not observed in the training data, we backoff to a unigram context. If the input word was not observed during training, we pass it to the output as it is.

- **Morphological Rewriter (MorphR):** For the morphological rewriter, we use the morphological analyzer and generator provided by CAMEL Tools (Obeid et al., 2020). We extend the Standard Arabic Morphological Analyzer database (SAMA) (Graff et al., 2009) used by the morphological generator to produce controlled gender alternatives. We make our extensions to the database publicly available. Given an input word and a desired word-level target gender, the morphological generator has the ability to produce gender alternatives by either rewriting the base word, its pronominal enclitics, or both. If an input word does not get recognized by the morphological analyzer and generator, we pass it to the output as it is. It is worth noting that this rewriting model does not require any training data.
- **Neural Rewriter (NeuralR):** The word-level neural rewriter shares the same character-level encoder-decoder architecture as the joint model described in (§4.4.1), utilizing a two-layer bidirectional GRU encoder and a two-layer GRU decoder with additive attention. To incorporate word-level target gender information, we use the same side constraint technique (Sennrich et al., 2016a) applied at the word level. Specifically, we prepend a special token representing the target gender (e.g., [1F+B]) to the input word and we feed that entire sequence to the model (i.e., [1F+B]سعيد). During inference, we use beam search to generate the top 3-best hypotheses.

In-Context Ranking and Selection: Since the three word-level gender alternative generation models we implement are out-of-context and given Arabic’s morphological richness, we expect to get multiple output words when generating a single gender alternative for a particular input word. This leads to producing multiple candidate gender alternative output sentences. To select the best candidate output sentence, we rank all candidates in full sentential context based on their *pseudo-log-likelihood* (PLL) scores (Salazar et al., 2020). We first use Hugging Face’s transformers to fine-tune the CAMeLBERT MSA model on the Input corpus of APGC by using a masked language modeling (Devlin et al., 2019) objective. This helps in mitigating the domain shift (Gretton et al., 2006) issue between CAMeLBERT’s pretraining data and APGC. We then compute the PLL score for each sentence using the fine-tuned CAMeLBERT MSA model by masking the sentence tokens one by one.

4.5 Experimental Setup

Evaluation Metrics We treat the gender rewriting problem as a grammatical error correction task and use the MaxMatch (M^2) Scorer (Dahlmeier and Ng, 2012) as our evaluation metric. The M^2 Scorer computes the precision (P), recall (R), and $F_{0.5}$ by maximally matching system edits with gold-standard edits. $F_{0.5}$ weighs precision twice as much as recall, to prioritize the accuracy of edits relative to all edits made by the system. The gold edits are computed by the M^2 Scorer based on provided gold references. We also report BLEU (Papineni et al., 2002) scores using SacreBLEU (Post, 2018). We report results in a normalized space for Alif, Ya, and Ta-Marbuta (Habash, 2010).

Baselines We introduce three baseline models. The first trivially copies input sentences to the output, highlighting the similarity between inputs and outputs. The second and third

use the sentence-level Seq2Seq joint rewriting model (§4.4.1). To assess the impact of word-level morphological features, we evaluate two variants: one without morphological features (**Joint**) and another incorporating them (**Joint+Morph**).

LLMs We evaluate four LLMs: two commercial models and two open-source, Arabic-centric models. The commercial models include OpenAI’s GPT-3.5-turbo and GPT-4o (OpenAI et al., 2024), while the Arabic-centric models are Jais-30B-Chat (Sengupta et al., 2023) and the recently introduced Fanar LLM (Team et al., 2025). We prompt GPT-3.5-turbo, GPT-4o, and Fanar through the OpenAI API, while Jais-30B-Chat is prompted using Hugging Face’s Transformers (Wolf et al., 2020). Our experiments use both English and Arabic prompts, employing 0-shot and 5-shot prompting strategies. Additionally, we experiment with incorporating gender identification (**GID**) predictions directly into the prompt, explicitly indicating gender-marking words. This strategy aims to enhance performance by guiding the model to change only the gender-marking words while preserving the original phrasing and lexical choices. Our prompt designs are detailed in Tables A.1, A.2, and A.3 in Appendix A.1.

Multi-Step Models We explore five variants of the multi-step gender rewriting model from §4.4.2, originally introduced by (Alhafni et al., 2022b). All five variants use the same gender identification (**GID**) and in-context selection models, but differ in their out-of-context word-level gender rewriting generation setup. The first three variants use one word-level gender rewriting model each – **CorpusR**, **MorphR**, or **NeuralR**. The fourth multi-step model uses **MorphR** as a backoff if the input words that need to be rewritten are not observed by the **CorpusR** model during training (**CorpusR»MorphR**). The fifth system uses all three word-level gender alternative generation models in a backoff cascade: **CorpusR»MorphR»NeuralR**.

Data Augmentation Given the relatively small size of APGC and motivated by work on using data augmentation to improve grammatical error correction (Wan et al., 2020; Stahlberg and Kumar, 2021), we investigate adding additional training examples through data augmentation. We randomly selected 800K sentences from the English-Arabic portion of the OpenSubtitles 2018 dataset, which was used to build APGC. We ensured that all extracted pairs include either first or second (or both) person pronouns on the English side: *I, me, my, mine, myself*, and *you, your, yours, yourself*. To generate gender alternatives of the selected Arabic sentences, we pass each sentence four times through our best gender rewriting system to generate all four user gender contexts (1M/2M, 1F/2M, 1M/2F, 1F/2F). We add the 800K selected Arabic sentences and their 1M/2M, 1F/2M, 1M/2F, 1F/2F generated gender alternatives to the Input, Target 1M/2M, Target 1F/2M, Target 1M/2F, and Target 1F/2F corpora of the training split of APGC, respectively. At the end, we end up with 857,603 Arabic parallel sentences (6,209,958 words).

4.5.1 Results

Table 4.6 presents the Dev set results. **Joint+Morph** improves over the **Joint** baseline with 1.9 increase in $M^2 F_{0.5}$, highlighting the usefulness of the morphological features.

We present the LLM results (Table 4.6(d–h)) using their best configurations, determined by prompt language and strategy (0-shot vs. 5-shot). Full results are in Table A.4 in Appendix A.2. Among the LLMs, GPT-4o performs best with an $F_{0.5}$ score of 52.8. However, all LLMs underperform compared to the joint models, mainly due to over-generation, making unnecessary edits beyond the intended gender-marking words, which is reflected in the low BLEU scores. To address this, we conducted an additional experiment where gender-marking words were explicitly identified using GID predictions before prompting GPT-4o. This setup, **GID+GPT-4o**, significantly improved perfor-

	P	R	F_{0.5}	BLEU
(a) Do Nothing	100.0	0.0	0.0	89.4
(b) Joint	77.1	77.7	77.2	95.6
(c) Joint + Morph	79.0	79.8	79.1	96.2
(d) Fanar	12.6	43.7	14.7	39.5
(e) Jais-30B-Chat	8.2	33.4	9.6	24.5
(f) GPT-3.5-turbo	21.9	64.8	25.2	69.5
(g) GPT-4o	49.2	74.6	52.8	88.8
(h) GID + GPT-4o	77.1	77.7	77.2	96.3
(i) GID + CorpusR + Selection	88.2	71.2	84.2	96.5
(j) GID + MorphR + Selection	84.5	75.3	82.5	97.0
(k) GID + NeuralR + Selection	84.6	73.3	82.1	96.8
(l) GID + CorpusR » MorphR + Selection	88.6	85.8	88.0	98.0
(m) GID + CorpusR » MorphR » NeuralR + Selection	88.5	86.7	88.1	98.0
(n) GID_{Aug} + CorpusR » MorphR » NeuralR_{Aug} + Selection	88.7	86.8	88.3	98.1

Table 4.6: Multi-user gender rewriting results on the Dev set of APGC v2.0. **Aug** indicates using augmented data.

mance, raising the $F_{0.5}$ score by 24.4 points to 77.2. These results highlight the utility of GID as a control mechanism to better steer LLM outputs.

When it comes to the multi-step rewriting models (Table 4.6(i-n)), The best performing system overall is the model using all rewrite components (Table 4.6(m)), henceforth, *Our Best Model*. It improves over the joint models and LLMs in every compared metric. *Our Best Model*’s biggest advantages seem to come from combining the three word-level out-of-context gender alternative generation models in a cascaded setup to deal with OOV words during the generation. Comparing (m) with (c,i,j) in Table 4.6, we observe improvements ranging from 3.91 to 6.02 $F_{0.5}$.

We used *Our Best Model* to conduct the data augmentation experiments. The best augmented model’s results, which benefits from augmentation in the **GID** and **NeuralR** components, are also presented in Table 4.6(n). However, its increase of 0.19 points in $F_{0.5}$ is not statistically significant with McNemar’s (McNemar, 1947) test at $p > 0.05$.

The results of our best models on the Test sets of APGC v2.0 are presented in

		P	R	F_{0.5}	BLEU
Joint	+ Morph	79.3	80.4	79.5	96.2
GID	+ GPT-4o	76.4	76.5	76.4	96.2
GID	+ CorpusR » MorphR » NeuralR + Selection	88.7	86.1	88.2	98.0
GID_{Aug}	+ CorpusR » MorphR » NeuralR_{Aug} + Selection	88.9	86.7	88.4	98.1

Table 4.7: Gender rewriting results on the Test sets of APGC v2.0.

Table 4.7. The results on APGC v2.0 Test show consistent conclusions with the Dev results. Our best augmented model improves over its non-augmented variant in every compared metric, including a 0.25 absolute increase in $F_{0.5}$ that is statistically significant with McNemar’s test at $p < 0.05$.

4.5.2 Error Analysis

We conducted an error analysis over the output of our best augmented system on APGC v2.0 Dev. In total, there were 1,475 (5.5% out of 26,588) sentences with errors across the four target corpora. Table 4.8 presents a summary of the error types our best augmented model makes. The majority of errors (67.3%) were caused by **GID** which achieves a word-level accuracy of 98.9% on Dev. The gender-rewriting errors constituted 18.1% and selection errors 14.6%. Considering different target corpora, we observe that every time an F target is added, the number of errors increases. The 1M/2M target outputs has the lowest number of errors (268 or 18%), while the 1M/2F targets outputs has the highest number of errors (480 or 33%).

4.5.3 Use Case: Post-Editing MT Output

We demonstrate next how our proposed gender rewriting model could be used to personalize the output of user-unaware MT systems through post-editing. We use the English

	1M/2M	1F/2M	1M/2F	1F/2F
GID	150 56%	194 70%	325 68%	324 72%
Rewrite	69 26%	50 18%	82 17%	66 15%
Select	49 18%	35 13%	73 15%	58 13%
<i>Total</i>	268	279	480	448

Table 4.8: Error type statistics of our best augmented system’s performance on APGC v2.0 Dev.

Target	1M/2M	1F/2M	1M/2F	1F/2F
Google Translate	13.6	13.2	11.4	11.0
Best System_{Aug}	13.7	13.6	13.3	13.2

Table 4.9: BLEU results on the post-edited Google Translate output of APGC v2.1 Test using our best augmented system.

to Arabic Google Translate output sentences that are part of APGC v2.0. We evaluate Google Translate’s output against all four target corpora (1M/2M, 1F/2M, 1M/2F, 1F/2F) separately. To re-target Google Translate’s Arabic output for the four user gender contexts we model, we pass each Arabic sentence four times through our best augmented system (Table 4.6(n)). We present the evaluation in terms of BLEU in Table 4.9 over APGC v2.0 Test. All the results are reported in a normalized space for Alif, Ya, and Ta-Marbuta.

Again, we observe that every time an M participant is switched to F, the BLEU scores drop for Google Translate’s output. This highlights the bias the machine translation output has towards masculine grammatical gender preferences. The post-edited outputs generated by our best augmented system improves over Google Translate’s across the four target user contexts, achieving the highest increase in 2.27 BLEU points for 1F/2F.

4.6 The User-Aware Arabic Gender Rewriter

We develop a web-based application introduced in [Alhafni et al. \(2023b\)](#) that leverages our best-performing multi-step gender rewriting model, enabling users to interact with a fully functional Arabic gender rewriting system. The system accepts both Arabic and English input sentences, allowing users to specify their desired first- and/or second-person grammatical gender preferences. For Arabic input, it generates gender-rewritten alternatives that align with the specified user preferences. For English input, it first translates the text into Arabic using Google Translate before applying gender rewriting. To the best of our knowledge, this is the first open-access web-based system for Arabic gender rewriting, providing a practical tool for generating personalized, user-aware outputs.

Figure 4.2 illustrates the web-based system. At the top, there is a text box to input either English or Arabic text. At each side of the text box, there are two selection buttons to indicate the desired target gender preferences for the speaker and the listener (σ is for masculine and φ is for feminine). Users can choose any combination of target genders, including no target gender selection (i.e., requesting no rewriting). Once the user clicks on the *Translate & Rewrite* button, any English input is first translated into Arabic using the Google Translate API before generating gendered alternatives. When the gender rewriting process is done, additional text boxes will appear: the first text box will always contain the gender-identified Arabic inputs and the rest of the text boxes will contain the gender rewritten alternatives. Each gender marking word in the gender-identified input text box will be labeled as either masculine (σ) or feminine (φ). First-person (i.e., speaker) gendered words are colored in **blue** and second-person (i.e., listener) gendered words are colored in **orange**.



Figure 4.2: The Arabic Gender Rewriter interface showing gender rewritten alternatives of three input sentences in four modes: (a) Target speaker ♀ gender rewrites, (b) Target speaker ♀ and target listener ♀ and ♂ gender rewrites, (c) Target speaker ♀ and ♂ and target listener ♀ gender rewrites, and (d) Target speaker ♀ and ♂ and target listener ♀ and ♂ gender rewrites. Speaker gendered words are in blue and listener gendered words are in orange.

The number of output text boxes corresponds to the selected target gender preferences. Each box is labeled according to the gender combination it represents. For instance, in Figure 4.2(a), two text boxes display first-person masculine and feminine alternatives, while Figure 4.2(b) shows four text boxes containing gendered alternatives for both first- and second-person references.

4.7 The Shared Task on Arabic Gender Rewriting

To raise awareness of gender bias in Arabic NLP and encourage the development of mitigation strategies, we organized a shared task on gender rewriting for Arabic as part of the Seventh Arabic Natural Language Processing Workshop (WANLP), collocated with EMNLP 2022. This was the first WANLP shared task in seven years to focus on Arabic language generation. A total of five teams from four countries participated.

4.7.1 Data

Participants were only allowed to use the publicly available APGC v2.0 to build their systems. To ensure a fair comparison between all participants, we manually annotated a new blind test set to evaluate their systems. The new blind test set was selected and annotated by following the same guidelines used to build the APGC (§4.3). This corpus has 7,318 sentences in total. Out of all sentences, 38.5% (2,818) are marked as B, whereas sentences with gendered references constituted 61.5% (4,500 sentences).

4.7.2 Participants and Systems

Table 4.10 presents the names of the participating teams and their affiliations. All participants leveraged pretrained language models such as AraBERT (Antoun et al., 2020), CAMeLBERT (Inoue et al., 2021), T5 (Raffel et al., 2020), and AraT5 (Nagoudi et al., 2022), when developing their systems. Some systems consisted of multiple components to do gender identification and then rewriting as we did in §4.4.2, while others treated the problem as a traditional Seq2Seq task. Table 4.11 presents a summary of the different approaches used to develop the different systems.

Team	Affiliation
Cairo Team	Microsoft Egypt, Egypt
CasaNLP	Archipel Cognitive; and Leyton, Morocco
Distinguishers	Taif University; and Umm Alqura University, KSA
Qaddoumi	New York University, USA
UDEL-NLP	University of Delaware, USA

Table 4.10: List of the five teams who participated in the gender rewriting shared task.

Team	Gender ID	Special Preprocessing	Pretrained Models
Cairo Team	✓		CAMeLBERT-MSA + AraT5-MSA
CasaNLP	✓	Word Side Constraints	CAMeLBERT-MSA + AraT5-MSA
Distinguishers	✓	Morphological Features	CAMeLBERT-MSA + AraBERT
Qaddoumi		Romanization	T5
UDEL-NLP		Sentence Side Constraints	ArabicT5

Table 4.11: Approaches and techniques used by the participants. Gender ID refers to gender identification. Special Preprocessing refers to any form of preprocessing done to modify the data (e.g., adding side-constraints, morphological processing, transliteration, etc.). Pretrained Models indicates the usage of pretrained models as part of the system.

4.7.3 Results

Table 4.12 presents the results on the newly annotated Blind Test set. The last row is for the state-of-the-art system in §4.4.2. The best result in terms of $F_{0.5}$ is achieved by the Cairo Team (75.4), the official winner of the shared task. This is mainly due to their high score in precision (76.3). Qaddoumi comes in second place achieving an $F_{0.5}$ of 59.7, followed by UDEL-NLP in third place with 59.1 in $F_{0.5}$. In fourth place, CasaNLP achieves an $F_{0.5}$ score of 55.45 with the highest recall of 84.6. Distinguishers comes in fifth place, achieving 20.5 in $F_{0.5}$. It is worth noting that none of the systems is able to beat the previously published system by our system applied to the new Blind Test.

Team	P	R	F _{0.5}	BLEU
Cairo Team	76.3 (1)	72.3 (3)	75.4 (1)	94.9 (1)
CasaNLP	51.1 (4)	84.6 (1)	55.5 (4)	86.1 (4)
Distinguishers	20.9 (5)	19.0 (5)	20.5 (5)	84.9 (5)
Qaddoumi	56.5 (3)	77.1 (2)	59.7 (2)	88.5 (3)
UDEL-NLP	57.1 (2)	68.6 (4)	59.1 (3)	91.0 (2)
Alhafni et al. (2022b)	88.5	85.0	87.8	97.6

Table 4.12: Results on the Blind Test set. Numbers in parentheses are the ranks.

(a)		(b)	
Team	Word Δ	Metric	Correl
Cairo Team	0.80%	P	-42.95%
CasaNLP	-0.02%	R	-77.56%
Distinguishers	1.28%	F_{0.5}	-50.86%
Qaddoumi	-0.63%	BLEU	-11.86%
UDEL-NLP	0.05%		

Table 4.13: (a) The relative difference in the number of generated words for each team in comparison with the Blind Test reference. (b) The Pearson correlation of the shared task metrics in Table 4.12 with the *absolute* values of Word Δ .

4.7.4 Error Analysis

We conducted a simple error analysis over the outputs of all system on the Blind Test set. Given that most teams employed sentence-level Seq2Seq models when developing their gender rewriting systems, we suspected that the outputs will be noisy since sentence-level models will not guarantee that changes are only applied to gendered words, or maintain the word-level parallelism between the input and output. Table 4.13(a) presents the relative difference in the number of generated words for each team in comparison with the Blind Test reference; and Table 4.13(b) presents their correlation with the shared task metrics. None of the teams maintained the total number of words. We observe a strong negative correlation between the absolute value of relative word count differences and the evaluation metrics – almost -51% correlation with F_{0.5}, and -78% correlation with recall.

4.8 Summary

In this chapter, we introduced the task of gender rewriting for Arabic. We presented the Arabic Parallel Gender Corpus and explored different gender rewriting approaches, including LLMs, a joint sentence-level Seq2Seq model, and a word-level multi-step approach. Our experiments demonstrated that the word-level multi-step approach outperforms the Seq2Seq model by providing finer control over the generation process. Additionally, we showcased how this system can help mitigate gender bias in English-to-Arabic machine translation. To facilitate user interaction, we developed a web-based application that seamlessly integrates the model. Lastly, we discussed our findings from a shared task on Arabic gender rewriting that we organized.

Our primary motivation behind this work is to enhance the inclusiveness of NLP applications for morphologically rich, gender-marked languages. Our work aims to empower users by enabling them to interact with NLP systems in ways that align with their social identities. However, we acknowledge that our approach—limited to grammatical gender in Arabic—excludes other alternatives, such as non-binary or gender-neutral expressions. Currently, we are unaware of any published sociolinguistic research exploring such alternatives in Arabic. Nevertheless, we emphasize the importance of adapting Arabic NLP models to accommodate emerging gender expressions as language usage evolves. Looking ahead, we envision a future where websites and translation systems integrate automatic gender rewriting, allowing users to customize gender presentation through intuitive settings, much like selecting a preferred language.

Chapter 5

Arabic Grammatical Error Detection and Correction

In this chapter, we present a comprehensive study on Modern Standard Arabic (MSA) grammatical error correction (GEC). We report the first results on MSA GEC using Transformer-based pretrained Seq2Seq models and introduce the task of multi-class MSA grammatical error detection (GED). We show that conditioning Seq2Seq models on error patterns by incorporating GED predictions as auxiliary input significantly improves GEC performance. Beyond model architectures, we investigate the impact of contextual morphological preprocessing on Arabic GEC. Additionally, we benchmark open-source and commercial LLMs to assess their performance on MSA GEC. Our models achieve state-of-the-art results on two MSA GEC shared task datasets: one consisting of comments written by native speakers (L1) and the other of essays written by second-language learners (L2). Additionally, we establish a strong benchmark on a recently introduced MSA GEC dataset consisting of L1 essays.

5.1 Introduction

Grammatical Error Correction (GEC) aims to correct errors in text, including grammatical mistakes such as missing prepositions and subject-verb agreement mismatches, as well as orthographic and semantic errors like misspellings and incorrect word choices. Most state-of-the-art systems adopt neural machine translation techniques to transform erroneous text into its corrected form. In contrast, grammatical error detection (GED) is framed as a sequence labeling task that identifies and classifies errors. Both GEC and GED have significant pedagogical applications for native (L1) and second-language (L2) learners.

While GEC and GED have been widely studied in English, research on morphologically rich languages remains limited due to the scarcity of annotated datasets with standardized error types. In Arabic, the application of Seq2Seq modeling for GEC is still underexplored, and multi-class Arabic GED has yet to be investigated.

In this chapter, we focus on Modern Standard Arabic (MSA). We benchmark pre-trained Arabic Seq2Seq models and LLMs on GEC and formalize the task of multi-class MSA GED by enriching existing parallel GEC datasets with error type annotations. We also show that conditioning models on GED predictions improves GEC performance. Formally, given an erroneous Arabic sentence X and its corresponding error type sequence E , the task is to generate the corrected version Y :

$$P(Y|X, E) = \prod_{t=1}^n P(y_t|y_1, \dots, y_{t-1}, X, E)$$

Additionally, we explore the impact of contextual morphological preprocessing on GEC performance. Our models achieve state-of-the-art results on two MSA GEC (L1 and L2) datasets and establish a strong benchmark on a recently created L1 MSA GEC dataset.

5.2 Background and Related Work

GEC Approaches

Early GEC efforts focused on building feature-based machine learning (ML) classifiers to fix common error types (Chodorow et al., 2007; Tetreault and Chodorow, 2008; Dahlmeier and Ng, 2011; Kochmar et al., 2012; Rozovskaya and Roth, 2013; Farra et al., 2014). Such models required feature engineering and lacked the ability to correct all error types simultaneously. Reformulating GEC as a monolingual machine translation task alleviated these issues, first with statistical machine translation approaches (Felice et al., 2014; Junczys-Dowmunt and Grundkiewicz, 2014, 2016) and then neural machine translation approaches (Yuan and Briscoe, 2016; Xie et al., 2016; Watson et al., 2018b), with Transformer-based models being the most dominant (Junczys-Dowmunt et al., 2018; Yuan et al., 2019; Zhao et al., 2019; Grundkiewicz et al., 2019; Katsumata and Komachi, 2020; Kaneko et al., 2020; Wan et al., 2020; Yuan et al., 2021; Yuan and Bryant, 2021; Stahlberg and Kumar, 2021; Rothe et al., 2021; Zhou et al., 2023a; Luhtaru et al., 2024).

To improve efficiency and interpretability, text editing models have emerged as an alternative to Seq2Seq approaches (Awasthi et al., 2019; Malmi et al., 2019; Stahlberg and Kumar, 2020; Mallinson et al., 2020; Omelianchuk et al., 2020; Straka et al., 2021; Mallinson et al., 2022; Tarnavskiy et al., 2022; Mesham et al., 2023; Zhang et al., 2023). Unlike Seq2Seq models, which generate corrected text from scratch, text editing models treat GEC as a sequence tagging task, producing a set of edit operations that modify the erroneous input. We present a novel text editing model for GEC in chapter 7.

LLMs have also been evaluated on GEC (Fang et al., 2023; Coyne et al., 2023; Wu et al., 2023; Loem et al., 2023; Raheja et al., 2023; Kaneko and Okazaki, 2023; Raheja

et al., 2024; Davis et al., 2024; Katinskaia and Yangarber, 2024; Omelianchuk et al., 2024; Mita et al., 2024; Kaneko and Okazaki, 2024). In our work, we benchmark commercial and open-source LLMs on MSA GEC.

GED Approaches

When it comes to GED, [Rei and Yannakoudakis \(2016\)](#) presented the first GED results using a neural approach framing GED as a binary (correct/incorrect) sequence tagging problem. Others used pretrained language models such as BERT ([Devlin et al., 2019](#)), ELECTRA ([Clark et al., 2020](#)), and XLNet ([Yang et al., 2019a](#)) to improve binary GED ([Bell et al., 2019](#); [Kaneko and Komachi, 2019](#); [Yuan et al., 2021](#); [Rothe et al., 2021](#)). [Zhao et al. \(2019\)](#) and [Yuan et al. \(2019\)](#) demonstrated that combining GED and GEC yields improved results: they used multi-task learning to add token-level and sentence-level GED as auxiliary tasks when training for GEC. Similarly, [Yuan et al. \(2021\)](#) showed that binary and multi-class GED improves GEC.

Arabic GED and GEC

Although GEC has been studied in other languages ([Bryant et al., 2023](#)) such as Chinese ([Zhao et al., 2018c](#); [Rao et al., 2020](#)), Czech ([Náplava and Straka, 2019](#); [Náplava et al., 2022](#)), German ([Boyd, 2018](#)), Japanese ([Koyama et al., 2020](#)), Russian ([Rozovskaya and Roth, 2019](#)), and Ukrainian ([Syvokon et al., 2023](#)), most research efforts have mainly focused on English and gained popularity through a series of shared tasks ([Dale and Kilgarriff, 2011](#); [Ng et al., 2013, 2014](#); [Bryant et al., 2019](#)). When it comes to Arabic, GEC research gained traction due to the QALB-2014 (L1) ([Mohit et al., 2014b](#)) and QALB-2015 (L1 and L2) ([Rozovskaya et al., 2015a](#)) shared tasks that were organized as part of the Qatar Arabic Language Bank (QALB) project ([Zaghoulani et al., 2014](#),

2015a). More recently, [Habash and Palfreyman \(2022\)](#) introduced the Zayed Arabic-English Bilingual Undergraduate Corpus (ZAEBUC) corpus, a dataset of essays written by L1 university students. In this work, we leverage the QALB-2014, QALB-2015, and ZAEBUC.

Arabic GEC modeling efforts ranged from feature-based ML classifiers to statistical MT models ([Rozovskaya et al., 2014](#); [Attia et al., 2014](#); [Bougares and Bouamor, 2015](#); [Nawar, 2015](#)). [Watson et al. \(2018b\)](#) introduced the first character-level Seq2Seq model and achieved state-of-the-art (SOTA) results on the L1 Arabic GEC data used in the QALB-2014 and 2015 shared tasks. Recently, vanilla Transformers were explored for synthetic data generation to improve L1 Arabic GEC and were tested on the L1 data of the QALB-2014 and 2015 shared tasks ([Solyman et al., 2021, 2022, 2023](#)). To our knowledge, the last reported QALB-2015 L2 results appeared in the original shared task.

A number of researchers reported on Arabic binary GED. [Habash and Roth \(2011\)](#) used feature-engineered SVM classifiers to detect Arabic handwriting recognition errors. [Alkhatib et al. \(2020\)](#) and [Madi and Al-Khalifa \(2020\)](#) used LSTM-based classifiers. None of them used any of the publicly available GEC datasets mentioned above to train and test their systems. In our work, we explore multi-class GED by obtaining error type annotations from ARETA ([Belkebir and Habash, 2021](#)), an automatic error type annotation tool for MSA. To our knowledge, we are the first to report on Arabic multi-class GED.

Arabic GEC Challenges

While its **orthography** is standardized, written MSA suffers many orthographic inconsistencies (§2) even in professionally written news articles ([Buckwalter, 2004b](#); [Habash](#)

et al., 2012a). For example, hamzated Alifs (أ, إ) are commonly confused with the un-hamzated letter (ا), and the word-final letters ي and ى are often used interchangeably. These errors affect 11% of all words (4.5 errors per sentence) in the Penn Arabic Treebank (Habash, 2010). Additionally, the use of punctuation in Arabic is very inconsistent, and omitting punctuation marks is very frequent (Awad, 2013; Zaghoulani and Awad, 2016). Punctuation errors account for approximately 40% of all errors in the QALB-2014 GEC shared task—ten times higher than those found in the English data used in the CoNLL-2013 GEC shared task (Ng et al., 2013).

Beyond orthography, Arabic’s rich **morphology** presents additional challenges for GEC. The language inflects for gender, number, person, case, state, mood, voice, and aspect, while also incorporating cliticized particles and pronouns. These factors significantly expand vocabulary size and introduce structural complexity. Moreover, due to **diglossia**, native speakers writing in MSA frequently code-switch by incorporating elements from their dialects (§2).

5.3 Approach

5.3.1 Arabic Grammatical Error Detection

Most of the work on GED has focused on English (§5.2), where error type annotations are provided manually (Yannakoudakis et al., 2011; Dahlmeier et al., 2013) or obtained automatically using an error type annotation tool such as ERRANT (Bryant et al., 2017). However, when it comes to morphologically rich languages such as Arabic, GED remains a challenge. This is largely due to the lack of manually annotated data and standardized error type frameworks. In this work, we treat GED as a multi-class sequence labeling task.

We present a method to automatically obtain error type annotations by extracting edits from parallel erroneous and corrected sentences and then passing them to an Arabic error type annotation tool. To the best of our knowledge, this is the first work that explores multi-class GED in Arabic.

Edit Extraction

Before automatically labeling each erroneous sentence token, we need to align the erroneous and corrected sentence pairs to locate the positions of all edits so as to map errors to corrections. This step is usually referred to as *edit extraction* in GEC literature.

We first obtain character-level alignment between the erroneous and corrected sentence pair by computing the weighted Levenshtein edit distance (Levenshtein, 1966) for each pair of tokens in the two sentences. The output of this alignment is a sequence of token-level edit operations representing the minimum number of insertions, deletions, and replacements needed to transform one token into another. Each of these operations involves one token at most belonging to either sentence. However, some errors may involve more than one single edit operation. To capture multi-token edits, we extend the alignment to cover merges and splits by implementing an iterative algorithm that greedily merges or splits adjacent tokens such that the overall cumulative edit distance is minimized.

Error Type Annotation

Next, we pass the extracted edits to an automatic annotation tool to label them with specific error types. We use ARETA, an automatic error type annotation tool for MSA (Belkebir and Habash, 2021). Internally, ARETA is built using a combination of rule-based components and an Arabic morphological analyzer (Taji et al., 2018a; Obeid et al.,

	QALB-2014			QALB-2015		
	P \uparrow	R \uparrow	AER \downarrow	P \uparrow	R \uparrow	AER \downarrow
M²	92.5	87.1	0.10	90.8	83.3	0.13
Lev.	86.8	84.3	0.14	84.5	84.2	0.16
ARETA	84.3	82.9	0.16	84.1	84.7	0.16
Ours	99.6	99.7	0.00	97.7	98.0	0.02

Table 5.1: Evaluation of different alignment algorithms.

2020). It uses the error taxonomy of the Arabic Learner Corpus (ALC) (Alfaifi and Atwell, 2012; Alfaifi et al., 2013) which defines seven error classes covering orthography (O), morphology (M), syntax (X), semantics (S), punctuation (P), merges, and splits. The error classes are further differentiated into 32 error tags that can be assigned individually or in combination.

ARETA comes with its own alignment algorithm that extracts edits, however, it does not handle many-to-one and many-to-many edit operations (Belkebir and Habash, 2021). We replace ARETA’s alignment algorithm with ours to increase the coverage of error typing. Using our edit extraction algorithm with ARETA enables us to automatically annotate single-token and multi-token edits with various error types. Table 5.2 presents the error types obtained from ARETA using our alignment over the three GEC datasets we use.

To demonstrate the effectiveness of our alignment algorithm, we compare our algorithm to the alignments generated by the M² scorer, a standard Levenshtein edit distance, and ARETA. Table 5.1 presents the evaluation results of the alignment algorithms against the manual gold alignments of the QALB-2014 and QALB-2015 Dev sets in terms of precision (P), recall (R), and alignment error rate (AER) (Mihalcea and Pedersen, 2003; Och and Ney, 2003). Results show that our alignment algorithm is superior across all metrics.

	Tag	Error Description	Example	QALB-2014		QALB-2015		ZAEBUC	
Orthography (O)	OA	Alif, Ya & Alif-Maqsura	علي ← على	7,627	3%	290	2%	27	0%
	OC	Char Order	تبرينا ← تربينا	466	0%	45	0%	30	0%
	OD	Additional Char	يعدوم ← يدوم	4,086	1%	283	2%	103	2%
	OG	Lengthening short vowels	نقيم ← نقيم	0	0%	0	0%	0	0%
	OH	Hamza errors	اكثر ← أكثر	90,579	30%	1,076	8%	1,905	32%
	OM	Missing char(s)	سالىن ← سائلين	4,062	1%	361	3%	123	2%
	ON	Nun & Tanwin Confusion	ثوين ← ثوب	0	0%	0	0%	0	0%
	OR	Char Replacement	مصلنا ← وصلنا	8,350	3%	762	6%	162	3%
	OS	Shortening long vowels	أوقت ← أوقات	0	0%	0	0%	0	0%
	OT	Ha/Ta/Marbuta Confusion	مشاركه ← مشاركة	14,688	5%	54	0%	408	7%
	OW	Confusion in Alif Fariqa	وكانو ← وكانوا	1,885	1%	32	0%	12	0%
	OO	Other orthographic errors	-	1,632	1%	38	0%	148	2%
Morphology (M)	MI	Word inflection	معروف ← عارف	1,360	0%	400	3%	127	2%
	MT	Verb tense	تفرحني ← أفرحتني	76	0%	136	1%	4	0%
	MO	Other morphological errors	-	15	0%	7	0%	3	0%
Syntax (X)	XC	Case	رائع ← رائع	5,980	2%	279	2%	201	3%
	XF	Definiteness	السن ← سن	852	0%	835	6%	51	1%
	XG	Gender	الغربي ← الغربية	809	0%	317	2%	86	1%
	XM	Missing word	Null ← على	1,375	0%	763	6%	68	1%
	XN	Number	فكرتي ← أفكاري	1,107	0%	210	2%	30	0%
	XT	Unnecessary word	على ← Null	1,047	0%	418	3%	116	2%
	XO	Other syntactic errors	-	3,270	1%	122	1%	57	1%
Semantics (S)	SF	Conjunction error	سبحان ← فسبحان	96	0%	46	0%	4	0%
	SW	Word selection error	من ← عن	4,711	2%	865	7%	120	2%
	SO	Other semantic errors	-	380	0%	114	1%	27	0%
Punctuation (P)	PC	Punctuation confusion	قال. ← قال:	11,361	4%	854	7%	237	4%
	PM	Missing punctuation	العظيم ← العظيم،	97,271	32%	2,915	22%	479	8%
	PT	Unnecessary punctuation	العام. ← العام	5,553	2%	213	2%	204	3%
	PO	Other errors in punctuation	-	0	0%	0	0%	0	0%
Merge	MG	Words are merged	لا يلزم ← لا يلزم	15,063	5%	377	3%	849	14%
Split	SP	Words are split	وقال ← وقال	7,828	3%	80	1%	49	1%
Unknown	UNK	Unkown Errors	الظالمون ← الذين ظلموا	2,053	1%	303	2%	93	2%
Comb.	-	Error Combinations	انسانيه ← إنسانية	11,304	4%	848	7%	314	5%
				304,886		13,043		6,037	

Table 5.2: The statistics of the error types in the Train sets of QALB-2014, QALB-2015, and ZAEBUC. The error types are based on the extended ALC (Alfaifi et al., 2013) taxonomy as used by Belkebir and Habash (2021).

Alignments	Erroneous	13 وإيجابيه wAyjAbyh	12 سلبية slbyh	11 منها mnhA	10 اثار AθAr	9 لها lhA	8 ف f	7 بحكمه bHkmh	6 الإجتماعي AlĀjtmAcy	5 التواصل AltwaSl	4 وسائل wsAyl	3 من إستخدام ĀstxdAm	2 من mn	1 لا بد lAbd
	Corrected	14 . وإيجابيه wAyjAbyh	13 سلبية slbyh	12 . سلبية slbyh	11 آثار ĀθAr	10 فلها flhA	9 ، bHkmh	8 بحكمة AlĀjtmAcy	7 الاجتماعي AltwaSl	6 التواصل wsAyl	5 وسائل AstxdAm	4 استخدام mn	3 من bd	2 بد lA

Edits	M²	R										K	K	R	K	S
	Lev.	R		M		R	M		R	R						
	ARETA	R	K	D	R		R									
	Ours	I	R	K	D		M				I					

Error Type	43-Class	PM	OH+OT		Delete	OH	Merge	PM	OT	OH			OH		Split
	13-Class	P	O		Delete	O	Merge	P	O	O			O		Split
	2-Class	E	E		E	E	E	E	E	E			E		E

Figure 5.1: An example showing the differences between the alignments of the M² scorer, a standard Levenshtein distance, ARETA, and our proposed algorithm. The edit operations are keep (**K**), replace (**R**), insert (**I**), delete (**D**), merge (**M**), and split (**S**). Dotted lines between the erroneous and corrected sentences represent gold alignment. The last three rows present different granularities of ARETA error types based on our alignment. The sentence in the figure can be translated as “*Social media must be used wisely, as it has both negative and positive effects*”.

Figure 5.1 presents an example of the different alignments generated by the algorithms we evaluated. The M² scorer’s alignment over-clusters multiple edits into a single edit (words 6–13). This is not ideal, particularly because the M² scorer does not count partial matches during the evaluation, which leads to underestimating the models’ performances (Felice and Briscoe, 2015). A standard Levenshtein alignment does not handle merges correctly, e.g., words 8 and 9 in the erroneous sentence are aligned to words 9 and 10 in the corrected version. Among the drawbacks of ARETA’s alignment is that it does not handle merges, e.g., erroneous words 8 and 9 are aligned with corrected words 9 and 10, respectively.

5.3.2 Arabic Grammatical Error Correction

Recently developed GEC models rely on Transformer-based architectures, from standard Seq2Seq models to edit-based systems built on top of Transformer encoders. Given Arabic’s morphological richness and the relatively small size of available data, we explore different GEC models, from morphological analyzers and rule-based systems to pre-trained Seq2Seq models. Primarily, we are interested in exploring modeling approaches to address the following two questions: **RQ1)** Does morphological preprocessing enhance Arabic GEC? **RQ2)** Does explicitly modeling GED improve Arabic GEC? We also evaluate LLMs on Arabic GEC to compare their performance with specialized models.

Morphological Disambiguation (Morph) We use the current SOTA MSA morphological analyzer and disambiguator from CAMEL Tools (Inoue et al., 2022; Obeid et al., 2020). Given an input sentence, the analyzer generates a set of potential morphological analyses for each word and the disambiguator selects the optimal analysis in context. The analyses include minimal spelling corrections for common errors, diacritizations, POS tags, and lemmas. We use the dediacritized (§2.3) spellings as the corrections.

Maximum Likelihood Estimation (MLE) We exploit our alignment algorithm to build a simple lookup model to map erroneous words to their corrections. We implement this model as a bigram maximum likelihood estimator over the training data: $P(c_i|w_i, w_{i-1}, e_i)$; where w_i and w_{i-1} are the erroneous word (or phrases in case of a merge error) and its bigram context, e_i is the error type of w_i , and c_i is the correction of w_i . During inference, we pick the correction that maximizes the MLE probability. If the bigram context (w_i and w_{i-1}) was not observed during training, we backoff to a unigram. If the erroneous input word was not observed in training, we pass it to the output.

LLMs Similar to Chapter 4 on gender rewriting, we evaluate four LLMs: two commercial models (OpenAI’s GPT-3.5-turbo and GPT-4o (OpenAI et al., 2024)) and two open-source, Arabic-centric models (Jais-30B-Chat (Sengupta et al., 2023) and Fanar LLM (Team et al., 2025)). We use both English and Arabic prompts with 0-shot and 5-shot strategies. To elicit minimal edit-style corrections, we design the prompts to keep the LLMs’ outputs as close as possible to the original input in phrasing and lexical choices. Additionally, we incorporate GED predictions directly into the prompts, explicitly marking erroneous words to guide the models further. Our prompts are presented in Tables B.1, B.2, and B.3 in Appendix B.1.

Seq2Seq with GED Models We experiment with two newly developed pretrained Arabic Transformer-based Seq2Seq models: **AraBART** (Kamal Eddine et al., 2022) (pretrained on 24GB of MSA data mostly in the news domain), and **AraT5** (Nagoudi et al., 2022) (pretrained on 256GB of both MSA and Twitter data). We extend the Seq2Seq models we use to incorporate token-level GED information during training and inference. Specifically, we feed predicted GED tags as auxiliary input to the Seq2Seq models. We add an embedding layer to the encoders of AraBART and AraT5 right after their corresponding token embedding layers, allowing us to learn representations for the auxiliary GED input. The GED embeddings have the same dimensions as the positional and token embeddings, so all three embeddings can be summed before they are passed to the multi-head attention layers in the encoders. Our approach is similar to what was done by Yuan et al. (2021), but it is much simpler as it reduces the model’s size and complexity by not introducing an additional encoder to process GED input. Since the training data we use is relatively small, not drastically increasing the size of AraBART and AraT5 becomes important not to hinder training.

5.4 Experimental Setup

5.4.1 Data

We report on three publicly available Arabic GEC datasets. The first two come from the **QALB-2014** (Mohit et al., 2014a) and **QALB-2015** (Rozovskaya et al., 2015b) shared tasks. The third is the newly created **ZAEBUC** dataset (Habash and Palfreyman, 2022). None of them were manually annotated for specific error types. Dataset statistics are presented in Table 5.3. QALB-2014 consists of Native/L1 user comments from the Aljazeera news website, whereas QALB-2015 consists of essays written by Arabic L2 learners with various levels of proficiency. Both datasets have publicly available training (Train), development (Dev), and test (Test) splits. The ZAEBUC dataset comprises essays written by Native Arabic speakers, which were manually corrected and annotated for writing proficiency using the Common European Framework of Reference (CEFR) (Council of Europe, 2001). Since the ZAEBUC dataset did not have standard splits, we randomly split it into Train (70%), Dev (15%), and Test (15%), while keeping a balanced distribution of CEFR levels. The three sets vary in a number of dimensions: domain, level, number of words, percentage of erroneous words, and types of errors.

The three sets vary in a number of dimensions: domain, level, number of words, percentage of erroneous words, and types of errors. Table 5.2 presents automatic error type distributions over the training portions of the three datasets. Orthographic errors are more common in the L1 datasets (QALB-2014 and ZAEBUC) compared to the L2 dataset (QALB-2015), with Hamza errors constituting 30% and 32% of all errors in QALB-2014 and ZAEBUC, respectively. In contrast, morphological, syntactic, and semantic errors are more common in QALB-2015. Punctuation errors are more common in QALB-

Dataset	Split	Lines	Words	Err. %	Level	Domain
QALB-2014	Train-L1	19,411	1,021,165	30%	Native	Comments
	Dev-L1	1,017	53,737	31%	Native	Comments
	Test-L1	968	51,285	32%	Native	Comments
QALB-2015	Train-L1	310	43,353	30%	L2	Essays
	Dev-L1	154	24,742	29%	L2	Essays
	Test-L2	158	22,808	27%	L2	Essays
	Test-L1	920	48,547	29%	Native	Comments
ZAEBUC	Train-L1	150	25,127	24%	Native	Essays
	Dev-L1	33	5,276	25%	Native	Essays
	Test-L1	31	5,118	26%	Native	Essays

Table 5.3: Corpus statistics of Arabic GEC datasets.

2014 and QALB-2015, compared with ZAEBUC. However, it is worth noting that the reported inter-annotator agreement for punctuation correction was relatively low in both QALB-2014 and 2015 (Mohit et al., 2014a; Zaghouani et al., 2015b), highlighting the inconsistencies of punctuation usage in Arabic. Moreover, error combinations constitute 4%, 7%, 5% in QALB-2014, QALB-2015, and ZAEBUC, respectively.

5.4.2 Experiments

Evaluation Metrics GEC systems are most commonly evaluated using reference-based metrics such as the MaxMatch (M^2) scorer (Dahlmeier and Ng, 2012), ERRANT (Bryant et al., 2017), and GLUE (Napoles et al., 2015). In our work, as in Chapter 4 on gender rewriting, we use the M^2 scorer because it is language agnostic and was the main evaluation metric used in previous work on Arabic GEC. The M^2 scorer compares hypothesis edits made by a GEC system against human-annotated reference edits and calculates the precision (P), recall (R), and $F_{0.5}$. In terms of GED, we follow previous work (Bell et al., 2019; Kaneko and Komachi, 2019; Yuan et al., 2021) and use macro precision (P), recall (R), and $F_{0.5}$ for evaluation.

Grammatical Error Detection We build word-level GED classifiers using Transformer-based pretrained language models. From the many available Arabic monolingual BERT models (Antoun et al., 2020; Abdul-Mageed et al., 2021a; Lan et al., 2020; Safaya et al., 2020; Abdelali et al., 2021), we chose to use CAMeLBERT MSA (Inoue et al., 2021), as it was pretrained on the largest MSA dataset to date.

In our GED modeling experiments, we project multi-token error type annotations to single-token labels. In the case of a Merge error (many-to-one), we label the first token as *Merge-B* (Merge beginning) and all subsequent tokens as *Merge-I* (Merge inside). For all other multi-token error types, we repeat the same label for each token. We further label all deletion errors with a single *Delete* tag. To reduce the output space of the error tags, we only model the 14 most frequent error combinations (appearing more than 100 times). We ignore unknown errors when we compute the loss during training; however, we penalize the models for missing them in the evaluation.

Since the majority of insertion errors are related to missing punctuation marks rather than missing words (see Table 5.2), and due to inconsistent punctuation error annotations (Mohit et al., 2014b), we exclude insertion errors from our GED modeling and evaluation. We leave the investigation of insertion errors to future work. The full GED output space we model consists of 43 error tags (43-Class).

We take advantage of the modularity of the ARETA error tags to conduct multi-class GED experiments, reducing the 43 error tags to their corresponding 13 main error categories as well as to a binary space (correct/incorrect). The statistics of the error tags we model across all datasets are in Table B.5. Figure 5.1 shows an example of error types at different granularity levels.

Grammatical Error Correction We explore different variants of the above-mentioned Seq2Seq models. For each model, we study the effects of applying morphological preprocessing (**+Morph**), providing GED tags as auxiliary input (**+GED**), or both (**+Morph+GED**). Applying morphological preprocessing simply means correcting the erroneous input using the morphological disambiguator before training and inference. When applying morphological preprocessing and providing GED tags to the models (**+Morph+GED**), both the GED and GEC systems are trained and tested on morphologically preprocessed text. To increase the robustness of the models that take GED tags as auxiliary input, we use predicted (not gold) GED tags when we train the GEC systems. For each dataset, we run its respective GED model on the same training data it was trained on and we pick the predictions of the *worst* checkpoint. During inference, we resolve merge and delete errors before feeding erroneous sentences to the model. This experimental setup yields the best performance across all GEC models.

To ensure fair comparison to previous work on Arabic GEC, we follow the same constraints that were introduced in the QALB-2014 and QALB-2015 shared tasks: systems tested on QALB-2014 are only allowed to use the QALB-2014 training data, whereas systems tested on QALB-2015 are allowed to use the QALB-2014 and QALB-2015 training data. For ZAEBUC, we train our systems on the combinations of the three training datasets. We report our results in terms of precision (P), recall (R), F_1 , and $F_{0.5}$. It is worth noting that F_1 was the official metric used in the QALB-2014 and QALB-2015 shared tasks. However, we follow the most recent work on GEC and use $F_{0.5}$ (weighing precision twice as much as recall) as our main evaluation metric.

		43-Class				13-Class				2-Class			
		P	R	F _{0.5}	Acc.	P	R	F _{0.5}	Acc.	P	R	F _{0.5}	Acc.
QALB-2014	Dev-L1	56.7	48.4	53.3	94.1	69.0	58.7	65.3	94.7	95.8	92.7	95.1	96.1
	Test-L1	55.0	45.5	50.6	93.6	58.1	54.2	56.8	94.1	95.4	91.5	94.5	95.5
QALB-2015	Dev-L2	39.0	35.0	36.9	84.5	55.1	47.3	51.7	85.3	87.0	80.4	85.2	88.9
	Test-L1	51.8	45.3	49.4	95.6	66.5	56.2	60.7	89.9	96.2	93.9	95.7	96.7
	Test-L2	37.0	35.4	35.8	85.5	52.8	48.6	51.0	94.9	88.6	81.3	86.6	86.5
ZAEBUC	Dev-L1	50.9	43.7	47.5	92.6	57.1	52.9	55.7	93.3	95.7	92.8	95.1	95.5
	Test-L1	54.9	43.3	49.8	91.9	69.2	56.6	62.4	92.6	95.5	92.5	94.8	95.2

Table 5.4: GED results on the Dev and Test sets in terms of macro precision, recall, F_{0.5}, and accuracy.

5.4.3 Results

GED Results

Table 5.4 presents the GED granularity results. Unsurprisingly, all numbers go up when we model fewer error types. However, modeling more error types does not significantly worsen the performance in terms of error detection accuracy. It seems that all systems are capable of detecting comparable numbers of errors despite the number of classes, but the verbose systems struggle with detecting the specific class labels.

GEC Results

Table 5.5 presents the GEC results on the Dev sets.

Baselines The Morph system which did not use any training data constitutes a solid baseline for mostly addressing the noise in Arabic spelling. The MLE system claims the highest precision of all compared systems, but it suffers from low recall as expected.

LLMs We present the LLMs results using their best setups, optimized for average F_{0.5} across all datasets based on the prompt language and strategy (0-shot vs. 5-shot). Full results are provided in Table B.4 in Appendix B.2. GPT-4o consistently achieves the best

	QALB-2014			QALB-2015			ZAEBUC			Avg.
	P	R	F _{0.5}	P	R	F _{0.5}	P	R	F _{0.5}	F _{0.5}
B&B (2015)	-	-	-	56.7	34.8	50.4	-	-	-	-
W+ (2018)	80.0	62.5	75.8	-	-	-	-	-	-	-
Morph	76.5	30.6	58.9	56.2	9.4	28.2	78.0	36.9	63.8	50.3
MLE	89.2	41.5	72.5	73.7	20.1	48.0	<u>90.1</u>	55.6	80.1	66.9
+Morph	88.5	44.9	74.1	68.3	22.0	48.0	89.1	61.8	81.9	68.0
Fanar	69.7	63.7	68.4	58.0	40.7	53.5	76.3	73.6	75.8	65.9
Jais-30B-Chat	53.8	44.5	51.6	46.3	19.1	36.0	51.5	29.4	44.8	44.1
GPT-3.5-turbo	70.6	54.8	66.7	59.6	39.6	54.1	70.8	70.3	70.7	63.9
GPT-4o	80.7	65.7	77.2	70.6	49.2	65.0	86.5	76.8	84.3	75.5
+GED ²	82.1	62.2	77.2	74.4	41.2	64.0	90.4	72.3	86.1	75.8
AraT5	82.5	66.3	78.6	69.3	39.4	60.2	84.1	67.4	80.1	73.0
+Morph	83.1	65.8	78.9	69.7	40.6	60.9	85.0	71.3	81.8	73.9
+GED ⁴³	82.6	67.1	79.0	69.5	41.9	61.4	85.7	66.7	81.0	73.8
+Morph +GED ⁴³	83.1	67.9	79.6	68.4	41.5	60.6	85.2	71.2	82.0	74.0
AraBART	83.2	64.9	78.7	68.6	42.6	61.2	87.3	70.6	83.4	74.4
+Morph	82.4	67.2	78.8	68.5	44.3	61.7	87.2	71.6	83.6	74.7
+GED ⁴³	83.3	65.9	79.1	68.2	45.3	61.9	87.2	72.9	83.9	75.0
+Morph +GED ⁴³	83.4	66.3	79.3	68.2	<u>46.6</u>	<u>62.4</u>	87.3	<u>73.6</u>	<u>84.2</u>	<u>75.3</u>

Table 5.5: GEC results on the Dev sets of QALB-2014, QALB-2015, and ZAEBUC. B&B (2015) and W+ (2018) refer to [Bougares and Bouamor \(2015\)](#) and [Watson et al. \(2018a\)](#), respectively. The best overall results are in bold. Results of our best systems are underlined.

performance across the LLMs on all datasets, surpassing previous work as well as our MLE and Morph baselines. Notably, GPT-4o also ranks as the top-performing system overall on QALB-2015 and ZAEBUC. Indicating erroneous words using our binary GED systems before prompting GPT-4o (GPT-4o + GED²) improves precision but reduces recall on all datasets. This strategy results in the best performance on ZAEBUC, while having no effect on QALB-2014 and leading to a drop in performance on QALB-2015. Although GPT-4o achieves the best performance, these results should be interpreted with caution, as the model is closed-source and cannot be replicated. Additionally, the lack of transparency around the training data used to build GPT-4o raises concerns about

	QALB-2014			QALB-2015			ZAEBUC			Avg.
	P	R	F _{0.5}	P	R	F _{0.5}	P	R	F _{0.5}	F _{0.5}
43-Class	85.5	73.3	82.8	73.9	57.2	69.8	89.8	82.0	88.1	80.2
13-Class	85.4	73.2	82.6	73.5	55.9	69.2	89.4	82.2	87.9	79.9
2-Class	84.2	72.1	81.4	71.6	54.5	67.4	86.6	80.0	85.2	78.0
43-Class	83.4	66.3	79.3	68.2	46.6	62.4	87.3	73.6	84.2	75.3
13-Class	83.9	65.7	79.5	68.0	46.6	62.3	87.6	73.9	84.5	75.4
2-Class	82.5	67.3	79.0	68.3	45.0	61.9	86.0	72.3	82.9	74.6

Table 5.6: GED granularity results when used within the best GEC system on the Dev sets of QALB-2014, QALB-2015, and ZAEBUC. Results in grey indicate using gold GEC labels (i.e., Oracle). The best results are in bold.

potential data contamination, making it unclear whether the model has been exposed to the datasets we test on during training.

Seq2Seq Models AraT5 and AraBART outperform previous work on QALB-2014 and QALB-2015, with AraBART being the better model on average.

Does morphological preprocessing improve Arabic GEC? Across all models (MLE, AraT5, and AraBART), training and testing on morphologically preprocessed text improves the performance, except for MLE+Morph on QALB-2015 where there is no change in $F_{0.5}$.

Does GED help Arabic GEC? We start off by using the most fine-grained GED model (43-Class) to exploit the full effect of the ARETA GED tags and to guide our choice between AraBART and AraT5. Using GED as an auxiliary input in both AraT5 and AraBART improves the results across all three Dev sets, with AraBART+GED demonstrating superior performance compared to the other models, on average. Applying morphological preprocessing as well as using GED as an auxiliary input yields the best performance across the three Dev sets, except for QALB-2015 in the case of AraT5+Morph+GED. Overall, among our models, **AraBART+Morph+GED** achieves the best average performance in terms of $F_{0.5}$. The improvements using GED with GEC

systems are mostly due to recall. To study the effect of GED granularity on GEC, we train two additional AraBART+Morph+GED models with 13-Class and 2-Class GED tags.

The results in Table 5.6 show that 13-Class GED was best in QALB-2014 and ZAEBUC, whereas 43-Class GED was best in QALB-2015 in terms of $F_{0.5}$. However, in terms of precision and recall, GED models with different granularity behave differently across the three Dev sets. On average, using any GED granularity improves over AraBART, with 13-Class GED yielding the best results, although it is only 0.1 higher than 43-Class GED in terms of $F_{0.5}$. For completeness, we further estimate an oracle upper bound by using gold GED tags with different granularity. The results (in Table 5.6) show that using GED with different granularity improves the results considerably. This indicates that GED is providing the GEC system with additional information; however, the main bottleneck is the GED prediction reliability as opposed to GED granularity. Improving GED predictions will most likely lead to better GEC results.

Test Results Table 5.7 presents the Test results. GPT-4o outperforms previous work on QALB-2014, QALB-2015-L1, and QALB-2015-L2. Incorporating GED predictions by marking erroneous words further improves its performance on QALB-2015-L1, QALB-2015-L2, and ZAEBUC. Notably, it is the best-performing system overall on QALB-2015-L2 and ZAEBUC. However, as previously noted, its results should be interpreted with caution due to the model’s closed nature and potential data contamination concerns. Turning to our models, we observe that different GED granularity levels yield optimal results across the three Dev sets when combined with AraBART+Morph. Therefore, we evaluate all GED variants on the Test sets. On QALB-2014, using Morph, GED, or both improves the results over AraBART, except for 2-Class GED. AraBART+43-

	QALB-2014			QALB-2015-L1			QALB-2015-L2			ZAEBUC			Avg.
	P	R	F _{0.5}	P	R	F _{0.5}	P	R	F _{0.5}	P	R	F _{0.5}	F _{0.5}
B&B (2015)	-	-	-	-	-	-	54.1	33.3	48.1	-	-	-	-
S+ (2022)	79.1	65.8	76.0	78.4	70.4	76.6	-	-	-	-	-	-	-
GPT-4o	81.5	65.5	77.7	81.1	74.3	79.6	69.1	50.0	64.2	84.4	75.9	82.5	76.0
+GED ²	82.9	62.0	77.7	82.8	71.1	80.2	75.2	42.5	65.1	89.0	73.2	85.3	77.1
AraBART	84.0	64.7	79.3	82.0	71.7	79.7	<u>69.6</u>	43.5	62.1	<u>86.0</u>	71.6	82.7	75.9
+Morph	83.3	67.4	79.5	81.7	73.0	79.8	68.7	43.6	61.6	85.3	71.8	82.3	75.8
+GED ⁴³	84.2	65.4	79.6	81.2	72.4	79.3	69.0	<u>45.4</u>	<u>62.5</u>	85.4	72.6	82.5	76.0
+Morph+GED ⁴³	83.9	65.7	79.5	<u>82.6</u>	72.1	80.3	67.6	45.2	61.5	85.4	<u>73.7</u>	82.7	76.0
+GED ¹³	84.1	65.0	79.4	81.5	72.7	79.5	69.3	44.9	<u>62.5</u>	85.9	73.4	<u>83.1</u>	76.1
+Morph+GED ¹³	83.9	65.3	79.4	81.1	73.4	79.5	68.2	44.8	61.8	85.2	<u>73.7</u>	82.6	75.8
+GED ²	83.8	64.5	79.1	81.4	71.5	79.2	69.1	44.9	62.4	85.7	71.5	82.4	75.8
+Morph+GED ²	83.0	67.0	79.2	81.3	<u>73.8</u>	79.7	68.1	45.3	61.9	85.7	72.4	82.7	75.9

Table 5.7: GED granularity results when used within GEC on the Test sets of QALB-2014, QALB-2015, and ZAEBUC. B&B (2015) and S+ (2022) refer to [Bougares and Bouamor \(2015\)](#) and [Solyman et al. \(2022\)](#), respectively. The best overall results are in bold. Results of our best systems are underlined.

Class GED is the best performer with a 0.3 increase in $F_{0.5}$, although this difference is not statistically significant. Statistical significance was determined using a two-sided approximate randomization test ([Graham et al., 2014](#); [Dror et al., 2018](#)). It is worth noting that AraBART+Morph achieves the highest recall on QALB-2014 (2.7 increase over AraBART and statistically significant at $p < 0.05$).

For QALB-2015-L1, using GED by itself across all granularity did not improve over AraBART, but when combined with Morph, the 43-Class GED model yields the best performance in $F_{0.5}$ (0.6 increase statistically significant at $p < 0.05$). When it comes to QALB-2015-L2, Morph does not help, but using GED alone improves the results over AraBART, with 43-Class and 13-Class GED being the best (0.4 increase). Lastly, in ZAEBUC, Morph does not help, but using 13-Class GED by itself improves over AraBART (0.4 increase). Overall, all the improvements we observe are attributed to recall, which is consistent with the Dev results.

	QALB-2014		QALB-2015		ZAEBUC	
	AraBART	Best System	AraBART	Best System	AraBART	Best System
Delete	39.8	40.6	33.0	36.4	47.5	51.9
Merge-B	91.2	93.1	84.2	86.0	96.7	96.7
Merge-I	91.0	93.1	83.7	85.8	96.7	96.7
M	25.5	28.4	37.0	40.8	48.9	48.6
M+O	54.8	37.7	17.2	15.2	100.0	55.6
O	94.1	94.4	80.2	80.1	93.9	94.3
O+X	67.7	73.9	0.0	0.0	0.0	0.0
P	76.2	77.3	64.8	63.5	66.8	62.8
S	43.7	45.3	33.2	31.9	36.1	40.4
X	59.6	62.3	58.4	63.7	69.5	71.2
Split	88.0	87.4	78.9	78.9	88.2	88.2
UNK	49.8	56.1	35.0	31.6	55.0	63.1
C	96.3	96.8	90.3	91.3	95.4	96.1
Macro Avg	67.5	68.2	53.5	54.3	68.8	66.6

Table 5.8: Specific error type performance of AraBART and our best system on the Dev sets of QALB-2014, QALB-2015, and ZAEBUC. Results are reported in terms of $F_{0.5}$. The best results are in bold.

5.4.4 Error Analysis

To investigate which error types benefit from using GED information within GEC, we perform a detail error analysis over Dev sets. Table 5.8 presents specific error type performance of AraBART (baseline) and our best system (AraBART+Morph+GED¹³). Our best system is the better performer on average in QALB-2014 and QALB-2015 but not in ZAEBUC. However, in the case of ZAEBUC, it is worth noting that AraBART’s 2.2 average macro $F_{0.5}$ improvements over our best system is coming from fixing the M+O errors, which only appeared once in the ZAEBUC Dev set.

5.5 Summary

In this chapter, we conducted a comprehensive study on MSA GEC. We presented the first results using Transformer-based pretrained Seq2Seq models and benchmarked both open-source Arabic-centric and commercial LLMs. We introduced the task of multi-class MSA GED and showed that incorporating GED predictions as auxiliary input improves GEC performance for both Seq2Seq models and LLMs. This demonstrates the importance of control in tasks where the input and output are largely similar, such as GEC, as explicitly modeling error types guides both model families toward more precise corrections. Additionally, we explored the role of contextual morphological preprocessing in improving error correction within Seq2Seq models. Our models achieved SOTA results on two Arabic GEC shared task datasets and established a strong benchmark on a recently created dataset. Finally, while commercial LLMs such as GPT-4o achieved impressive performance, these results should be interpreted with caution due to the closed nature of these models and the lack of transparency around their training data, raising concerns about reproducibility and potential data contamination.

Chapter 6

Dialectal Text Normalization

Dialectal Arabic is the primary spoken language used by native Arabic speakers in daily communication. With the rise of social media, its use in written form has grown significantly. However, the absence of standardized orthographies for Arabic dialects, coupled with the inherent noise in user-generated content, poses major challenges for NLP applications dealing with Dialectal Arabic. In this chapter, we present a comprehensive study on the final language generation task explored in this dissertation: dialectal text normalization. This task, known as CODAfication, involves normalizing Dialectal Arabic into the Conventional Orthography for Dialectal Arabic (CODA). Similar to GEC, we benchmark pretrained Seq2Seq models on CODAfication and demonstrate that conditioning these models on dialect identification predictions enhances performance. Additionally, we benchmark open-source and commercial LLMs to assess their performance on CODAfication. We present results using a unique parallel corpus covering multiple Arabic dialects, focusing on five cities: Beirut, Cairo, Doha, Rabat, and Tunis.

6.1 Introduction

Arabic exhibits a diglossic (Ferguson, 1959) linguistic situation where a non-standard variety, Dialectal Arabic (DA), coexists with Modern Standard Arabic (MSA), the standard form of the language. Complicating matters, DA consists of multiple regional dialects, such as Egyptian, North African, Levantine, and Gulf Arabic, that differ from both MSA and each other in phonology, morphology, and lexicon (§2.1). While primarily spoken, DA has increasingly been used in written form on social media, where the lack of a standardized orthography (Habash et al., 2018) leads to highly variable and noisy text. This high degree of noise poses major challenges for NLP systems as it increases data sparsity. Such noise can be handled using modeling techniques that normalize DA if it is used as an input to the system, e.g., in machine translation from dialects to other languages. However, challenges arise when the dialect itself is the desired output, for example, in automatic speech recognition systems (Ali et al., 2019; Sahyoun and Shehata, 2023). Consequently, evaluating and optimizing these systems can become problematic.

To mitigate the lack of orthographic standards for DA, several efforts in Arabic NLP introduced a common convention for DA spelling, named Conventional Orthography for Dialectal Arabic (CODA) (Habash et al., 2012a, 2018). However, CODA has largely been treated as a secondary task in areas such as morphological disambiguation, diacritization, and lemmatization, rather than as a main, primary task.

In our work, we explore the task of CODAfication, normalizing DA text into the CODA convention as a standalone task. We work with a unique parallel corpus of multiple Arabic dialects (Eryani et al., 2020), focusing on five cities: Beirut, Cairo, Doha, Rabat, and Tunis. We benchmark pretrained Seq2Seq models and LLMs on CODAfication and show that conditioning Seq2Seq models on dialect identification predictions improves

CODAfication performance. Formally, given a dialectal Arabic sentence X and its corresponding dialect D , the task is to generate the CODAfied version Y according to:

$$P(Y|X, D) = \prod_{t=1}^n P(y_t|y_1, \dots, y_{t-1}, X, D)$$

6.2 Background and Related Work

Dialectal Arabic Text Normalization

DA NLP research has been receiving a considerable amount of attention, mainly due to the availability of monolingual and multilingual DA corpora (McNeil and Faiza, 2011; Zaidan and Callison-Burch, 2011; Zbib et al., 2012; Cotterell and Callison-Burch, 2014; Jeblee et al., 2014; Al-Badrashiny and Diab, 2016; Zaghoulani and Charfi, 2018b; Abdul-Mageed et al., 2018a; Bouamor et al., 2019). While MSA has well-defined orthographic standards, none of the Arabic dialects do today. As a result, almost all DA corpora were created without following any spelling conventions or standards, which are necessary for building robust DA NLP applications, e.g., machine translation (Erdmann et al., 2017).

To mitigate this problem, several efforts have been introduced to standardize and develop orthographic conventions for Arabic dialects. Habash et al. (2012a) introduced the Conventional Orthography for Dialectal Arabic (CODA), the very first attempt to create guidelines and spelling conventions for Egyptian Arabic orthography. The convenience CODA offered by providing a standardized orthography led to the creation of many CODA extensions covering various dialects including Tunisian, Algerian, Palestinian, Moroccan, Yemeni, and Gulf Arabic (Zribi et al., 2014; Saadane and Habash, 2015; Jarrar et al., 2016; Turki et al., 2016; Khalifa et al., 2018). Each of these extensions tended to

curate its own list of exceptional spellings for closed class words. [Habash et al. \(2018\)](#) introduced a unified set of guidelines for Arabic Dialect orthography – dubbed CODA* (CODA Star). CODA has been used in the creation of a number of resources for DA NLP ([Habash et al., 2012b](#); [Eskander et al., 2013](#); [Maamouri et al., 2014](#); [Diab et al., 2014](#); [Pasha et al., 2014b](#); [Jarrar et al., 2016](#); [Khalifa et al., 2018](#); [Eryani et al., 2020](#)). Most relevant to this paper is the work of [Eryani et al. \(2020\)](#) who extended a portion of the MADAR Corpus ([Bouamor et al., 2018](#)) to create the MADAR CODA Corpus, a collection of 10,000 sentences from five Arabic city dialects (Beirut, Cairo, Doha, Rabat, and Tunis) represented in the CODA standard in parallel with their original raw form. We use this corpus to train and test our models.

In terms of modeling approaches to CODAfication, the first work was proposed by [Eskander et al. \(2013\)](#) where they introduced CODAFY, a feature-based machine learning classifier to normalize Egyptian Arabic into CODA. [Al-Badrashiny et al. \(2014\)](#) and [Shazal et al. \(2020\)](#) targeted CODA output for dialectal Arabizi (Romanized Arabic) input. Most other approaches attempted to normalize DA texts into CODA as part of morphological analysis and disambiguation ([Pasha et al., 2014a](#); [Zalmout et al., 2018](#); [Khalifa et al., 2020](#); [Zalmout and Habash, 2020](#); [Obeid et al., 2022](#)). Our work is most similar to the one of [Eskander et al. \(2013\)](#) where we consider the task of CODAfication as a standalone text normalization task.

There has been some work on normalizing DA into MSA ([Shaalán et al., 2007](#); [Salloum and Habash, 2011, 2012](#); [Alnajjar and Hämäläinen, 2024](#)). While all this work is similar to ours in that dialectal input is processed, our output is still dialectal and not in MSA. Moreover, CODAfication has some similarities to GEC for MSA (§5). However, CODAfication is different from GEC for MSA since GEC assumes a standard orthography that the writer is also assumed to aim for.

Dialect Identification

Dialect Identification (DID) is the task of determining the dialect of a given speech or text fragment (Etman and Beex, 2015). As informal conversations in both real-world and online settings are predominantly conducted DA, there has been a growing interest in developing and scaling automatic Arabic DID systems. This is reflected in the organization of multiple shared tasks (Malmasi et al., 2016; Zampieri et al., 2017, 2018; Bouamor et al., 2019; Abdul-Mageed et al., 2021b, 2022, 2023, 2024). In terms of datasets, several mono-dialectal corpora covering different Arabic dialects were built and made available (Gadalla et al., 1997; Diab et al., 2010; Zaidan and Callison-Burch, 2011; Al-Sabbagh and Girju, 2012; Sadat et al., 2014; Smaïli et al., 2014; Cotterell and Callison-Burch, 2014; Jarrar et al., 2016; Khalifa et al., 2016b; Al-Twairesh et al., 2018; El-Haj, 2020). Over time, datasets have expanded to include multi-dialectal resources at various levels of granularity, such as region, country, province, and city (McNeil and Faiza, 2011; Zaidan and Callison-Burch, 2014; Elfardy et al., 2014; Bouamor et al., 2014; Salama et al., 2014; Abdul-Mageed et al., 2018b; Bouamor et al., 2018; Zaghouani and Charfi, 2018a). More recently, research has shown that many instances in existing DID datasets could have multiple labels as opposed to a single label (Keleg and Magdy, 2023). This has led to the development of metrics that assess the degree of dialectness in Arabic text, moving beyond single-label classification (Keleg et al., 2023).

Besides its obvious use for profiling (Rangel et al., 2019), DA identification has proven beneficial for various NLP tasks, including machine translation (Salloum et al., 2014), code-switching detection (Elfardy et al., 2014; Solorio et al., 2014; Molina et al., 2016), and morphological tagging (Obeid et al., 2022). In our work, we leverage sentence-level DID to enhance CODAfication.

CODA

As discussed earlier (§2.3), DA lacks a standardized orthography, leading speakers to write words in ways that often reflect either their phonological or etymological characteristics. This phenomenon, known as *spontaneous orthography*, means that no spelling of a dialectal word can be considered strictly “incorrect”. CODA, proposed by (Habash et al., 2012a) addresses this challenge by proposing a set of guidelines aimed at unifying the writing of DA, providing a consistent and systematic approach to representing dialectal variations. CODA* (Habash et al., 2018)—pronounced *CODA Star*, as in, *for any dialect*—consolidates and standardizes several prior dialect-specific CODA conventions (Habash et al., 2012a; Saadane and Habash, 2015; Turki et al., 2016; Khalifa et al., 2016a; Jarrar et al., 2016).

CODA*, henceforth CODA, is an internally consistent and coherent convention that strives to regulate some of the DA natural spelling tendencies in an internally consistent system and (generally) according to a MSA reference, more or less familiar to everyone. As (Habash et al., 2018) explain, CODA’s design tries to “strike an optimal balance between maintaining a level of dialectal uniqueness and establishing conventions based on MSA-DA similarities,” following a sense that the success of such optimization would ensure CODA stays easily learnable and seamlessly readable to the average Arabic speaker without compromising their ability to interpret a written form in their own dialect. For instance, the Beirut dialect word زغيري *zγyry* /zɒi:ri/ ‘small [feminine singular]’ is written in a form reflective of MSA etymology: صغيرة *Sγyrh̄*. Other examples of CODA from the MADAR CODA Corpus (Eryani et al., 2020) appear in Table 6.1. Note that foreign words pose a particular challenge to CODA due to the ambiguous phonological signals in the Arabic raw text. Consequently, Eryani et al. (2020) adopted a minimalistic

Dialect	Raw	CODA
Beirut	إذا بتريد ، تنان همبرغر و تنان أهوة . بدي آخدون معي . <i>Āza btryd , tnAn hmbryr wtnAn Āhwh . bdy Āxdwn mcy .</i>	اثنين همبرغر واثنين قهوة . بدي آخذهن معي . <i>AḏA btryd , Aḏnyn hmbryr wAḏnyn qhwh . bdy Āxdhn mcy .</i>
Cairo	اثنين هامبورجر واثنين قهوة ، لوسمحت . عليزهم تيك اوي . <i>Atnyn hAmbwryr wAtnyn qhwh , lw smHt . ʕAyzhm tyk AwAy .</i>	اثنين هامبورجر واثنين قهوة ، لو سمحت . عليزهم تيك اوي . <i>Aḏnyn hAmbryr wAḏnyn qhwh , lw smHt . ʕAyzhm tyk Awy .</i>
Doha	اثنين همبرغر واثنين قهوة ، لوسمحت . باخذهم تيك اوي . <i>Aḏnyn hmbryr wAḏnyn qhwh , lw smHt . bĀxdhm tyk Awy .</i>	اثنين همبرجر واثنين قهوة ، لو سمحت . باخذهم تيك اوي . <i>Aḏnyn hmbryr wAḏnyn qhwh , lw smHt . bĀxdhm tyk Awy .</i>
Rabat	جوج هامبورغر و جوج قهيووات ، عافاك . غادي نديهم معايا . <i>jwj hAmbwryr wjwj qhywAt , ʕAfAk . γAdy ndyhm mʕAyA .</i>	جوج هامبورغر و جوج قهيووات ، عافاك . غادي نديهم معاي . <i>jwj hAmbwryr wjwj qhywAt , ʕAfAk . γAdy ndyhm mʕAy .</i>
Tunis	زوز همبرغر وزوز قهاوي ، يعيشك . نحب نهزهم معايا . <i>zwz hmbryr wzvz qhAwy , ycyšk . nHb nhzhm mʕAyA .</i>	زوز همبرغر وزوز قهاوي ، يعيشك . نحب نهزهم معاي . <i>zwz hmbryr wzvz qhAwy , ycyšk . nHb nhzhm mʕAy .</i>

Table 6.1: An example sentence from the MADAR CODA Corpus in its raw and CODA parallel forms across five city dialects. The DA sentences are provided along with their transliterations in the HSB scheme (Habash et al., 2007). The sentence in the table can be translated as “We would like two hamburgers and two coffees. To go, please.”

strategy for CODAifying these words, resulting in some plausible but inconsistent variants.

For example, the word for ‘hamburger’ in Table 6.1 appears as both همبرغر *hmbryr* and همبرجر *hmbryr*.

6.3 Approach

We frame the CODAfication task as a controlled text generation problem. Formally, given a dialectal input sentence X and its dialect D , the goal is to generate the CODAified sentence Y according to $P(Y|X, D)$. One way to condition text generation models on the desired dialect, D , is to represent it as a special “control” token appended to the input sequence $[D; X]$, which acts as a side constraint (Sennrich et al., 2016a). In Seq2Seq models, this allows the encoder to learn a representation for this token as any other token in its vocabulary, and the decoder attends to this representation to guide the generation of the output sequence. This is similar to what we did in the joint gender rewriting modeling experiments (§4.4.1) discussed in Chapter 4. This simple strategy has also been used in various controlled text generation tasks such as machine translation (Sennrich et al.,

Dialect	City	MSA Phrase	DA Phrase	Digit
Beirut	بيروت <i>byrwt</i>	في بيروت نقول <i>fy byrwt nqwl</i>	في بيروت منقول <i>fy byrwt mnqwl</i>	1
Cairo	القاهرة <i>AlqAhrh</i>	في القاهرة نقول <i>fy AlqAhrh nqwl</i>	في القاهرة بنقول <i>fy AlqAhrh bnqwl</i>	2
Doha	الدوحة <i>AldwHh</i>	في الدوحة نقول <i>fy AldwHh nqwl</i>	في الدوحة نقول <i>fy AldwHh nqwl</i>	3
Rabat	الرباط <i>AlrbAT</i>	في الرباط نقول <i>fy AlrbAT nqwl</i>	في الرباط كنقولو <i>fy AlrbAT knqwlw</i>	4
Tunis	تونس <i>twns</i>	في تونس نقول <i>fy twns nqwl</i>	في تونس نقولو <i>fy twns nqwlw</i>	5

Table 6.2: The four different types of control tokens we use in our experiments.

2016b; Sennrich and Haddow, 2016; Johnson et al., 2016; Agrawal and Carpuat, 2019), style transfer (Niu et al., 2017, 2018), and text simplification (Yanamoto et al., 2022; Agrawal and Carpuat, 2023).

Just like our experiments in GEC, we experiment with both **AraBART** (Kamal Eddine et al., 2022) and **AraT5-v2** (Nagoudi et al., 2022; Elmadany et al., 2023). We explore using four different control tokens to pass the dialect information to the models. Table 6.2 presents the control tokens we considered in our experiments:

- **City:** The name of the city where the Arabic dialect is spoken.
- **MSA Phrase:** An MSA phrase that follows the template *في <city> نقول* ('in <city> we say'), where <city> represents one of the five cities whose dialects we are modeling.
- **DA Phrase:** A DA phrase that follows the template *<we-say> <city> في* ('in <city> we say'), where <city> represents one of the five dialects we are modeling, and <we-say> represents a spontaneous orthography of the dialectal version of the phrase 'we say'.

- **Digit:** An ad hoc unique numerical value for each dialect.

During training, we use the gold dialect for each sentence to induce its control tokens. To obtain the dialect during inference, we use the DID system that is available in CAMEL Tools (Obeid et al., 2020). The system is an implementation of Salameh et al. (2018)’s best-performing model on the MADAR shared task on DID (Bouamor et al., 2019). The system models DID for the five city dialects and MSA.

6.4 Experimental Setup

6.4.1 Data

We use the manually annotated MADAR CODA Corpus (Eryani et al., 2020), a collection of 10,000 sentences from five Arabic city dialects (Beirut, Cairo, Doha, Rabat, and Tunis) represented in the CODA standard in parallel with their original raw form. The sentences come from the Multi-Arabic Dialect Applications and Resources (MADAR) Project (Bouamor et al., 2018) and are in parallel across the cities (2,000 sentences from each city).

The corpus is originally split into train and test, with each split consisting of 5,000 parallel sentences (1,000 per dialect). In our setup, we combine the original train and test splits and then divide the data randomly into separate training (Train), development (Dev), and testing (Test) sets. We use a 70/15/15 split, resulting in 1400, 300, and 300 sentences, respectively, per dialect. In total, we end up with 7,000 sentences for Train, 1,500 for Dev, and 1,500 for Test. Table 6.1 shows an example of a sentence from the corpus in its raw and CODA parallel forms across the five city dialects.

Table 6.3 presents the top 10 character-level edit changes from raw text to CODA in the five city dialects. It is noteworthy that while there are many shared transformations,

BEI			CAI			DOH			RAB			TUN		
RAW	CODA	FREQ	RAW	CODA	FREQ	RAW	CODA	FREQ	RAW	CODA	FREQ	RAW	CODA	FREQ
<SPC>		863	<SPC>		1166		A ʾ	150		A ʾ	548		A ʾ	458
Ā ʾ	A ʾ	409	Ā ʾ	A ʾ	608	Ā ʾ	A ʾ	124	A ʾ		352	Ā ʾ	A ʾ	288
Ā ʾ	A ʾ	405	h ʾ	h ʾ	323		h ʾ	82	<SPC>		324	Ā ʾ	A ʾ	189
	A ʾ	324	t ٤	θ ٤	257	ð ٤	Að ٤	62	Ā ʾ	A ʾ	256	<SPC>		175
t ٤	θ ٤	294	Ā ʾ	A ʾ	146	j ٤	tʃ ٤	33	t ٤	θ ٤	190		<SPC>	148
	h ʾ	173		<SPC>	142	j ٤	k ٤	31		<SPC>	168	A ʾ	h ʾ	115
w ٤	h ʾ	138	d ٤	ð ٤	95		<SPC> ʾ <SPC>A	28	A ʾ	h ʾ	160	A ʾ		109
Ā ʾ	q ٤	129	ý ٤	y ٤	80	A ʾ	Ā ʾ	25	Ā ʾ	A ʾ	84		l ٤	100
d ٤	ð ٤	119		A ʾ	73	Ā ʾ	A ʾ	23	d ٤	ð ٤	68	w ٤	h ʾ	97
	<SPC>	106	A ʾ		68	y ٤	j ٤	20		l ٤	67		n ٤	85

Table 6.3: The top 10 character edit transformations from raw to CODA in the entire MADAR CODA dataset across the five dialects. <SPC> indicates an explicit white space; whereas an empty cell indicates a *null* string.

they appear with different distributions. This suggests that a model making use of DID could learn dialect-specific preferences. At the same time, the shared phenomena can aid in learning dialect-independent general patterns.

6.4.2 Experiments

Evaluation Metrics Just like we did in Chapters 4 and 5, we use the MaxMatch (M^2) scorer (Dahlmeier and Ng, 2012) to assess the edits made by the system against the ‘gold standard’ edits in the target CODA, calculating precision (P), recall (R), and $F_{0.5}$ scores.

Baselines The first baseline simply copies the input sentences to the output (**Do Nothing**). This baseline highlights the level of similarity between the inputs and outputs. For the second baseline, we build a simple word-level lookup model to map input words to their CODAfied versions. We first obtain word-level alignments over all the training data from all the dialects (**Joint**) by using the algorithm introduced in §5.3.1. We then exploit the alignments to implement the lookup model as a bigram maximum likelihood

estimator: given an input word with its bigram surrounding context (w_i, w_{i-1}) , and a CODAified target word (y_i) , the model is built by computing $P(y_i|w_i, w_{i-1})$ over the training examples. During inference, we generate all possible alternatives for the given input word (w_i) . If the bigram context (w_i, w_{i-1}) was not observed in the training data, we backoff to a unigram context. If the input word was not observed during training, we pass it to the output as it is.

LLMs Similar to Chapters 4 and 5, we evaluate four LLMs: two commercial models (OpenAI’s GPT-3.5-turbo and GPT-4o (OpenAI et al., 2024)) and two open-source, Arabic-centric models (Jais-30B-Chat (Sengupta et al., 2023) and Fanar LLM (Team et al., 2025)). We use both English and Arabic prompts with 0-shot and 5-shot strategies. Additionally, we experiment with dialect-aware prompting (**DAP**), where the LLM is prompted with dialect-specific examples and an explicit dialect tag; in this setup, inference is performed separately for each dialect, and predictions are later aggregated for evaluation. Our prompts are presented in Tables C.1, C.2, and C.3 in Appendix C.1.

Seq2Seq Models We train both AraBART and AraT5 on all the dialects’ training data jointly with and without using DID information. We refer to this modeling setup as **Joint**. Moreover, to examine the effect of the joint dialectal training, we train five separate models, one for each dialect. During inference, we combine the separate models in an ensemble setup where we use the DID predictions for each sentence to select the appropriate model. We refer to this setup as **Ensemble**.

6.4.3 Results

Overall Results Table 6.4 shows the results on the Dev set. We present the LLMs results using their best setups, optimized for average $F_{0.5}$ across all datasets based on the prompt language and strategy (0-shot vs. 5-shot). Full results are provided in Table C.4 in Appendix C.2. Among the LLMs, GPT-4o achieves the best performance and benefits from dialect-aware prompting (**GPT-4o+DAP**), which improves precision, though not recall, resulting in a higher $F_{0.5}$. This suggests that explicitly specifying the dialect in the prompt can help LLMs better control their outputs. However, despite this improvement, none of the LLMs surpasses the simple MLE baseline in $F_{0.5}$, largely due to the MLE’s strong precision.

Among the Seq2Seq baselines, both AraBART and AraT5 demonstrate superior performance compared to the MLE model. In terms of training setups, **Joint** training outperforms **Ensemble** models for both AraBART and AraT5, with AraT5 being the better performer achieving 84.72 $F_{0.5}$.

When we train the AraBART **Joint** variants with DID control tokens, the performance increases compared to the AraBART **Joint** baseline, except when training with the **DA Phrase** DID control token. All the AraT5 **Joint** variants benefit from training with DID control tokens compared to the AraT5 baseline, with the **City** control token being the best performer with 85.80 $F_{0.5}$ (1.08 increase over the AraT5 baseline and statistically significant at $p < 0.05$). Statistical significance was determined using a two-sided approximate randomization test (Graham et al., 2014; Dror et al., 2018). Notably, the AraT5 variants outperform their AraBART counterparts across all experiments. We suspect this is due to the fact the data used to pretrain AraT5 consisted of a mix of MSA, DA, and CA compared to only MSA in the case of AraBART’s pretraining.

Model	Training	Control Token	P	R	F _{0.5}
Do Nothing	-	-	100.0	0	0
MLE	Joint	-	66.8	44.6	60.8
Fanar	-	-	24.5	28.8	25.2
Jais-30B-Chat	-	-	12.8	13.4	12.9
GPT-3.5-turbo	-	-	35.5	29.7	34.1
GPT-4o	-	-	53.7	54.4	53.8
GPT-4o + DAP	-	-	57.8	52.9	56.7
AraT5	Joint	-	86.8	77.4	84.7
		City	87.6	79.3	85.8
			<u>87.5</u>	<u>79.3</u>	<u>85.8</u>
		MSA Phrase	87.4	79.1	85.6
			87.4	79.1	85.6
		DA Phrase	87.3	78.6	85.4
			87.3	78.6	85.4
		Digit	87.4	79.0	85.6
			87.4	79.0	85.6
	Ensemble	-	85.7	72.8	82.7
AraBART	Joint	-	85.4	74.4	82.9
		City	85.7	74.4	83.2
			85.6	74.5	83.2
		MSA Phrase	85.5	74.5	83.0
			85.5	74.5	83.0
		DA Phrase	85.0	74.6	82.7
			85.0	74.6	82.7
		Digit	86.1	74.0	83.4
			86.1	74.0	83.4
	Ensemble	-	84.6	67.9	80.6
			84.4	68.5	80.7

Table 6.4: Dev set results for multiple systems. Results in grey indicate using gold DID labels (i.e., Oracle). Best results are in bold. Best oracle results are underlined.

Since AraT5 performed better than AraBART across all experiments, we present the results on the Test set using AraT5 and its variants in Table 6.5. Training AraT5 with the **DA Phrase** control token yields the best performance on the Test set with 86.29 F_{0.5} (1.06 increase over the AraT5 baseline and statistically significant at $p < 0.05$).

Model	Training	Control Token	P	R	F _{0.5}
AraT5	Joint	-	87.3	78.0	85.2
		City	88.0	78.3	85.9
		MSA Phrase	88.2	78.85	86.1
		DA Phrase	88.4	79.0	86.3
		Digit	87.7	78.3	85.6

Table 6.5: Results on the Test set.

Dialect	AraT5 (Baseline)			AraT5 + City		
	P	R	F _{0.5}	P	R	F _{0.5}
Beirut	86.1	79.7	84.71	89.3	82.4	87.8
Cairo	89.5	85.4	88.7	89.1	85.4	88.36
Doha	83.5	67.9	79.8	85.3	72.3	82.3
Rabat	85.2	72.7	82.4	86.4	76.0	84.1
Tunis	86.1	70.4	82.4	84.2	71.6	81.3

Table 6.6: Dialect-specific results of the best system (AraT5 + City) against the baseline (AraT5) on the Dev set.

DID Efficacy We estimate an oracle upper bound by using gold DID labels during inference on the Dev set (Table 6.4). We do not notice significant improvements across all variants compared to the models that use predicted DID labels. In some cases, using gold DID labels results in identical performance to models using predicted labels. This can be attributed to the robustness of our CODAfication models and the reliability of the DID system we are using, which achieves a high accuracy of 92.1% on the Dev set. Most of the prediction errors made by the DID system occur in sentences lacking distinctive cues that would allow clear assignment to a specific dialect. Therefore, these errors cannot be considered true errors, but rather stem from the MADAR dataset’s limitation of not having multi-dialectal labels. This is consistent with the findings of [Keleg and Magdy \(2023\)](#) where they manually analyzed the errors of a single-label DID system and found that $\sim 66\%$ of the errors are not true errors and could be resolved with multi-dialect labels.

Dialect	AraT5 (Baseline)			AraT5 + DA Phrase		
	P	R	F _{0.5}	P	R	F _{0.5}
Beirut	85.6	78.5	84.1	87.0	80.0	85.5
Cairo	89.7	83.4	88.33	89.6	84.4	88.5
Doha	85.9	70.8	82.4	89.8	73.3	85.9
Rabat	87.1	73.9	84.1	87.3	73.6	84.2
Tunis	86.7	75.6	84.3	89.2	76.6	86.4

Table 6.7: Dialect-specific results of the best system (AraT5 + DA Phrase) against the baseline (AraT5) on the Test set.

Dialect-Specific Results We present the dialect-specific results on the Dev and Test sets in Tables 6.6 and 6.7, respectively. Our best system on the Dev set, AraT5 trained with the **City** DID control token, improves over the AraT5 baseline for all dialects (with the largest increase seen for Beirut at 3.11 F_{0.5}), except for Cairo and Tunis, where the performance drop is attributed to decreased precision rather than recall. This suggests that our best system may be making unnecessary extra rewrites. On the Test set, our best system, AraT5 trained with the **DA Phrase** DID control token, improves over the AraT5 baseline across all dialects, with the largest increase for Doha at 3.58 F_{0.5}.

6.4.4 Error Analysis

To gain insights into the errors present in our best performing system on the Dev set, we conducted an error analysis on a sample of 100 cases, which accounted for 21% of the total 471 erroneous instances in the generated output. We classified these errors into specific categories, with results and examples provided in Table 6.8:

- **Non-CODA:** These are cases characterized by having plausible spontaneous spelling but incorrect CODA. This is the largest group of errors.
- **Hallucination and Related Hallucination:** Hallucinations refer to word rewrites that are implausible under any circumstance as a CODA correction or non-CODA

Category	%	Error	CODA
Non-CODA	46%	تحدثت <i>tHdst</i>	تحدثت <i>tHdθt</i>
Hallucination	19%	دقيقة. <i>dyqħ.</i>	دقيقة. <i>dqyqħ.</i>
Valid	13%	هامبورجر <i>hAmbwrjr</i>	هامبرجر <i>hAmbrjr</i>
Deletion	9%	أوصلة <i>AwSlħ</i>	أوصل له <i>AwSl lh</i>
Related Hallucination	9%	شرف <i>šrf</i>	الشرف <i>Alšrf</i>
Punctuation	4%	فاتتني <i>fAttny</i>	فاتتني <i>fAttny</i>

Table 6.8: Distribution of errors in the Dev set with one example per error type.

spelling. We distinguish cases that seem morphologically related to the input but are actually unrelated forms. We observe that 2/3 of the cases were largely unrelated to the reference.

- **Valid:** This category encompasses valid alternative spellings, particularly those associated with proper nouns and foreign words.
- **Deletion:** Deletions refer to omitted words. 55.6% of these are non-CODA spellings, e.g., a missed split (Table 6.8 example), while the rest are divided between gold errors and hallucinations.
- **Punctuation:** Punctuation generation errors.

The error analysis highlights that CODA issues constitute a significant portion of the remaining errors, potentially accounting for half of the cases between non-CODA words and deletions. Hallucinations, whether minor or severe, make up nearly a third of the errors. This suggests the need for more training data and improved models to address these problems. The presence of valid variants, which represent one-eighth of the errors, indicates the need to adopt a multi-reference approach for text normalization evaluation.

6.5 Summary

In this chapter, we presented a study on CODAfication, the task of normalizing Dialectal Arabic (DA) text into the Conventional Orthography for Dialectal Arabic (CODA). We benchmarked Arabic pretrained Seq2Seq models and LLMs on the task of CODAfication. We demonstrated that conditioning these models on dialect identification information significantly improves normalization performance. We reported results on a unique parallel corpus covering multiple Arabic dialects, focusing on five cities: Beirut, Cairo, Doha, Rabat, and Tunis.

Chapter 7

Text Editing

So far, this dissertation has introduced and explored three Arabic NLG tasks: gender rewriting, grammatical error correction, and dialectal text normalization. Across these tasks, our controlled NLG approaches have relied on an explicit identification step to extract linguistic traits, such as gender, error types, or dialect, which are then used to condition generation models. In this final chapter, we introduce a novel, unified text editing framework that reframes all three tasks as sequence tagging problems. Instead of generating text autoregressively, our approach assigns edit tags to input tokens, and applying these tags transforms the input into the desired output, ultimately combining both identification and generation into a single step. These edit tags are automatically derived from data, eliminating the need for hand-crafted or language-specific edit operations. We show that this text editing approach achieves state-of-the-art or highly competitive performance across all three tasks, surpassing prior models on most benchmarks, and offering substantial efficiency gains. Furthermore, we explore ensemble strategies and show that combining different models can lead to further performance improvements across tasks.

7.1 Introduction

Neural Seq2Seq models offer a powerful framework for translating source texts into target texts (§3.1.3). Since their introduction in machine translation (MT) (Sutskever et al., 2014), they have become the standard approach for nearly all conditional text generation tasks. Raffel et al. (2020) further demonstrated that even tasks not traditionally framed as sequence transduction problems can benefit from large-scale pretraining when reformulated in the Seq2Seq paradigm. However, for tasks like those explored in this dissertation, gender rewriting, grammatical error correction (GEC), and dialectal text normalization (CODAfication), the input and output often share substantial overlap. In such cases, using a full-sequence autoregressive model can be inefficient, as most tokens are simply copied from the input to the output.

A highly efficient and competitive alternative to Seq2Seq models is text editing, which frames generation tasks as sequence tagging problems. Text editing is tailored towards problems that require only small changes to the input. Rather than generating the target sentence autoregressively as a series of tokens, text editing models predict a sequence of edit operations that, when applied to the source sentence, yields the target sentence. However, most commonly used sequence tagging approaches require effort to design language-specific edit tag sets (Awasthi et al., 2019; Omelanchuk et al., 2020; Mesham et al., 2023). This limits the adaptability of current text editing approaches for morphologically rich languages like Arabic (Kwon et al., 2023), where the number of possible edits can be vast.

In this chapter, inspired by recent advancements in text editing (Awasthi et al., 2019; Malmi et al., 2019; Omelanchuk et al., 2020; Straka et al., 2021; Mesham et al., 2023), we introduce a novel text editing approach that eliminates the need for language-specific

edits. Instead, our method derives edit tags directly from data, making it more adaptable and scalable across different linguistic settings. We demonstrate the effectiveness of our approach on the NLG tasks we study in this dissertation: gender rewriting, GEC, and dialectal text normalization. Our models achieve SOTA or highly competitive performance across all three tasks, surpassing prior models on most benchmarks. In addition to strong performance, they are over six times faster than previously introduced models, making them more practical for real-world use. Finally, we show that ensemble strategies can further boost performance, as different models capture complementary strengths across tasks.

7.2 Background and Related Work

Text editing has gained increasing attention in recent years, especially for tasks like GEC, due to its efficiency in scenarios where most of the input text remains unchanged. At the heart of text editing is the process of *edit extraction*, which aligns input-output pairs to identify the minimal sequence of operations (e.g., keep, delete, insert, replace) required to transform the input into the target output. These edit operations are then assigned to individual tokens, enabling direct generation via tagging. Several approaches have implemented this framework with varying edit schemes and architectures. [Malmi et al. \(2019\)](#) introduced LaserTagger, which uses three primary operations (keep, delete, prepend) and a BERT encoder, optionally paired with an autoregressive decoder. They demonstrated its effectiveness on tasks such as sentence fusion, splitting, summarization, and GEC. Around the same time, [Awasthi et al. \(2019\)](#) proposed PIE, a model that uses a BERT encoder to predict edits (keep, delete, append, and morphological inflections) without a decoder, and applied it to GEC and OCR post-editing. Building on this work,

[Omelianchuk et al. \(2020\)](#) introduced GECToR, expanding the tag set to capture more complex grammatical transformations, such as verb forms and noun number. [Mesham et al. \(2023\)](#) extends GECToR further by introducing more general transformation edit tags. [Straka et al. \(2021\)](#) moved beyond word-level tagging by introducing a character-level text editing model operating at the subword level, making it especially suited for morphologically rich languages. They applied their model to English, Czech, German, and Russian GEC. Beyond token-level approaches, [Stahlberg and Kumar \(2020\)](#) proposed Seq2Edit, a span-based model leveraging a Seq2Seq architecture and applied it to text normalization, sentence fusion and splitting, simplification, and GEC. [Mallinson et al. \(2022\)](#) introduced EDIT5, a semi-autoregressive text editing model that combines the efficiency of tagging with the flexibility of generation, yielding strong results in low-resource settings for GEC and sentence fusion.

Despite growing interest in text editing, most existing approaches have been developed for English, with limited success in applying them to morphologically rich languages like Arabic. To our knowledge, [Kwon et al. \(2023\)](#) are the only ones to attempt Arabic text editing by adapting GECToR to Modern Standard Arabic (MSA) GEC. However, their system performed significantly worse than standard Seq2Seq models, likely due to the challenge of handcrafting edit tags that adequately capture Arabic’s morphological complexity. In this chapter, we introduce a novel text editing framework that addresses this limitation by deriving edit tags directly from data, eliminating the need for predefined, language-specific operations. Our approach is general, scalable, and particularly well-suited for morphologically rich languages. We demonstrate its effectiveness across the three Arabic NLG tasks explored in this dissertation: gender rewriting, GEC, and CODAfication.

7.3 Approach

We adopt a text editing approach for the three generation tasks explored in this dissertation: gender rewriting, GEC, and CODAfication. Rather than generating output from scratch, we frame each task as a sequence tagging problem. Formally, given an input sequence $x = x_1, x_2, \dots, x_n$, the goal is to assign a sequence of edit operations $e = e_1, e_2, \dots, e_n$; $e_i \in E$, where E is the edit vocabulary, such that applying edit e_i on the input token x_i at each position i would result in the output sequence $y = y_1, y_2, \dots, y_m$.

7.3.1 Edit Extraction

We begin by aligning input and output sentence pairs at the word level using a weighted Levenshtein edit distance (Levenshtein, 1966), which computes the minimum number of insertions, deletions, and replacements required to transform the input sentence into the output, with each edit affecting a single word. However, some transformations span multiple words. To handle such cases, we follow the approach introduced in §5.3.1 by extending the alignment process with an iterative algorithm that greedily merges or splits adjacent words to minimize the overall cumulative edit distance. After obtaining the word-level alignment, we apply the algorithm again, this time to each aligned word pair rather than the entire sentence, to determine character-level alignments. This process identifies the minimal character edits in terms of keep (**K**), delete (**D**), merge before (**M**), insert (**I**[**c**]), and replace (**R**[**c**]) that are needed to transform each erroneous word into its correction, where the inserted or replaced character (**c**) is explicitly specified.

Figure 7.1 presents an example from GEC of an aligned erroneous-corrected sentence pair along with the corresponding edits. For instance, in row b, the erroneous word الإهتمام *AlĀhtmAm* (word 1) requires the edit `KKR_[] KKKKK` (row c) which consists

Corrected	النفسية . <i>Alnfsyh</i>	الصحة <i>AlSHh</i>		سيما <i>symA</i>	ولا <i>wlA</i>	بالصحة <i>bAlSHh</i>	الاهتمام <i>AlAhtmAm</i>	يجب <i>yjb</i>	(a)				
	7	6	5	4	3	2	1	0					
Erroneous	النفسيه <i>Alnfsyh</i>	الصحه <i>AlSHh</i>	في <i>fȳ</i>	ولاسيما <i>wlAsymA</i>	لصحه <i>lSHh</i>	ب <i>b</i>	الإهتمام <i>AlĀhtmAm</i>	يجب <i>yjb</i>	(b)				
Word Edits	KKKKKKR_[ة]A_[.]	KKKKR_[ة]	DD	KKKI_[]KKKK	MI_[]KKKR_[ة]	K	KKR_[]KKKKK	KKK	(c)				
Word Edits (Compressed)	K*R_[ة]A_[.]	K*R_[ة]	D*	KKKI_[]K*	MI_[]K*R_[ة]	K*	KKR_[]K*	K*	(d)				
	7b	7a	6b	6a	5	4	3b	3a	2	1b	1a	0	
Tokenized Erroneous	## ##h	النفسي <i>Ālnfsy</i>	## ##h	الصح <i>AlSH</i>	في <i>fȳ</i>	ولاسيما <i>wlAsymA</i>	## ##h	لصح <i>lSH</i>	ب <i>b</i>	## ##htmAm	الإ <i>AlĀ</i>	يجب <i>yjb</i>	(e)
Subword Edits	R_[ة]A_[.]	KKKKKK	R_[ة]	KKKK	DD	KKKI_[]KKKK	R_[ة]	MI_[]KKK	K	KKKKK	KKR_[]	KKK	(f)
Subword Edits (Compressed)	R_[ة]A_[.]	K*	R_[ة]	K*	D*	KKKI_[]K*	R_[ة]	MI_[]K*	K*	K*	K*R_[]	K*	(g)

Figure 7.1: An example showing the different edit representations: words, words (compressed), subwords, and subwords (compressed). The edit operations are keep (**K/K***), delete (**D/D***), merge before (**M**), replace (**R_[c]**), insert (**I_[c]**), and append (**A_[c]**). Solid lines indicate word alignments between the corrected and erroneous sentences, while dotted lines denote erroneous subword boundaries. The sentence in the figure can be translated as “*Health, especially mental health, must be taken care of*”.

of eight character edits—one replacement and seven keeps—to produce its corrected form *AlAhtmAm*. Similarly, *lSHh* (row b, word 3), must be merged with the word before it, in addition to one insertion and one replacement (**MI_[]KKKR_[ة]**, row c).

In some cases, transformations require the insertion of entirely new characters, forming additional words in the input. Since we frame the task as a sequence tagging problem, we represent these insertions as appends (**A_[c]**) to existing edits rather than introducing standalone edits. This ensures that all edits, including word insertions, remain within the tagging framework. For example, to insert a period at the end of the erroneous sentence in Figure 7.1, we append the tag (**A_[.]**) to the edit of the final word (row c, word 7).

7.3.2 Edit Representation

The edit representation directly influences the size of the edit vocabulary ($|E|$), creating an important trade-off: a larger vocabulary offers more precise transformations but increases model complexity, whereas a smaller vocabulary enhances learning efficiency at the cost of expressiveness. Controlling $|E|$ is crucial to avoid the explosion of possible edits, which is particularly important when working with morphologically rich languages like Arabic. We explore four methods for controlling $|E|$ while maintaining sufficient coverage.

Edit Compression Once we obtain character-level edits for each word, we compress them into a more compact representation. The motivation behind this transformation is that while different words may undergo the same type of correction, their character-level edits can differ due to variations in word length. For example, in row b of Figure 7.1, both words 0 and 2 share a keep edit, yet they receive different edit labels because of their length differences (row c). To address this, we introduce a generalized notation for common edit patterns. Consecutive keep (**K**) and delete (**D**) operations are represented as **K*** and **D***, respectively. Similarly, consecutive insertions and appends are merged into a single operation, represented as **I_[c*]** for insertions and **A_[c*]** for appends, indicating the insertion or appending of multiple characters.

Since there are multiple ways to compress an edit sequence, we select the optimal strategy based on the frequency distribution of edit patterns in the training data. This approach ensures that the most common transformations are encoded in a way that balances expressiveness with efficiency, resulting in a more structured and learnable edit representation.

Input Unit Since Transformer-based models operate at the subword level, we project character-level edits onto subwords while maintaining their boundaries to ensure proper alignment. This not only ensures consistency with the model’s input representation but also helps reduce the edit vocabulary size. Our approach is inspired by the method of [Straka et al. \(2021\)](#), but it differs in several key aspects: (1) [Straka et al. \(2021\)](#) tokenize the erroneous and corrected sentence pairs before aligning them to extract the edits at the subword level. In contrast, our method extracts edits at the word level and then projects them onto subwords; (2) They limit the number of character-level edits per subword edit, while our approach imposes no such restrictions, allowing for broader coverage.

Figure 7.1 presents the subword-level edits in both their uncompressed (row f) and compressed (row g) forms. In the uncompressed subword-level edits, we observe that two subwords (3b and 6b in row e), which belong to different words, share the same edit ($R_{\text{[} \delta]}$). In the compressed representation, we notice that several subwords—such as 0, 1b, 2, 6a, and 7a—end up sharing the same edit (K^*).

Edit Segregation The MSA GEC datasets we report on exhibit high frequencies of punctuation errors, as detailed in §5.4.1: 40% in QALB-2014 and 15% in ZAEBUC training sets. To reduce the number of edits that the MSA GEC models must learn, we *segregate* punctuation edits from non-punctuation edits. This results in two versions of the data: one where only non-punctuation errors are tagged, and another where all non-punctuation errors are corrected, leaving only punctuation errors for the model to focus on. This separation is applied only to the MSA GEC datasets, not to APGC v2.0 (gender rewriting) or MADAR CODA (CODAfication). Additionally, this approach requires training two systems to be applied sequentially during inference: the first system fixes non-punctuation errors, while the second system addresses only punctuation errors.

Edit Pruning In tasks like GEC and CODAfication, morphologically rich languages such as Arabic often exhibit a long tail of infrequent edits in the training data. To improve the model’s learning ability, we analyze the distribution of edits in the training data for each task and prune those that occur less frequently than a threshold T , replacing them with the “keep” edit. This pruning is applied exclusively during training, enabling the model to focus on frequent and informative edits.

7.3.3 Edits Coverage

Input	Comp.	Subset	Prune	QALB-2014			QALB-2015			ZAEBUC		
				Edits	OOV%	F _{0.5}	Edits	OOV%	F _{0.5}	Edits	OOV%	F _{0.5}
Word	✗	All	-	16,221	1.00%	98.4	3,289	5.77%	92.2	1,097	2.94%	96.2
Subword	✗	All	-	9,060	0.36%	98.7	2,896	4.33%	92.0	905	1.85%	96.5
Word	✓	All	-	10,410	1.00%	98.4	2,425	5.77%	92.2	687	2.94%	96.2
Subword	✓	All	-	6,170	0.36%	98.7	2,125	4.33%	92.0	563	1.85%	96.5
Subword	✓	NoPnx	-	4,799	0.27%	98.8	1,906	3.93%	91.0	498	1.74%	96.2
Subword	✓	Pnx	-	160	0.01%	99.4	40	0.01%	99.3	23	0.06%	99.9
Subword	✓	All	10	683	0.75%	98.1	120	6.77%	88.0	58	3.71%	93.9
Subword	✓	All	20	442	1.02%	97.7	80	7.86%	86.4	35	4.67%	92.6
Subword	✓	All	30	329	1.24%	97.4	58	8.83%	84.6	27	5.26%	91.8
Subword	✓	NoPnx	10	520	0.56%	98.2	103	6.15%	86.0	52	3.39%	93.7
Subword	✓	NoPnx	20	335	0.75%	97.8	71	7.04%	84.1	30	4.31%	92.3
Subword	✓	NoPnx	30	250	0.92%	97.5	50	7.90%	81.7	22	4.90%	91.4
Subword	✓	Pnx	10	48	0.02%	99.4	15	0.11%	99.1	6	0.11%	99.9
Subword	✓	Pnx	20	35	0.05%	99.4	10	0.30%	98.9	6	0.11%	99.9
Subword	✓	Pnx	30	29	0.05%	99.3	9	0.34%	98.9	6	0.11%	99.9

Table 7.1: Edit statistics on QALB-2014, QALB-2015, and ZAEBUC. **Input** is the input unit representation (word or subword). **Comp.** indicates whether the edit is compressed. **Subset** specifies whether the edits capture all errors, punctuation-only errors (Pnx), or non-punctuation errors (NoPnx). **Edits** represents the total number of unique edits in the training set of each dataset. **OOV%** is the percentage of out-of-vocabulary edits (non-unique) in the Dev set of each dataset.

Table 7.1 presents edit statistics for the GEC datasets: QALB-2014, QALB-2015, and ZAEBUC. Table 7.2 shows the corresponding statistics for the CODAfication dataset,

Input	Comp.	Prune	MADAR CODA			APGCv2.0		
			Edits	OOV%	F _{0.5}	Edits	OOV%	F _{0.5}
Word	✗	-	1,228	1.52%	98.0	1,106	0.07%	99.3
Subword	✗	-	677	0.55%	98.1	844	0.05%	99.3
Word	✓	-	741	1.52%	98.0	588	0.08%	99.3
Subword	✓	-	454	0.55%	98.1	514	0.07%	99.3
Subword	✓	10	84	1.33%	96.2	127	0.12%	98.4
Subword	✓	20	52	2.02%	94.1	92	0.17%	97.7
Subword	✓	30	45	2.28%	93.4	74	0.21%	97.0

Table 7.2: Edit statistics on MADAR CODA and APGCv2.0. **Input** is the input unit representation (word or subword). **Comp.** indicates whether the edit is compressed. **Edits** represents the total number of unique edits in the training set of each dataset. **OOV%** is the percentage of out-of-vocabulary edits (non-unique) in the Dev set of each dataset.

MADAR CODA, and the gender rewriting parallel corpus, APGC v2.0. These tables illustrate the impact of our strategies to reduce the edit vocabulary size $|E|$ on edit coverage and upper-bound (oracle) performance on the development (Dev) sets. Edit coverage measures the proportion of training edits found in the Dev sets, while oracle performance is evaluated using the MaxMatch (M^2) scorer (Dahlmeier and Ng, 2012) $F_{0.5}$. We use AraBERTv02 (Antoun et al., 2020) for subword tokenization, as it yielded the best results among our tested models (more details in §7.4.3).

For the GEC datasets, and specifically QALB-2014, switching from word-level to subword-level edits reduces unique training edits by 44% (16,221 to 9,060) and lowers the Dev set OOV rate from 1% to 0.4%, yielding a 0.3-point $F_{0.5}$ gain. Edit compression further reduces unique edits while preserving OOV% and oracle performance. Segregating punctuation (Pnx) from non-punctuation (NoPnx) edits reduces combined training edits (4,799+160 from 6,170). However, NoPnx results are not directly comparable since punctuations are explicitly removed before the evaluation. Pnx $F_{0.5}$ scores are higher as they are evaluated on a Dev set with non-punctuation errors already corrected,

making the test easier. To assess the impact of pruning, we apply frequency thresholds of 10, 20, and 30 to remove low-frequency edits. As expected, pruning reduces the number of unique training edits and increases the OOV% in the Dev set, yet $F_{0.5}$ remains largely unaffected. This suggests that the majority of the 6,170 compressed subword edits occur infrequently and contribute little to the model’s upper-bound performance. A similar trend is observed for both Pnx and NoPnx edits, reinforcing the idea that many low-frequency edits can be pruned without degrading oracle performance. Similar conclusions hold for the QALB-2015 and ZAEBUC datasets.

For both MADAR CODA and APGC v2.0, we observe similar trends to those found in the GEC datasets. Switching from word-level to subword-level edits significantly reduces the number of unique training edits while also lowering the OOV rate on the Dev sets. However, unlike GEC, the shift to subword-level edits does not lead to improvements in oracle $F_{0.5}$ scores, suggesting that the benefits of subword modeling are primarily in vocabulary compression rather than correction potential. Applying edit compression further reduces the number of edits without affecting OOV% or $F_{0.5}$.

7.4 Experimental Setup

7.4.1 Data

We use the same datasets introduced in earlier chapters for each task. For gender rewriting, we use the Arabic Parallel Gender Corpus v2.0 (APGC v2.0) (§4.3). For GEC, we use the QALB-2014, QALB-2015, and ZAEBUC datasets (§5.4.1). For CODAfication, we use the MADAR CODA corpus (§6.4.1).

7.4.2 Experiments

Baselines For GEC, we compare our text editing approach to the best-performing vanilla Seq2Seq models introduced in Chapter 5, as well as to enhanced variants that incorporate morphological preprocessing and are conditioned on grammatical error detection (GED) predictions (§5.3.2). A similar comparison is made for CODAfication, where we compare against the top vanilla Seq2Seq model from Chapter 6, along with a variant conditioned on dialect identification (§6.3). For gender rewriting, we compare against the joint and the best-performing multi-step rewriting systems introduced in Chapter 4 (§4.4). For all tasks, we also report results from the strongest LLM setup.

Edit Taggers To investigate the impact of the edit representation design on performance (§7.3.2), we build several edit taggers with different configurations. For word-level tagging, we use the representation of the first subword of each word and pass it through the subsequent layers. For subword-level tagging, we use the representation of each subword individually. Several Arabic pretrained transformer encoders based on BERT (Devlin et al., 2019) have been developed (Antoun et al., 2020; Abdul-Mageed et al., 2021a; Inoue et al., 2021; Ghaddar et al., 2022). We select the three best-performing Arabic BERT models, as identified by Inoue et al. (2021) across various sentence and token classification tasks: AraBERTv02 (Antoun et al., 2020), ARBERTv2 (Abdul-Mageed et al., 2021a), and CAMeLBERT-MSA (Inoue et al., 2021).

Ensembles We construct majority vote ensemble models by aggregating the outputs of multiple GEC systems. This is enabled by our edit extraction algorithm (§7.3.1), which allows us to align and extract edits from models with different architectures. Using this algorithm, we first align each model’s output with the input text, extract the proposed

	QALB-2014			QALB-2015			ZAEBUC		
	P	R	F _{0.5}	P	R	F _{0.5}	P	R	F _{0.5}
AraBART	83.2	64.9	78.7	68.6	42.6	61.2	87.3	70.6	83.4
AraT5+Morph+GED ⁴³	83.1	67.9	79.6	68.2	46.6	62.4	87.6	73.9	84.5
AraBART+Morph+GED ⁴³	83.4	66.3	79.3	68.2	46.6	62.4	87.3	73.6	84.2
AraBART+Morph+GED ¹³	<u>83.9</u>	65.7	79.5	68.0	46.6	62.3	87.6	73.9	84.5
GPT-4o	80.7	65.7	77.2	70.6	<u>49.2</u>	<u>65.0</u>	86.5	<u>76.8</u>	84.3
GPT-4o+GED	82.1	62.2	77.2	<u>74.4</u>	41.2	64.0	<u>90.4</u>	72.3	<u>86.1</u>
SWEET	81.8	68.8	78.8	64.8	40.6	57.9	85.8	72.3	82.7
SWEET ²	81.9	<u>70.4</u>	79.3	65.9	43.3	59.7	85.8	73.3	83.0
SWEET ² _{NoPnx} + SWEET _{Pnx}	83.7	68.8	<u>80.3</u>	69.6	37.4	59.4	86.7	73.9	83.8
3-Ensemble	84.9	68.8	81.1	74.0	39.9	63.2	89.6	72.8	85.6
+GPT-4o	89.1	61.6	81.8	81.2	33.1	62.9	93.3	68.3	86.9
+GPT-4o+GED	89.4	61.0	81.8	81.3	32.4	62.5	93.3	67.7	86.7

Table 7.3: MSA GEC results on the Dev sets of QALB-2014, QALB-2015, and ZAEBUC. Best non-ensemble results are underlined. The best overall results are in bold.

edits, and then determine the final edit sequence through majority voting. Following Tarnavskiy et al. (2022), we retain an edit only if at least $k - 1$ models out of k models predict it; otherwise, we leave the input unchanged.

7.4.3 Results

Table 7.3 presents development set results for GEC, while Table 7.4 shows corresponding results for CODAfication and gender rewriting. Full edit tagging results on the Dev sets, including experiments with different edit design choices using CAMELBERT-MSA, AraBERTv02, and ARBERTv2, are presented in Tables D.2 and D.1 in Appendix D.1. AraBERTv02 consistently achieves the best performance. For GEC and CODAfication, using subword-level edits alongside compression and pruning leads to improved results. The most effective pruning threshold is 10 for QALB-2014, QALB-2015, and MADAR CODA, and 30 for ZAEBUC. In the case of gender rewriting, AraBERTv02 also performs best. Although subword-level edits do not outperform word-level edits in this task,

	MADAR CODA			APGCv 2.0		
	P	R	F _{0.5}	P	R	F _{0.5}
AraT5	86.8	77.4	84.7	-	-	-
AraT5+City	87.6	<u>79.3</u>	85.8	-	-	-
Joint	-	-	-	79.0	79.8	79.1
Multi-Step	-	-	-	88.7	86.8	88.3
GPT-4o	53.7	54.4	53.8	49.2	74.6	52.8
GPT-4o+DAP	57.8	52.9	56.7	-	-	-
GPT-4o+GID	-	-	-	77.1	77.7	77.2
SWEET	<u>89.1</u>	75.5	<u>86.0</u>	<u>89.7</u>	87.3	<u>89.2</u>
SWEET ²	87.5	73.5	84.3	<u>89.7</u>	<u>87.4</u>	<u>89.2</u>
3-Ensemble	91.7	77.4	88.4	91.0	89.1	90.6
+GPT-4o	93.8	72.5	88.6	92.1	86.5	90.9
+GPT-4o+DAP	93.8	72.3	88.6	-	-	-
+GPT-4o+GID	-	-	-	92.2	86.3	90.9

Table 7.4: CODAfication and gender rewriting results on the Dev sets of MADAR CODA and APGC v2.0. Best non-ensemble results are underlined, best overall results are in bold.

applying compression and pruning still proves beneficial, with a pruning threshold of 10 yielding the best results. The optimal setup for each dataset is presented in Tables 7.3 and 7.4. We henceforth refer to this system as SWEET (Subword Edit Error Tagger).

SWEET achieves an F_{0.5} of 78.8 on QALB-2014, 86.0 on MADAR CODA, and 89.2 on the AGPCv2.0, outperforming AraBART on QALB-2014 and setting a new SOTA on MADAR CODA and APGC v2.0. On ZAEBUC and QALB-2015, it scores 82.7 and 57.9 F_{0.5}, respectively, trailing behind AraBART. Consistent with previous work on text editing (Awasthi et al., 2019; Omelianchuk et al., 2020), we find that iterative correction improves GEC up to two iterations (SWEET²), achieving 79.3 F_{0.5} on QALB-2014, 59.7 on QALB-2015, and 83.0 on ZAEBUC, with the highest overall recall on QALB-2014 (70.4). However, it degrades CODAfication and does not help gender rewriting.

Separating non-punctuation edits (SWEET_{NoPnx}) from punctuation edits (SWEET_{Pnx}) improves GEC performance on QALB-2014 and ZAEBUC but not QALB-2015. The best

setup applies these systems in sequence: two iterations of non-punctuation correction followed by one iteration of punctuation correction ($\text{SWEET}_{\text{NoPnx}}^2 + \text{SWEET}_{\text{Pnx}}$). This setup achieves the highest $F_{0.5}$ score among text editing models, setting a new SOTA on QALB-2014 with 80.3 (driven by a precision improvement of 83.7). Its performance on ZAEBUC leads other edit tagging techniques but trails behind GPT-4o.

For our ensemble models (3-Ensemble), we combine the outputs of the top three non-LLM models per dataset. For QALB-2014, QALB-2015, and ZAEBUC, the ensemble includes the best Seq2Seq model incorporating morphological preprocessing and GED predictions, SWEET^2 , and the cascaded setup $\text{SWEET}_{\text{NoPnx}}^2 + \text{SWEET}_{\text{Pnx}}$. For MADAR CODA, we combine AraT5+City, SWEET, and the second-best SWEET model using CAMELBERT-MSA (see Table D.2 in Appendix D.1). For APGC v2.0, the ensemble includes the Multi-Step model, SWEET, and the second-best SWEET variant using AraBERTv02 with word-level edits and no pruning (Table D.1, Appendix D.1).

The 3-Ensembles outperform single models, achieving SOTA results across all datasets except QALB-2015, primarily by improving precision at the cost of recall. Adding GPT-4o to the ensemble (3-Ensemble+GPT-4o) further boosts performance, reaching $F_{0.5}$ scores of 81.8 on QALB-2014, 86.9 on ZAEBUC, 88.6 on MADAR CODA, and 90.9 on APGC v2.0. Notably, incorporating GPT-4o outputs generated using control prompting, such as GPT-4o+GED for GEC, GPT-4o+DAP for CODAfication, or GPT-4o+GID for gender rewriting, offers no gains over the standard 3-Ensemble+GPT-4o.

Test Results: Table 7.5 presents test results for GEC, while Table 7.6 reports corresponding results for CODAfication and gender rewriting, using the best setups identified on the Dev sets. For GEC, the cascaded setup $\text{SWEET}_{\text{NoPnx}}^2 + \text{SWEET}_{\text{Pnx}}$ sets a new SOTA on QALB-2014 with 80.5 $F_{0.5}$, outperforming all Seq2Seq models introduced in Chapter 5.

	QALB-2014			QALB-2015-L1			QALB-2015-L2			ZAEBUC		
	P	R	F _{0.5}	P	R	F _{0.5}	P	R	F _{0.5}	P	R	F _{0.5}
AraBART	84.0	64.7	79.3	82.0	71.7	79.7	69.6	43.5	62.1	86.0	71.6	82.7
AraBART+GED ⁴³	84.2	65.4	79.6	81.2	72.4	79.3	69.0	45.4	62.5	85.4	72.6	82.5
AraBART+Morph+GED ⁴³	83.9	65.7	79.5	82.6	72.1	80.3	67.6	45.2	61.5	85.4	73.7	82.7
AraBART+GED ¹³	84.1	65.0	79.4	81.5	72.7	79.5	69.3	44.9	62.5	85.9	73.4	83.1
GPT-4o+GED	82.9	62.0	77.7	<u>82.8</u>	71.1	80.2	<u>75.2</u>	42.5	65.1	<u>89.0</u>	73.2	<u>85.3</u>
SWEET ²	82.6	69.5	79.6	80.0	74.3	78.8	66.1	42.5	59.5	85.5	74.4	83.0
SWEET ² _{NoPhx} + SWEET _{Phx}	<u>84.5</u>	67.7	<u>80.5</u>	82.2	73.6	<u>80.3</u>	70.0	37.3	59.6	85.7	74.1	83.1
3-Ensemble	85.7	67.4	81.3	83.7	73.3	81.3	73.3	39.2	62.4	89.7	73.7	85.9
+ GPT-4o+GED	89.9	59.7	81.6	88.5	66.2	82.9	81.5	31.9	62.2	93.5	69.5	87.5

Table 7.5: MSA GEC results on the Test sets of QALB-2014, QALB-2015 (L1), QALB-2015 (L2), and ZAEBUC. Best non-ensemble results are underlined. The best overall results are in bold.

This improvement is statistically significant at $p < 0.05$, based on a two-sided approximate randomization test, compared to the best Seq2Seq baseline (AraBART+GED⁴³). On QALB-2015-L1, the same setup matches the performance of AraBART+Morph+GED⁴³ with an F_{0.5} of 80.3, and on ZAEBUC, it achieves 83.1, on par with AraBART+GED¹³. However, on QALB-2015, it underperforms relative to all Seq2Seq baselines.

For CODAfication and gender rewriting, SWEET achieves SOTA performance, reaching 86.5 F_{0.5} on MADAR CODA and 89.8 on APGC v2.0. Both results are statistically significant improvements over the strongest previously introduced systems: AraT5+DA Phrase for CODAfication and the Multi-Step model for gender rewriting.

Our ensemble models further enhance performance across all datasets, achieving F_{0.5} scores of 81.6 on QALB-2014, 82.9 on QALB-2015-L1, 87.5 on ZAEBUC, 88.9 on MADAR CODA, and 91.3 on APGC v2.0. Notably, we incorporate outputs from the best performing GPT-4o control prompting setups: GPT-4o with GED for GEC, GPT-4o with DAP for CODAfication, and GPT-4o with GID for gender rewriting. This addition improves performance across all GEC datasets except QALB-2015-L2 but yields no further gains for CODAfication.

	MADAR CODA			APGC v2.0		
	P	R	F _{0.5}	P	R	F _{0.5}
AraT5	87.3	78.0	85.2	-	-	-
AraT5+DA Phrase	88.4	<u>79.0</u>	86.3	-	-	-
Joint	-	-	-	79.3	80.4	79.5
Multi-Step	-	-	-	88.9	86.7	88.4
GPT-4o+DAP	59.3	55.1	58.4	-	-	-
GPT-4o+GID	-	-	-	76.4	76.5	76.4
SWEET	<u>89.4</u>	76.6	<u>86.5</u>	<u>90.3</u>	<u>87.6</u>	<u>89.8</u>
3-Ensemble	92.2	77.7	88.9	91.4	89.1	90.9
+ GPT-4o+DAP	94.0	73.1	88.9	-	-	-
+ GPT-4o+GID	-	-	-	92.9	85.2	91.3

Table 7.6: CODAfication and gender rewriting results on MADAR CODA and APGC v2.0 Test sets. Best non-ensemble results are underlined, best overall results are in bold.

7.4.4 Runtime Performance

Table 7.7 compares our text editing models to the Seq2Seq models from Chapter 5 in terms of model size, initialization time, and inference runtime. Initialization and inference times were averaged over 10 runs on the QALB-2014 Dev set using a single A100 GPU with a batch size of 32. The reported values for AraT5+Morph+GED⁴³ reflect the combined size, initialization, and inference times of all its components. Our SWEET model is 4x smaller than AraT5+Morph+GED⁴³, while the cascaded system SWEET²_{NoPnx} + SWEET_{Pnx} is about half the AraT5+Morph+GED⁴³ model size. In terms of speed, SWEET initializes 19x faster than AraT5+Morph+GED⁴³, while the cascaded system achieves a 9x initialization speedup. For inference, SWEET is also 19x faster, SWEET² is 9x faster, and the cascaded setup is 6x faster. Compared to the vanilla AraBART model, SWEET runs 6x faster, SWEET² is 3x faster, and the cascaded system is twice as fast. Although the 3-Ensemble setup achieves the best performance, it is the largest in model size and the slowest overall.

	Params	Time	
		Init.	Run
AraBART	139M	1.7	70.7
AraT5+Morph+GED ⁴³	502M	24.7	218.5
SWEET	135M	1.3	11.6
SWEET ²	135M	1.3	23.2
SWEET ² _{NoPnx} + SWEET _{Pnx}	270M	2.7	34.8
3-Ensemble	908M	28.7	276.4

Table 7.7: Number of parameters (Params.), initialization time (Init.), and runtime for different models on the Dev set of QALB-2014. Init. and runtime are in seconds and averaged over 10 runs on a single A100 GPU using a batch size of 32.

7.4.5 Error Analysis

GEC: Table 7.8 presents specific error type performance of the best Seq2Seq model (AraBART+Morph+GED¹³), our best SWEET system (SWEET²_{NoPnx} + SWEET_{Pnx}), and the best ensemble (3-Ensemble+GPT-4o). SWEET outperforms the Seq2Seq baseline on most error types in QALB-2014 and ZAEBUC, achieving macro-average improvements of 3.6 and 4.5 $F_{0.5}$, respectively. However, the Seq2Seq baseline remains the best-performing model on QALB-2015, both on macro-average $F_{0.5}$ and most error types. Notably, on ZAEBUC, although SWEET improves error-type performance over the baseline, it does not surpass it in overall GEC performance (Table 7.3). This is primarily due to the skewed error distribution in ZAEBUC, where O/C/Merge-related errors dominate, and both models perform comparably on these types. The error types where SWEET excels are relatively infrequent (Table 5.2). The ensemble model yields modest gains over SWEET on QALB-2014 and QALB-2015, but not on ZAEBUC.

CODification: In Table 7.9, we revisit the 100 erroneous cases sampled in Chapter 6 (§6.4.4). We replicate our manual error analysis over the outputs of our best SWEET model and the best ensemble system (3-Ensemble+GPT-4o), comparing them to the baseline

	QALB-2014			QALB-2015			ZAEBUC		
	Seq2Seq	SWEET	Ensemble	Seq2Seq	SWEET	Ensemble	Seq2Seq	SWEET	Ensemble
Delete	40.8	44.3	45.2	45.3	<u>48.6</u>	64.6	51.9	63.6	62.5
Merge-B	93.0	<u>93.7</u>	93.8	86.6	<u>87.1</u>	90.4	96.7	96.9	96.6
Merge-I	93.0	<u>93.5</u>	93.6	86.4	<u>87.0</u>	90.5	96.7	96.9	96.6
M	27.6	33.6	28.6	40.8	18.8	21.6	48.6	50.0	41.7
M+O	37.7	61.0	60.6	15.2	0.0	0.0	55.6	100.0	0.0
O	94.3	94.5	94.4	80.2	72.5	76.5	94.4	94.4	94.1
O+X	73.9	<u>81.5</u>	83.3	0.0	33.3	0.0	0.0	0.0	0.0
P	77.4	75.6	76.8	63.7	65.0	64.8	62.8	71.9	70.4
S	44.5	<u>57.1</u>	57.4	34.1	27.3	26.4	40.4	47.6	46.9
X	61.1	<u>61.4</u>	62.4	<u>63.9</u>	60.2	65.0	72.9	74.1	74.1
Split	<u>87.1</u>	83.8	87.6	78.9	70.7	76.7	88.2	<u>90.0</u>	95.2
UNK	57.2	55.0	56.0	35.2	29.6	29.6	63.1	44.6	47.6
C	96.8	96.4	94.7	91.4	88.4	87.1	96.1	96.0	93.9
Macro Avg	68.0	<u>71.6</u>	71.9	55.5	53.0	53.3	66.7	71.2	63.1

Table 7.8: Error type performance ($F_{0.5}$) on the Dev sets of QALB-2014, QALB-2015, and ZAEBUC for the best baseline Seq2Seq model, our best SWEET system, and the best ensemble. Best non-ensemble scores are underlined, best overall scores are in bold.

Seq2Seq model (AraT5+City). Errors are categorized using the taxonomy introduced in §6.4.4. These 100 cases represent 21% of the 471 total erroneous predictions made by the Seq2Seq model. SWEET reduces the number of errors by 27%, correctly rewriting 27 out of the 100 cases, whereas the ensemble reduces the number of errors by 16%. Both models show reductions across all error categories when compared to the Seq2Seq baseline, with notable improvements in non-coda and hallucination errors. These gains underscore the enhanced controllability offered by the text editing approach over traditional Seq2Seq models.

Gender Rewriting: Table 7.10 presents the distribution of erroneous sentences in the outputs of the best Multi-Step model, the best SWEET model, and the best ensemble (3-Ensemble+GPT-4o). The SWEET model reduces the proportion of erroneous sentences across all four target corpora, except for 1F/2M, with the largest reduction observed for 1F/2F (0.4%). The ensemble model further lowers the error rates across all target

Error Type	Seq2Seq	SWEET		Ensemble	
Non-Coda	46%	42%	(-33%)	46%	(-15%)
Hallucination	19%	22%	(-16%)	20%	(-11%)
Valid	13%	16%	(-8%)	14%	(-8%)
Deletion	9%	11%	(-11%)	10%	(-11%)
Related Hallucination	9%	7%	(-44%)	10%	(-11%)
Punctuation	4%	1%	(-75%)	0%	(-100%)
Correct	0	37%		20%	

Table 7.9: Distribution of errors over a sample of 100 cases from the MADAR CODA Dev set for the best Seq2Seq model, the best SWEET model, and the best ensemble. Percentages in parentheses indicate the reduction in errors relative to the Seq2Seq model.

Target	Multi-Step	SWEET	Ensemble	SWEET			Ensemble		
				Fixed	Shared	Error	Fixed	Shared	Error
1M/2M	3.8%	3.7%	3.3%	1.4%	2.4%	1.3%	1.1%	2.7%	0.6%
1F/2M	4.0%	4.1%	4.4%	1.4%	2.5%	1.6%	0.7%	3.2%	1.2%
1M/2F	7.1%	7.0%	7.0%	2.2%	4.8%	2.1%	1.4%	5.6%	1.4%
1F/2F	6.6%	6.2%	5.7%	2.1%	4.5%	1.7%	1.7%	4.9%	0.8%

Table 7.10: Distribution of erroneous sentences from the APGC v2.0 Dev set across the four target corpora for the Multi-Step model, the best SWEET model, and the top ensemble. The second half of the table reports the distribution of **Fixed**, **Shared**, and newly introduced **Errors** by the SWEET and ensemble models.

corpora except 1F/2M, where it increases the error rate by 0.4%. The most substantial improvement is again seen in the 1F/2F setting, with a 0.9% reduction compared to the Multi-Step model. Table 7.10 also reports the distribution of fixed, shared, and newly introduced errors by the SWEET and ensemble models relative to the Multi-Step baseline. Across all four target corpora, both models achieve notable gains by correcting a substantial portion of the baseline’s errors. While SWEET fixes more errors than the ensemble, it also introduces more new errors. In contrast, the ensemble introduces fewer errors and retains a higher proportion of shared errors with the baseline, suggesting a more balanced trade-off that improves over the baseline with fewer regressions.

Error Type	Multi-Step				SWEET				Ensemble			
	1M/2M	1F/2M	1M/2F	1F/2F	1M/2M	1F/2M	1M/2F	1F/2F	1M/2M	1F/2M	1M/2F	1F/2F
No Change	25%	32%	52%	37%	17%	20%	37%	26%	18%	25%	44%	30%
Rewrite	43%	47%	35%	40%	16%	31%	22%	14%	18%	36%	23%	16%
Gold	32%	21%	13%	23%	26%	17%	13%	21%	27%	19%	13%	21%
Correct	0	0	0	0	41%	32%	28%	39%	37%	20%	20%	33%

Table 7.11: Error type distribution for a sample of 100 Dev sentences from APGC v2.0, comparing the Multi-Step model, the best-performing SWEET model, and the top ensemble system.

Table 7.11 presents a manual error analysis of 100 erroneous Dev sentences from APGC v2.0 for each of the four target corpora across the three systems. We categorized errors into: (1) **No Change** errors, where the system failed to make a necessary change (false negatives), (2) **Rewrite** errors, where incorrect changes were made (false positives), and (3) **Gold** errors, where the output was correct but did not match the reference due to annotation inconsistencies. We began our analysis with the errors made by the Multi-Step model and then examined how the SWEET and ensemble systems performed on the same set.

The Multi-Step model’s errors are dominated by Rewrite errors across all contexts, indicating a tendency to over-correct. The SWEET model reduces errors across all target corpora, with the most substantial improvement in the 1M/2M setting, cutting errors by 41% through reductions in both No Change and Rewrite categories. The ensemble model also improves over the Multi-Step baseline but underperforms compared to SWEET.

7.5 Summary

In this chapter, we introduced a text editing framework that reframes gender rewriting, GEC, and CODAfication as sequence tagging tasks. Unlike traditional Seq2Seq models, our approach assigns edit operations directly to input tokens, effectively combining identification and generation in a single step. These edit tags are automatically derived from data, removing the need for hand-crafted or language-specific operations and enabling adaptability to morphologically rich languages like Arabic. We demonstrated that our models achieve SOTA or highly competitive results across all tasks while being substantially more efficient, over six times faster than previously introduced approaches. Additionally, we showed that ensembling diverse models further boosts performance.

Chapter 8

Summary and Conclusions

In this dissertation, we explored three NLG tasks for Arabic: gender rewriting, grammatical error correction, and dialectal text normalization. For each task, we introduced controllable modeling approaches that incorporate linguistic traits to enhance the controllability of NLG systems. Our models achieve state-of-the-art results across all tasks. In this final chapter, we highlight the main findings and contributions of the dissertation and discuss promising directions for future research.

8.1 Arabic Gender Rewriting

We first presented the task of Arabic gender rewriting as generating alternatives of a given Arabic sentence to match different target user gender contexts: a *male speaker* with a *male listener*, a *female speaker* with a *male listener*, a *male speaker* with a *female listener*, and a *female speaker* with a *female listener*. This requires changing the grammatical gender (masculine or feminine) of certain words referring to the users (speaker/first person and listener/second person). Formally, given an input X that combines both the input Arabic sentence and the target gender preferences, and a sequence of word-level

gender labels G corresponding to the input Arabic sentence, the goal is to generate a rewritten version Y that matches the user specified gender preferences:

$$P(Y|X, G) = \prod_{t=1}^n P(y_t|y_1, \dots, y_{t-1}, X, G)$$

To facilitate this task, we introduced the Arabic Parallel Gender Corpus v2.0, a new dataset designed for gender identification and rewriting. We also developed and benchmarked various gender rewriting systems and LLMs on this corpus and demonstrated their effectiveness.

The Arabic Parallel Gender Corpus We presented the Arabic Parallel Gender Corpus v2.0 (APGC v2.0), which extends APGC v1.0 by adding second person targets as well as increasing the total number of sentences over 6.5 times, reaching over 590K words. The new corpus consists of multiple parallel components: four combinations of first- and second-person grammatical gender (masculine and feminine), English source sentences, and English-to-Arabic machine translation outputs. We selected English-Arabic sentence pairs from the OpenSubtitles 2018 dataset for manual annotation. The annotation process involved gender identification followed by rewriting to generate all parallel gender alternatives corresponding to the target user gender contexts studied. In total, the corpus includes 80,326 parallel sentences (596,799 words), with gender annotations provided at both the sentence and word levels.

Gender Rewriting Approaches We introduced two gender rewriting approaches: joint Seq2Seq models, which perform sentence-level rewriting in a single pass without an explicit identification step, and multi-step word-level models, which decompose the task into separate identification and rewriting stages for finer control. We also evaluated

four LLMs: two commercial (OpenAI’s GPT-3.5-turbo and GPT-4o) and two open-source, Arabic-centric models (Jais-30B-Chat and the recently introduced Fanar LLM). Our results showed that the multi-step approach outperforms joint models due to its controllability. Furthermore, we demonstrated that incorporating word-level gender identification when prompting LLMs improves their performance on gender rewriting.

We further demonstrated how gender rewriting can be used to personalize the output of user-unaware machine translation systems through post-editing. To facilitate real-world use, we introduced the User-Aware Arabic Gender Rewriter, a web-based application that integrates our best-performing model, enabling seamless user interaction.

Lastly, we summarized our findings from the Shared Task on Arabic Gender Rewriting, which we organized as part of the Seventh Arabic NLP Workshop.

8.2 Arabic Grammatical Error Detection and Correction

We presented a comprehensive study on grammatical error correction (GEC) for Modern Standard Arabic (MSA) and formalized the task of Arabic grammatical error detection (GED). We demonstrated that conditioning models on GED predictions enhances GEC performance. Formally, given an erroneous Arabic sentence X and its corresponding sequence of error types E , the goal is to generate a corrected sentence Y :

$$P(Y|X, E) = \prod_{t=1}^n P(y_t|y_1, \dots, y_{t-1}, X, E)$$

Additionally, we explored the impact of contextual morphological preprocessing on model performance. Our systems achieved state-of-the-art results on two MSA GEC datasets (L1 and L2) and established a strong benchmark on a newly introduced L1 dataset.

Arabic Grammatical Error Detection (GED) GED has received little attention in Arabic NLP due to the absence of manually annotated corpora and standardized error type frameworks. To address this, we proposed an automatic method that extracts edits from parallel GEC corpora and assigns error types using an Arabic error type annotation tool. We framed GED as a multi-class sequence labeling task and evaluated models under three levels of granularity: 43-Class, 13-Class, and 2-Class (binary).

Arabic Grammatical Error Correction (GEC) We were the first to benchmark pre-trained Seq2Seq models on Arabic GEC, experimenting with AraBART and AraT5. We also benchmarked both open-source Arabic-centric and commercial LLMs including. We investigated whether conditioning these models on GED predictions could improve correction performance and whether contextual morphological preprocessing offers additional benefits. Our findings show that GED predictions consistently enhance Seq2Seq model performance, and morphological preprocessing further improves results in some cases. Moreover, we found that commercial LLMs like GPT-4o perform remarkably well on certain GEC datasets. However, we emphasized that these results should be interpreted with caution due to the models’ closed-source nature and the lack of transparency regarding their training data, raising concerns about reproducibility and potential data contamination.

8.3 Dialectal Text Normalization

We presented the task of dialectal text normalization, known as CODAfication, which involves converting Dialectal Arabic (DA) into the Conventional Orthography for Dialectal Arabic (CODA). DA represents the non-standard variety of Arabic and comprises multiple regional dialects, such as Egyptian, North African, Levantine, and Gulf Arabic,

that differ significantly from both MSA and from one another in terms of phonology, morphology, and lexicon. While DA is primarily spoken, it is increasingly used in written form on social media. However, the lack of standardized spelling leads to noisy, highly variable text, creating challenges for NLP systems due to increased data sparsity. To address this issue, CODA was proposed as a unified spelling convention. Nevertheless, CODA has largely been treated as a secondary task, supporting applications like morphological disambiguation, diacritization, and lemmatization, rather than being studied as a primary NLG task.

In this dissertation, we framed CODAfication as a standalone generation task. We worked with the MADAR CODA Corpus, a unique parallel dataset covering five major dialects spoken in Beirut, Cairo, Doha, Rabat, and Tunis. We benchmarked pretrained Seq2Seq models and LLMs on the task, and showed that conditioning these models on dialect identification predictions leads to improved normalization performance. Formally, given a dialectal Arabic sentence X and its corresponding dialect D , the task is to generate the CODAfied version Y according to:

$$P(Y|X, D) = \prod_{t=1}^n P(y_t|y_1, \dots, y_{t-1}, X, D)$$

This work provided the first comprehensive evaluation of modern generation models on CODAfication.

8.4 Text Editing

All the introduced controlled NLG approaches for the tasks we explored in this dissertation have relied on an explicit identification step to extract linguistic traits, such as gender, error types, or dialect, which are then used to condition Seq2Seq models. A key

observation across the three NLG tasks we explored is that the input and output sequences often share substantial overlap. As a result, full-sequence autoregressive models can be inefficient, since most tokens are simply copied from the input to the output.

To address this, we introduced a novel, text editing framework that reframes all three tasks as sequence tagging problems. Instead of generating text autoregressively, our approach assigns edit tags to input tokens; applying these tags transforms the input into the desired output, effectively combining both identification and generation into a single step. Formally, given an input sequence $x = x_1, x_2, \dots, x_n$, the goal is to assign a sequence of edit operations $e = e_1, e_2, \dots, e_n$ where $e_i \in E$, and E is the edit vocabulary, such that applying each e_i to x_i produces the output sequence $y = y_1, y_2, \dots, y_m$. These edit tags are derived automatically from data, removing the need for hand-crafted or language-specific edits.

We demonstrated that this text editing approach achieves state-of-the-art or highly competitive performance across all three tasks, surpassing prior models on most benchmarks. In addition to strong performance, they are over six times faster than previously introduced models, making them more practical for real-world use. Lastly, we showed that ensemble strategies further improve performance by leveraging complementary strengths across different models.

8.5 Future Work

Building on the findings and contributions of this dissertation, several promising directions emerge for future research in Arabic NLG. In this section, we outline three major avenues: extending controlled NLG to additional Arabic tasks, building personalized writing assistants, and developing human-centered evaluation frameworks.

Expanding Controlled NLG to Additional Arabic Tasks This dissertation laid the foundation for controlled Arabic NLG by introducing linguistically grounded approaches for gender rewriting, grammatical error correction (GEC), and dialectal text normalization (CODAfication). These contributions open the door to adapting our frameworks to other Arabic NLG tasks where linguistic traits matter. One promising direction is Arabic text simplification, a significantly underexplored task. In fact, our recent work on the SAMER corpus ([Alhafni et al., 2024b](#)), the first publicly available Arabic simplification dataset annotated with readability levels for school-aged learners, provides a concrete starting point for building controllable simplification systems that adapt to users’ reading proficiency. Similarly, tasks such as code-switching or style transfer across Arabic dialects could benefit from trait-conditioned generation models.

Personalized Arabic Writing Assistants The NLG tasks explored in this dissertation serve as key building blocks for designing user-tailored writing assistants for Arabic. A typical Arabic user may benefit from personalized outputs across multiple dimensions: receiving grammatically correct text tailored to their dialect, reading content rendered in their gender form, or being offered corrections that adapt to their specific error patterns. In Chapter 4, we demonstrated a proof of concept with the *User-Aware Arabic Gender Rewriter*, which aligns system outputs with user-specified gender preferences. This lays the groundwork for personalized assistants that can evolve with users, potentially through adaptive learning. Moreover, these systems could be used as data collection interfaces in an active learning setup, where user interactions trigger targeted annotation campaigns. Such strategies can help mitigate the scarcity of annotated Arabic NLG data and close the loop between model deployment and dataset expansion.

Human-Centered Evaluation Human-Centered Evaluation. Traditional automatic metrics in NLG, such as the M² scorer or BLEU, fall short in assessing how well a model’s outputs align with users’ specific needs or preferences. This is especially important when building systems that condition generation on user-specific linguistic traits. For example, in our gender rewriting experiments, we used BLEU to demonstrate reduced gender bias in machine translation outputs; however, a thorough human evaluation is still needed to verify that the rewritten outputs meet users’ expectations. The same applies to our GEC and CODAfication systems, where user-centered evaluation could provide more meaningful insights into system effectiveness and usability. Future work should prioritize incorporating human feedback, user studies, and task-specific satisfaction metrics to assess NLG systems beyond traditional reference-based scores.

Beyond Arabic Although this dissertation has focused on Arabic, many of its insights and the challenges it addresses are broadly applicable to other languages, particularly those that are morphologically rich.

For instance, the gender rewriting task, motivated by gender bias in machine translation, is directly relevant to other gender-marking languages such as Spanish, Italian, and French. One promising direction for future work is to extend the Arabic Parallel Gender Corpus (APGC) to other languages by leveraging multilingual parallel corpora such as OpenSubtitles, from which APGC was originally drawn.

In the context of GEC, we showed that conditioning models on error types can significantly improve performance, a finding that aligns with previous results in English (Yuan et al., 2021). Crucially, we were only able to enable this line of work through the availability of ARETA, an automatic error typing tool for Arabic. This highlights a broader need: for other languages, developing similar tools and annotation schemes

is essential not only to advance GEC but also facilitate related tasks such as GED and error-aware feedback generation.

In the case of dialectal text normalization, our work demonstrates that conditioning Seq2Seq models on dialect labels can significantly improve performance. While efforts in multilingual dialect normalization exist ([Kuparinen et al., 2023](#)), they typically treat dialect as an implicit factor rather than an explicit modeling signal. Our approach differs in that it leverages dialect identification as an attribute to guide normalization. This technique is easily transferable to other languages with dialectal variation: given a parallel normalization corpus and a dialect classifier, one could replicate our setup to build dialect-aware normalization systems that are more accurate, controllable, and linguistically informed.

Finally, our text editing framework offers a promising direction for language-agnostic applications. By deriving edit tags directly from data, it eliminates the need for language-specific resources. Its underlying edit extraction algorithm is language-agnostic and requires only monolingual parallel corpora, making it well-suited for a wide range of languages. Future work could explore applying this framework to other languages with NLG tasks that exhibit high input-output overlap.

Chapter 4 Appendix

A.1 LLMs Prompts

Prompt Lang	Prompt
EN	<p>You are an Arabic text rewriting tool that can rewrite gender-marking words based on the user needs. Please rewrite the following sentence marked with the tag <code><input> SRC </input></code> such that the speaker is <code>{speaker-gender}</code> and the listener is <code>{listener-gender}</code>. Only change the gender marking words. Do not rephrase any parts of the sentence. If the sentence does not have any gender marking words, do not rewrite it. Once you are done, output the revised sentence directly without providing any explanations. Remember to format the rewritten output with the tag <code><output> Your Rewritten Version </output></code>. Please start: <code><input> Input Sentence </input></code></p>
AR	<p>أنت أداة لإعادة صياغة النصوص العربية حيث يمكنك إعادة كتابة الكلمات التي تحمل علامات تمييز الجنس بناء على احتياجات المستخدم. يرجى إعادة كتابة الجملة التالية المحددة بالوسم <code><input></code> النص المدخل <code><input/></code> بحيث يكون المتكلم <code>{speaker-gender}</code> والمخاطب <code>{listener-gender}</code>. يجب فقط تغيير الكلمات التي تحمل علامات تمييز الجنس. لا تعيد صياغة أي أجزاء أخرى من الجملة. إذا لم تحتوي الجملة على أي كلمات تحمل علامات تمييز الجنس، فلا تقم بإعادة كتابتها. قم بإخراج الجملة المعدلة مباشرة دون تقديم أي تفسيرات. تذكر تنسيق النص المعدل باستخدام الوسم <code><output></code> النص المعدل <code><output/></code> الرجاء البدء: <code><input></code> النص المدخل <code><input/></code></p>

Table A.1: 0-shot prompts used to evaluate LLMs performance on gender rewriting. Prompt Lang is the prompt language either in English (EN) or Arabic (AR).

Prompt Lang	Prompt
EN	<p>You are an Arabic text rewriting tool that can rewrite gender-marking words based on the user needs. We will provide you with example sentences marked with the tag <input> SRC </input>. These sentences are followed by their rewritten versions, marked with <output> TGT </output>, as reviewed and rewritten by human experts. Please rewrite the following sentence marked with the tag <input> SRC </input> such that the speaker is {speaker-gender} and the listener is {listener-gender}. Only change the gender marking words. Do not rephrase any parts of the sentence. If the sentence does not have any gender marking words, do not rewrite it. Once you are done, output the revised sentence directly without providing any explanations. Here are some in-context examples:</p> <p>(1) <input> Input Sentence </input>: <output> Rewritten Sentence </output> (2) <input> Input Sentence </input>: <output> Rewritten Sentence </output> (3) <input> Input Sentence </input>: <output> Rewritten Sentence </output> (4) <input> Input Sentence </input>: <output> Rewritten Sentence </output> (5) <input> Input Sentence </input>: <output> Rewritten Sentence </output></p> <p>Please feel free to refer to these examples. Remember to format the rewritten output with the tag <output> Your Rewritten Version </output>. Please start: <input> Input Sentence </input></p>
AR	<p>أنت أداة لإعادة صياغة النصوص العربية حيث يمكنك إعادة كتابة الكلمات التي تحمل علامات تمييز الجنس بناء على احتياجات المستخدم. يرجى إعادة كتابة الجملة التالية المحددة بالوسم <input> النص المدخل </input> بحيث يكون المتكلم {speaker-gender} والمخاطب {listener-gender}. يجب فقط تغيير الكلمات التي تحمل علامات تمييز الجنس. لا تعيد صياغة أي أجزاء أخرى من الجملة. إذا لم تحتوي الجملة على أي كلمات تحمل علامات تمييز الجنس، فلا تقم بإعادة كتابتها. قم بإخراج الجملة المعدلة مباشرة دون تقديم أي تفسيرات. إليك بعض الأمثلة ضمن السياق:</p> <p>(١) <input> النص المدخل </input>: <output> النص المعدل </output> (٢) <input> النص المدخل </input>: <output> النص المعدل </output> (٣) <input> النص المدخل </input>: <output> النص المعدل </output> (٤) <input> النص المدخل </input>: <output> النص المعدل </output> (٥) <input> النص المدخل </input>: <output> النص المعدل </output></p> <p>لما تتردد في الرجوع إلى هذه الأمثلة. تذكر تنسيق النص المعدل باستخدام الوسم <output> النص المعدل </output> الرجاء البدء: <input> النص المدخل </input></p>

Table A.2: 5-shot prompts used to evaluate LLMs performance on gender rewriting. Prompt Lang is the prompt language either in English (EN) or Arabic (AR).

Prompt Lang	Prompt
EN	<p>You are an Arabic text rewriting tool that can rewrite gender-marking words based on the user needs. We will provide you with example sentences marked with the tag <input> SRC </input>. These sentences are followed by their rewritten versions, marked with <output> TGT </output>, as reviewed and rewritten by human experts. The words that need to be changed are marked with distinctive parentheses ((#word#)). Rewrite the sentence marked with the tag <input> Input Text </input> such that the speaker is {speaker-gender} and the listener is {listener-gender}. Change only the marked words without rephrasing any other part of the sentence. If no marked words are present, return the sentence as is without any modification. Once you are done, output the revised sentence directly without providing any explanations. Here are some in-context examples:</p> <p>(1) <input> Input Sentence </input>: <output> Rewritten Sentence </output> (2) <input> Input Sentence </input>: <output> Rewritten Sentence </output> (3) <input> Input Sentence </input>: <output> Rewritten Sentence </output> (4) <input> Input Sentence </input>: <output> Rewritten Sentence </output> (5) <input> Input Sentence </input>: <output> Rewritten Sentence </output></p> <p>Please feel free to refer to these examples. Remember to format the rewritten output with the tag <output> Your Rewritten Version </output>. Please start: <input> Input Sentence </input></p>
AR	<p>أنت أداة لإعادة صياغة النصوص العربية بحيث تعيد كتابة الكلمات التي تحمل علامات تمييز الجنس بناء على احتياجات المستخدم. سنزودك بجملة محددة بالوسم <input> النص المدخل </input> تلي هذه الجملة نسخ معاد كتابتها، محددة بالوسم <output> النص المعدل </output>، والتي تمت مراجعتها وتحريرها من قبل خبراء لغويين. الكلمات التي يجب تغييرها تكون محددة بين الأقواس المميزة ((#الكلمة#)). أعد كتابة الجملة المحددة بالوسم <input> النص المدخل </input> بحيث يكون المتكلم {speaker-gender} والمخاطب {listener-gender}. غير فقط الكلمات المحددة دون إعادة صياغة أي جزء آخر من الجملة. إذا لم توجد كلمات محددة، أعد الجملة كما هي بدون أي تعديل. عند الانتهاء، اخرج الجملة المعدلة مباشرة دون تقديم أي تفسيرات. تذكر استخدام الوسم <output> النص المعدل </output> لتنسيق النص المعدل. ابدأ: <input> النص المدخل </input> إليك بعض الأمثلة ضمن السياق:</p> <p>(١) <input> النص المدخل </input>: <output> النص المعدل </output> (٢) <input> النص المدخل </input>: <output> النص المعدل </output> (٣) <input> النص المدخل </input>: <output> النص المعدل </output> (٤) <input> النص المدخل </input>: <output> النص المعدل </output> (٥) <input> النص المدخل </input>: <output> النص المعدل </output></p> <p>لا تتردد في الرجوع إلى هذه الأمثلة. تذكر تنسيق النص المعدل باستخدام الوسم <output> النص المعدل </output> الرجاء البدء: <input> النص المدخل </input></p>

Table A.3: 5-shot prompts with gender identification used to evaluate LLMs performance on gender rewriting. Prompt Lang is the prompt language either in English (EN) or Arabic (AR).

A.2 LLMs Results

Model	P-Lang	Shots	P	R	F _{0.5}	BLEU
GPT-3.5-turbo	EN	0	19.4	<u>65.4</u>	22.6	61.5
GPT-3.5-turbo	EN	5	14.6	37.8	16.6	59.2
GPT-3.5-turbo	AR	0	18.9	59.8	21.9	64.5
GPT-3.5-turbo	AR	5	<u>21.9</u>	64.8	<u>25.2</u>	<u>69.5</u>
GPT-4o	EN	0	38.1	73.2	42.1	84.4
GPT-4o	EN	5	44.6	73.2	48.3	<u>88.8</u>
GPT-4o	AR	0	38.8	<u>75.8</u>	43.0	84.4
GPT-4o	AR	5	<u>49.2</u>	74.6	<u>52.8</u>	<u>88.8</u>
Fanar	EN	0	9.1	31.8	10.6	<u>42.3</u>
Fanar	EN	5	11.9	41.2	13.8	41.7
Fanar	AR	0	9.9	36.0	11.5	40.6
Fanar	AR	5	<u>12.6</u>	<u>43.7</u>	<u>14.7</u>	39.5
Jais-30B-Chat	EN	0	<u>8.2</u>	<u>33.4</u>	<u>13.1</u>	<u>24.5</u>
Jais-30B-Chat	EN	5	7.2	29.3	11.6	21.2
Jais-30B-Chat	AR	0	7.8	32.2	12.5	18.2
Jais-30B-Chat	AR	5	6.9	27.9	11.0	16.5

Table A.4: LLMs results on the Dev sets of APGCv2.0. P-Lang is the prompt language either in English (EN) or Arabic (AR). Best F_{0.5} results for each LLM are underlined.

Chapter 5 Appendix

B.1 LLMs Prompts

Prompt Lang	Prompt
EN	You are an Arabic grammatical error correction tool that can identify and correct grammatical and spelling errors in written text. Please identify and correct any grammatical and spelling errors in the following sentence marked with the tag <input> Input Sentence </input>. Make the minimal changes necessary to correct the sentence. Do not rephrase any parts of the sentence that are already grammatically correct, and avoid altering the meaning by adding or removing information. After making the corrections, output the revised sentence directly without providing any explanations. Remember to format the corrected output with the tag <output> Your Corrected Version </output>. Please start: <input> Input Sentence </input>
AR	أنت أداة لتصحيح الأخطاء النحوية والإملائية في اللغة العربية، حيث يمكنك تحديد وتصحيح الأخطاء النحوية والإملائية في النصوص المكتوبة. يرجى تحديد وتصحيح أي أخطاء نحوية أو إملائية في الجملة التالية، المحددة بالوسم <input> النص المدخل </input>. قم بإجراء الحد الأدنى من التعديلات اللازمة لتصحيح الجملة. لا تعد صياغة أي أجزاء صحيحة نحويًا، وتجنب تغيير المعنى من خلال إضافة أو حذف أي معلومات. بعد إجراء التصحيحات، قم بإخراج الجملة المصححة مباشرة دون أي تفسيرات. تذكر تنسيق النص المصحح باستخدام الوسم <output> النص المصحح </output> الرجاء البدء: <input> النص المدخل </input>

Table B.1: 0-shot prompts used to evaluate LLMs performance on GEC. Prompt Lang is the prompt language either in English (EN) or Arabic (AR).

Prompt Lang	Prompt
EN	<p>You are an Arabic grammatical error correction tool that can identify and correct grammatical and spelling errors in written text. We will provide you with example sentences marked with the tag <code><input> SRC </input></code>, which contain grammatical and spelling errors. These sentences are followed by the corrected versions, marked with <code><output> TGT </output></code>, as reviewed and edited by human experts. Please identify and correct any grammatical and spelling errors in the following sentence marked with the tag <code><input> SRC </input></code>. Make the minimal changes necessary to correct the sentence. Do not rephrase any parts of the sentence that are already grammatically correct, and avoid altering the meaning by adding or removing information. After making the corrections, output the revised sentence directly without providing any explanations. Here are some in-context examples:</p> <p>(1) <code><input> Input Sentence </input></code>: <code><output> Corrected Sentence </output></code> (2) <code><input> Input Sentence </input></code>: <code><output> Corrected Sentence </output></code> (3) <code><input> Input Sentence </input></code>: <code><output> Corrected Sentence </output></code> (4) <code><input> Input Sentence </input></code>: <code><output> Corrected Sentence </output></code> (5) <code><input> Input Sentence </input></code>: <code><output> Corrected Sentence </output></code> Please feel free to refer to these examples. Remember to format the corrected output with the tag <code><output> Your Corrected Version </output></code>. Please start: <code><input> Input Sentence </input></code></p>
AR	<p>أنت أداة لتصحيح الأخطاء النحوية والإملائية في اللغة العربية، حيث يمكنك تحديد وتصحيح الأخطاء النحوية والإملائية في النصوص المكتوبة. سنزودك بجمل تحتوي على أخطاء نحوية وإملائية، محددة بالوسم <code><input> النص المدخل </input></code>. تلي هذه الجمل النسخ المصححة، المحددة بالوسم <code><output> النص المصحح </output></code>، والتي تمت مراجعتها وتحريرها من قبل خبراء لغويين. يرجى تحديد وتصحيح أي أخطاء نحوية أو إملائية في الجملة التالية، المحددة بالوسم <code><input> النص المدخل </input></code>. قم بإجراء الحد الأدنى من التعديلات اللازمة لتصحيح الجملة. لا تعد صياغة أي أجزاء صحيحة نحويًا، وتجنب تغيير المعنى من خلال إضافة أو حذف أي معلومات. بعد إجراء التصحيحات، قم بإخراج الجملة المصححة مباشرة دون أي تفسيرات. إليك بعض الأمثلة ضمن السياق:</p> <p>(١) <code><input> النص المدخل </input></code>: <code><output> النص المصحح </output></code> (٢) <code><input> النص المدخل </input></code>: <code><output> النص المصحح </output></code> (٣) <code><input> النص المدخل </input></code>: <code><output> النص المصحح </output></code> (٤) <code><input> النص المدخل </input></code>: <code><output> النص المصحح </output></code> (٥) <code><input> النص المدخل </input></code>: <code><output> النص المصحح </output></code> لا تتردد في الرجوع إلى هذه الأمثلة. تذكر تنسيق النص المصحح باستخدام الوسم <code><output> النص المصحح </output></code> الرجاء البدء: <code><input> النص المدخل </input></code></p>

Table B.2: 5-shot prompts used to evaluate LLMs performance on GEC. Prompt Lang is the prompt language either in English (EN) or Arabic (AR).

Prompt Lang	Prompt
EN	<p>You are an Arabic grammatical error correction tool that can identify and correct grammatical and spelling errors in written text. We will provide you with example sentences marked with the tag <input> SRC </input>, which contain grammatical and spelling errors. These sentences are followed by the corrected versions, marked with <output> TGT </output>, as reviewed and edited by human experts. The words that need to be corrected are explicitly marked with double parentheses and hash symbols ((#word#)). Please correct only the marked words while keeping the rest of the sentence unchanged. Make the minimal changes necessary to correct the marked words. Do not rephrase any parts of the sentence that are already grammatically correct, and avoid altering the meaning by adding or removing information. If there are no errors, output the sentence as-is. After making the corrections, output the revised sentence directly without providing any explanations. Here are some in-context examples:</p> <p>(1) <input> Input Sentence </input>: <output> Corrected Sentence </output> (2) <input> Input Sentence </input>: <output> Corrected Sentence </output> (3) <input> Input Sentence </input>: <output> Corrected Sentence </output> (4) <input> Input Sentence </input>: <output> Corrected Sentence </output> (5) <input> Input Sentence </input>: <output> Corrected Sentence </output></p> <p>Please feel free to refer to these examples.</p> <p>Remember to format the corrected output with the tag <output> Your Corrected Version </output>. Please start: <input> Input Sentence </input></p>
AR	<p>أنت أداة لتصحيح الأخطاء النحوية والإملائية في اللغة العربية، حيث يمكنك تحديد وتصحيح الأخطاء النحوية والإملائية في النصوص المكتوبة. سنزودك بجملة تحتوي على أخطاء نحوية وإملائية، محددة بالوسم <input> النص المدخل </input>. تلي هذه الجملة النسخ المصححة، المحددة بالوسم <output> النص المصحح </output>، والتي تمت مراجعتها وتحريرها من قبل خبراء لغويين. الكلمات التي تحتاج إلى تصحيح تكون محددة بين الأقواس المميزة ((#الكلمة#)). يرجى تصحيح الكلمات المحددة فقط مع الحفاظ على باقي الجملة دون تغيير. قم بإجراء الحد الأدنى من التعديلات اللازمة لتصحيح الكلمات المحددة. لا تعد صياغة أي أجزاء صحيحة نحويًا، وتجنب تغيير المعنى من خلال إضافة أو حذف أي معلومات. بعد إجراء التصحيحات، قم بإخراج الجملة المصححة مباشرة دون أي تفسيرات. إليك بعض الأمثلة ضمن السياق:</p> <p>(١) <input> النص المدخل </input>: <output> النص المصحح </output> (٢) <input> النص المدخل </input>: <output> النص المصحح </output> (٣) <input> النص المدخل </input>: <output> النص المصحح </output> (٤) <input> النص المدخل </input>: <output> النص المصحح </output> (٥) <input> النص المدخل </input>: <output> النص المصحح </output></p> <p>لا تتردد في الرجوع إلى هذه الأمثلة.</p> <p>تذكر تنسيق النص المصحح باستخدام الوسم <output> النص المصحح </output> الرجاء البدء: <input> النص المدخل </input></p>

Table B.3: 5-shot prompts with binary GED used to evaluate LLMs performance on GEC. Prompt Lang is the prompt language either in English (EN) or Arabic (AR).

B.2 LLMs Results

Model	P-Lang	Shots	QALB-2014			QALB-2015			ZAEBUC			Avg.
			P	R	F _{0.5}	P	R	F _{0.5}	P	R	F _{0.5}	F _{0.5}
GPT-3.5-turbo	EN	0	70.6	54.8	66.7	59.6	39.6	54.1	70.8	70.3	70.7	<u>63.9</u>
GPT-3.5-turbo	EN	5	68.6	58.6	66.3	57.2	39.5	52.5	71.0	63.5	69.4	<u>62.7</u>
GPT-3.5-turbo	AR	0	70.0	58.5	67.3	58.2	39.2	53.1	68.3	71.3	68.9	63.1
GPT-3.5-turbo	AR	5	68.1	58.0	65.8	56.9	39.4	52.2	71.4	63.7	69.7	62.6
GPT-4o	EN	0	82.1	56.4	75.2	72.1	45.0	64.3	80.2	75.5	79.2	72.9
GPT-4o	EN	5	80.7	65.7	77.2	70.6	49.2	65.0	86.5	76.8	84.3	<u>75.5</u>
GPT-4o	AR	0	78.9	62.8	75.1	69.4	47.8	63.6	77.4	77.7	77.4	72.0
GPT-4o	AR	5	79.5	66.8	76.6	70.1	49.7	64.8	82.6	75.7	81.1	74.1
Fanar	EN	0	57.4	31.4	49.2	51.1	18.9	38.1	58.4	18.6	40.9	42.7
Fanar	EN	5	63.3	58.8	62.4	54.4	35.4	49.1	69.2	63.5	68.0	59.8
Fanar	AR	0	62.4	57.3	61.3	52.1	29.4	45.1	57.5	33.9	50.4	52.3
Fanar	AR	5	69.7	63.7	68.4	58.0	40.7	53.5	76.3	73.6	75.8	<u>65.9</u>
Jais-30B-Chat	EN	0	53.8	44.5	51.6	46.3	19.1	36.0	51.5	29.4	44.8	<u>44.1</u>
Jais-30B-Chat	EN	5	54.6	45.8	52.6	43.0	18.4	34.0	48.2	23.9	40.1	42.2
Jais-30B-Chat	AR	0	51.4	43.6	49.6	43.5	15.9	32.3	48.8	16.9	35.4	39.1
Jais-30B-Chat	AR	5	50.0	43.6	48.6	42.5	17.0	32.7	47.3	15.8	33.8	38.3

Table B.4: LLMs results on MSA and DA GEC on the Dev sets of QALB-2014, QALB-2015, and ZAEBUC. P-Lang is the prompt language either in English (EN) or Arabic (AR). Best average F_{0.5} results for each LLM are underlined. Best overall results are in bold.

B.3 Error Type Statistics

2-Class	13-Class	43-Class	QALB-2014			QALB-2015				ZAEUBC		
			Train	Dev	Test	Train	Dev	Test-L1	Test-L2	Train	Dev	Test
E	Delete	Delete	6,442	346	540	584	339	250	309	305	64	66
	Merge-B	Merge-B	15,063	797	795	377	231	625	199	849	180	133
	Merge-I	Merge-I	15,296	812	807	390	241	629	200	851	180	133
	M	M	30	0	0	14	3	4	4	6	2	2
		MI	1,360	69	59	400	220	56	169	127	30	25
		MT	76	0	4	136	72	2	40	4	0	1
	M+O	MI+OH	243	17	15	9	9	8	5	7	1	8
	O	O	3,255	166	164	75	52	144	70	296	64	68
		OA	7,627	313	252	290	136	514	138	27	4	6
		OC	466	27	19	45	23	17	26	30	7	7
		OD	4,086	207	204	283	167	146	166	103	24	21
		OH	90,579	4,785	4,632	1,076	599	4,499	587	1,905	401	451
		OM	4,062	228	217	361	215	188	184	123	23	30
		OR	8,358	425	446	763	415	369	362	162	32	36
		OT	14,688	758	623	54	37	733	26	408	101	138
		OW	1,885	149	107	32	12	77	9	12	4	2
		OA+OH	480	19	12	4	1	23	0	1	1	1
		OA+OR	215	8	6	4	4	11	3	0	1	0
		OD+OG	573	32	32	22	15	23	9	11	4	2
		OD+OH	317	11	17	13	2	10	2	8	1	1
		OD+OM	104	4	5	12	7	1	6	0	2	1
		OD+OR	675	33	26	61	32	22	32	8	2	2
		OH+OM	2,339	134	123	231	106	114	109	54	15	13
		OH+OT	1,468	56	65	2	1	71	1	31	9	9
		OM+OR	382	15	19	62	27	23	15	17	0	4
		OR+OT	193	10	7	4	4	2	1	7	0	0
	O+X	OH+XC	323	24	18	6	3	15	2	20	0	4
	P	P	11,379	598	687	855	453	446	483	237	51	36
	S	S	536	41	19	188	125	26	103	44	14	21
		SF	96	5	4	46	33	2	21	4	0	2
		SW	4,804	201	229	887	502	186	422	121	22	28
	X	X	3,668	216	182	144	59	161	57	106	26	17
		XC	5,980	373	369	279	180	289	141	201	31	46
		XC+XG	296	23	40	0	3	1	0	1	0	0
		XC+XN	500	18	41	29	13	23	9	24	3	3
		XF	852	63	25	835	494	35	463	51	12	14
		XG	809	38	30	317	175	35	158	86	20	24
		XM	225	15	6	151	91	12	68	14	6	3
		XN	1,107	47	41	210	115	47	84	30	9	2
		XT	155	16	9	46	26	6	24	15	3	4
	Split	Split	7,828	432	399	80	42	382	34	49	10	10
	UNK	UNK	6,835	331	300	969	454	257	416	361	78	61
C	C	C	795,510	41,875	39,690	33,007	19,004	38,063	17,651	18,411	3,839	3,683
			1,021,165	53,737	51,285	43,353	24,742	48,547	22,808	25,127	5,276	5,118

Table B.5: The statistics of the different GED granularity error types we model across the three datasets. The description of the labels in the 13-Class and 43-Class categories are in Table 5.2. For the 2-Class labels, **E** refers to erroneous words and **C** refers to correct words.

Chapter 6 Appendix

C.1 LLMs Prompts

Prompt Lang	Prompt
EN	<p>You are a dialectal Arabic text normalization tool that can normalize dialectal Arabic text into the Conventional Orthography for Dialectal Arabic (CODA). CODA provides a standardized system for writing Arabic dialects, which are often written informally or phonetically. By using CODA, your task is to convert these informal, dialectal texts into a consistent, standardized orthographic form, making them more uniform while retaining the nuances of the original dialect. Please standardize the following sentence marked with the tag <code><input></code> Input Sentence <code></input></code> into the CODA convention. Avoid altering the meaning by adding or removing information. Make sure the normalized output sentence is in Arabic script. Output the normalized sentence directly without providing any explanations. Remember to format the CODA standardized output with the tag <code><output></code> Your CODA Version <code></output></code>. Please start: <code><input></code> Input Sentence <code></input></code></p>
AR	<p>أنت أداة لتصحيح النصوص العربية العامية، حيث يمكنك تصحيح النصوص العامية إلى الإملاء القياسي للعامية العربية (CODA). يوفر CODA نظاماً موحداً للكتابة اللهجات العربية التي تكتب غالباً بطريقة غير رسمية أو صوتية. باستخدام CODA، مهمتك هي تصحيح هذه النصوص العامية غير الرسمية إلى شكل إملائي موحد ومتسق، مع الحفاظ على الفروق الدقيقة للهجة الأصلية. يرجى تصحيح الجملة التالية المميزة بالوسم <code><input></code> النص المدخل <code></input></code> وفقاً لمعيار CODA. تجنب تغيير المعنى عن طريق إضافة أو إزالة معلومات. أخرج الجملة مباشرة دون أي تفسيرات. تذكر تنسيق النص المصحح باستخدام الوسم <code><output></code> النص المصحح <code></output></code> الرجاء البدء: <code><input></code> النص المدخل <code></input></code></p>

Table C.1: 0-shot prompts used to evaluate LLMs performance on CODAfication. Prompt Lang is the prompt language either in English (EN) or Arabic (AR).

Prompt Lang	Prompt
EN	<p>You are a dialectal Arabic text normalization tool that can normalize dialectal Arabic text into the Conventional Orthography for Dialectal Arabic (CODA). CODA provides a standardized system for writing Arabic dialects, which are often written informally or phonetically. By using CODA, your task is to convert these informal, dialectal texts into a consistent, standardized orthographic form, making them more uniform while retaining the nuances of the original dialect. We will provide you with example sentences marked with the tag <code><input></code> Input Sentence <code></input></code>, which are written in dialectal Arabic. These sentences are followed by their CODA versions, marked with <code><output></code> Corrected Sentence <code></output></code>, as reviewed and edited by human experts. Please standardize the following sentence marked with the tag <code><input></code> Input Sentence <code></input></code> into the CODA convention. Avoid altering the meaning by adding or removing information. Output the normalized sentence directly without providing any explanations. Here are some in-context examples:</p> <p>(1) <code><input></code> Input Sentence <code></input></code>: <code><output></code> Corrected Sentence <code></output></code> (2) <code><input></code> Input Sentence <code></input></code>: <code><output></code> Corrected Sentence <code></output></code> (3) <code><input></code> Input Sentence <code></input></code>: <code><output></code> Corrected Sentence <code></output></code> (4) <code><input></code> Input Sentence <code></input></code>: <code><output></code> Corrected Sentence <code></output></code> (5) <code><input></code> Input Sentence <code></input></code>: <code><output></code> Corrected Sentence <code></output></code></p> <p>Please feel free to refer to these examples. Remember to format the corrected output with the tag <code><output></code> Your Corrected Version <code></output></code>. Please start: <code><input></code> Input Sentence <code></input></code></p>
AR	<p>أنت أداة لتصحيح النصوص العربية العامية، حيث يمكنك تصحيح النصوص العامية إلى الإملاء القياسي للعامية العربية (CODA). يوفر CODA نظاما موحدا لكتابة اللهجات العربية التي تكتب غالبا بطريقة غير رسمية أو صوتية. باستخدام CODA، مهمتك هي تصحيح هذه النصوص العامية غير الرسمية إلى شكل إملائي موحد ومتسق، مع الحفاظ على الفروق الدقيقة لل لهجة الأصلية. سنزودك بجملة أمثلة مميزة بالوسم <code><input></code> النص المدخل <code></input></code>، وهي مكتوبة بالعامية العربية. تتبع هذه الجملة نسخها المطابقة لمعيار CODA، والمحددة بالوسم <code><output></code> النص المصحح <code></output></code>، والتي تمت مراجعتها وتحريرها من قبل خبراء لغويين. يرجى تصحيح الجملة التالية المميزة بالوسم <code><input></code> النص المدخل <code></input></code> وفقا لمعيار CODA. تجنب تغيير المعنى عن طريق إضافة أو إزالة معلومات. أخرج الجملة مباشرة دون أي تفسيرات. إليك بعض الأمثلة ضمن السياق:</p> <p>(١) <code><input></code> النص المدخل <code></input></code>: <code><output></code> النص المصحح <code></output></code> (٢) <code><input></code> النص المدخل <code></input></code>: <code><output></code> النص المصحح <code></output></code> (٣) <code><input></code> النص المدخل <code></input></code>: <code><output></code> النص المصحح <code></output></code> (٤) <code><input></code> النص المدخل <code></input></code>: <code><output></code> النص المصحح <code></output></code> (٥) <code><input></code> النص المدخل <code></input></code>: <code><output></code> النص المصحح <code></output></code></p> <p>لا تتردد في الرجوع إلى هذه الأمثلة. تذكر تنسيق النص المصحح باستخدام الوسم <code><output></code> النص المصحح <code></output></code> الرجاء البدء: <code><input></code> النص المدخل <code></input></code></p>

Table C.2: 5-shot prompts used to evaluate LLMs performance on CODAfication. Prompt Lang is the prompt language either in English (EN) or Arabic (AR).

Prompt Lang	Prompt
EN	<p>You are a dialectal Arabic text normalization tool that can normalize dialectal Arabic text into the Conventional Orthography for Dialectal Arabic (CODA). CODA provides a standardized system for writing Arabic dialects, which are often written informally or phonetically. By using CODA, your task is to convert these informal, dialectal texts into a consistent, standardized orthographic form, making them more uniform while retaining the nuances of the original dialect. We will provide you with example sentences in the {city} dialect marked with the tag <input> Input Sentence </input>. These sentences are followed by their CODA versions, marked with <output> Corrected Sentence </output>, as reviewed and edited by human experts. Please standardize the following sentence marked with the tag <input> Input Sentence </input> into the CODA convention. Avoid altering the meaning by adding or removing information. Output the normalized sentence directly without providing any explanations. Here are some in-context examples:</p> <p>(1) <input> Input Sentence </input>: <output> Corrected Sentence </output> (2) <input> Input Sentence </input>: <output> Corrected Sentence </output> (3) <input> Input Sentence </input>: <output> Corrected Sentence </output> (4) <input> Input Sentence </input>: <output> Corrected Sentence </output> (5) <input> Input Sentence </input>: <output> Corrected Sentence </output></p> <p>Please feel free to refer to these examples. Remember to format the corrected output with the tag <output> Your Corrected Version </output>. Please start: <input> Input Sentence </input></p>

Table C.3: 5-shot prompts with specified dialect used to evaluate LLMs performance on CODAfication. Prompt Lang is the prompt language either in English (EN) or Arabic (AR).

C.2 LLMs Results

Model	P-Lang	Shots	P	R	F _{0.5}
GPT-3.5-turbo	EN	0	22.8	17.7	21.5
GPT-3.5-turbo	EN	5	<u>35.5</u>	<u>29.7</u>	<u>34.1</u>
GPT-3.5-turbo	AR	0	24.2	22.7	23.9
GPT-3.5-turbo	AR	5	27.0	26.5	26.9
GPT-4o	EN	0	28.8	25.5	28.1
GPT-4o	EN	5	<u>53.7</u>	<u>54.4</u>	<u>53.8</u>
GPT-4o	AR	0	36.4	33.5	35.8
GPT-4o	AR	5	50.1	48.6	49.8
Fanar	EN	0	13.7	14.6	13.9
Fanar	EN	5	22.4	26.8	23.1
Fanar	AR	0	17.2	19.0	17.5
Fanar	AR	5	<u>24.5</u>	<u>28.8</u>	<u>25.2</u>
Jais-30B-Chat	EN	0	9.7	9.8	9.7
Jais-30B-Chat	EN	5	<u>12.8</u>	<u>13.4</u>	<u>12.9</u>
Jais-30B-Chat	AR	0	9.4	9.2	9.4
Jais-30B-Chat	AR	5	12.1	12.5	12.2

Table C.4: LLMs results on the Dev set of MADAR CODA. P-Lang is the prompt language either in English (EN) or Arabic (AR). Best F_{0.5} results for each LLM are underlined.

Chapter 7 Appendix

D.1 Edit Tagging Results

Model	Input	Comp.	Prune	MADAR CODA			APGC v2.0		
				P	R	F _{0.5}	P	R	F _{0.5}
AraBERTv02	Word	✗	-	87.9	66.5	82.6	89.1	85.3	88.3
AraBERTv02	Subword	✗	-	87.6	76.8	85.2	89.2	86.7	88.7
AraBERTv02	Word	✓	-	85.6	76.9	83.7	89.3	88.3	89.1
AraBERTv02	Subword	✓	-	86.9	79.2	85.2	88.9	88.2	88.7
ARBERTv2	Word	✗	-	86.4	61.0	79.8	87.9	82.4	86.7
ARBERTv2	Subword	✗	-	84.5	69.0	80.8	87.4	84.9	86.9
ARBERTv2	Word	✓	-	81.8	68.4	78.7	87.5	86.4	87.2
ARBERTv2	Subword	✓	-	84.2	70.8	81.2	87.2	86.7	87.1
CAMeLBERT	Word	✗	-	88.3	66.4	82.8	88.9	85.1	88.1
CAMeLBERT	Subword	✗	-	87.1	76.8	84.8	88.1	87.5	88.0
CAMeLBERT	Word	✓	-	85.6	76.0	83.5	88.5	87.7	88.3
CAMeLBERT	Subword	✓	-	87.0	78.8	85.2	88.2	87.4	88.0
AraBERTv02	Subword	✓	10	89.1	81.7	86.0	-	-	-
AraBERTv02	Subword	✓	20	87.7	73.1	84.4	-	-	-
AraBERTv02	Subword	✓	30	88.3	72.1	84.5	-	-	-
CAMeLBERT	Subword	✓	10	88.4	76.3	85.7	-	-	-
CAMeLBERT	Subword	✓	20	88.2	72.6	84.6	-	-	-
CAMeLBERT	Subword	✓	30	88.7	71.3	84.6	-	-	-
AraBERTv02	Word	✓	10	-	-	-	89.7	87.3	89.2
AraBERTv02	Word	✓	20	-	-	-	89.4	86.3	88.8
AraBERTv02	Word	✓	30	-	-	-	89.7	85.6	88.9
CAMeLBERT	Word	✓	10	-	-	-	88.6	87.0	88.3
CAMeLBERT	Word	✓	20	-	-	-	88.9	86.3	88.3
CAMeLBERT	Word	✓	30	-	-	-	89.4	85.3	88.5

Table D.1: CODAfication and gender rewriting results on the Dev sets of MADAR CODA and APGC v2.0. **Input** is the input unit representation (word or subword). **Comp.** indicates whether the edit is compressed. Pruning experiments were conducted using the top two models (AraBERTv02 and CAMeLBERT). Best results are in bold.

Model	Input	Comp.	Subset	Prune	QALB-2014			QALB-2015			ZAEUC		
					P	R	F _{0.5}	P	R	F _{0.5}	P	R	F _{0.5}
AraBERTv02	Word	✗	All	-	81.0	64.3	77.0	63.0	36.5	55.0	84.8	69.5	81.2
AraBERTv02	Subword	✗	All	-	81.0	67.8	77.9	63.8	37.1	55.8	84.4	71.3	81.4
AraBERTv02	Word	✓	All	-	80.8	66.6	77.5	61.3	41.5	55.9	83.8	71.4	81.0
AraBERTv02	Subword	✓	All	-	81.1	69.1	78.4	62.6	40.4	56.4	84.3	72.9	81.7
ARBERTv2	Word	✗	All	-	78.9	57.7	73.5	58.0	32.5	50.2	82.2	54.8	74.8
ARBERTv2	Subword	✗	All	-	78.7	60.8	74.3	60.0	33.1	51.6	79.7	58.1	74.2
ARBERTv2	Word	✓	All	-	77.8	61.4	73.8	57.9	35.5	51.4	80.7	62.8	76.3
ARBERTv2	Subword	✓	All	-	78.6	60.0	74.0	59.8	36.7	53.1	82.7	62.1	77.5
CAMELBERT	Word	✗	All	-	81.2	61.5	76.3	63.0	35.1	54.3	84.6	66.4	80.2
CAMELBERT	Subword	✗	All	-	80.4	65.2	76.9	63.2	37.5	55.6	83.5	69.3	80.2
CAMELBERT	Word	✓	All	-	79.9	65.4	76.5	62.3	39.9	56.0	84.2	69.3	80.7
CAMELBERT	Subword	✓	All	-	80.6	67.4	77.6	63.2	42.0	57.4	84.6	70.8	81.4
AraBERTv02	Subword	✓	All	10	81.8	68.8	78.8	64.8	40.6	57.9	84.5	71.9	81.6
AraBERTv02	Subword	✓	All	20	81.4	68.6	78.5	65.4	38.7	57.5	85.3	72.0	82.2
AraBERTv02	Subword	✓	All	30	81.6	68.1	78.5	65.7	39.0	57.8	85.8	72.3	82.7
CAMELBERT	Subword	✓	All	10	81.2	67.4	78.0	65.1	39.9	57.8	85.1	71.0	81.8
CAMELBERT	Subword	✓	All	20	81.3	66.7	77.9	67.6	36.7	57.9	84.4	70.1	81.1
CAMELBERT	Subword	✓	All	30	81.1	67.5	77.9	65.6	39.2	57.8	84.7	70.0	81.3
AraBERTv02	Subword	✓	NoPnx	-	88.3	77.7	85.9	62.5	<u>35.3</u>	54.2	87.2	<u>77.0</u>	85.0
AraBERTv02	Subword	✓	NoPnx	10	88.8	<u>78.1</u>	86.4	69.1	30.7	55.2	87.6	76.1	85.0
AraBERTv02	Subword	✓	NoPnx	20	89.0	77.8	86.5	70.4	30.1	<u>55.5</u>	87.9	75.8	85.1
AraBERTv02	Subword	✓	NoPnx	30	<u>89.4</u>	77.5	<u>86.7</u>	<u>70.9</u>	28.7	54.8	<u>88.1</u>	76.8	<u>85.6</u>
AraBERTv02	Subword	✓	Pnx	-	90.6	83.0	<u>89.0</u>	90.5	<u>85.6</u>	<u>89.5</u>	96.8	94.0	96.2
AraBERTv02	Subword	✓	Pnx	10	89.5	<u>83.6</u>	88.3	90.5	85.1	89.4	<u>96.9</u>	93.8	<u>96.3</u>
AraBERTv02	Subword	✓	Pnx	20	<u>90.7</u>	82.8	<u>89.0</u>	<u>90.7</u>	84.9	<u>89.5</u>	96.7	93.6	96.1
AraBERTv02	Subword	✓	Pnx	30	90.1	<u>83.6</u>	88.7	90.5	85.0	89.4	96.5	<u>94.0</u>	96.0

Table D.2: MSA GEC results on the Dev sets of QALB-2014, QALB-2015, and ZAEUC. **Input** is the input unit representation (word or subword). **Comp.** indicates whether the edit is compressed. **Subset** specifies whether the edits capture all errors, punctuation-only errors (Pnx), or non-punctuation errors (NoPnx). NoPnx models are evaluated after removing punctuation, while Pnx models are evaluated on a version of the Dev set where all non-punctuation errors are corrected. Pruning experiments were conducted using the top two models (AraBERTv02 and CAMELBERT), while punctuation segregation experiments used the best model (AraBERTv02). Best **All** results are in bold. Best **NoPnx** and **Pnx** results are underlined.

Bibliography

Ahmed Abdelali, Kareem Darwish, Nadir Durrani, and Hamdy Mubarak. 2016. Farasa: A fast and furious segmenter for Arabic. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, pages 11–16, San Diego, California.

Ahmed Abdelali, Sabit Hassan, Hamdy Mubarak, Kareem Darwish, and Younes Samih. 2021. [Pre-training bert on arabic tweets: Practical considerations](#). *Preprint*, arXiv:2102.10684.

Muhammad Abdul-Mageed, Hassan Alhuzali, and Mohamed Elaraby. 2018a. [You tweet what you speak: A city-level dataset of Arabic dialects](#). In *Proceedings of the 11th Language Resources and Evaluation Conference*, Miyazaki, Japan. European Language Resource Association.

Muhammad Abdul-Mageed, Hassan Alhuzali, and Mohamed Elaraby. 2018b. [You tweet what you speak: A city-level dataset of Arabic dialects](#). In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Muhammad Abdul-Mageed, AbdelRahim Elmadany, and El Moatez Billah Nagoudi. 2021a. [ARBERT & MARBERT: Deep bidirectional transformers for Arabic](#). 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 7088–7105, Online. Association for Computational Linguistics.

Muhammad Abdul-Mageed, AbdelRahim Elmadany, Chiyu Zhang, El Moatez Billah Nagoudi, Houda Bouamor, and Nizar Habash. 2023. [NADI 2023: The fourth nuanced Arabic dialect identification shared task](#). In *Proceedings of ArabicNLP 2023*, pages 600–613, Singapore (Hybrid). Association for Computational Linguistics.

Muhammad Abdul-Mageed, Amr Keleg, AbdelRahim Elmadany, Chiyu Zhang, Injy Hamed, Walid Magdy, Houda Bouamor, and Nizar Habash. 2024. [NADI 2024: The fifth nuanced Arabic dialect identification shared task](#). In *Proceedings of The Second Arabic Natural Language Processing Conference*, pages 709–728, Bangkok, Thailand. Association for Computational Linguistics.

Muhammad Abdul-Mageed, Chiyu Zhang, AbdelRahim Elmadany, Houda Bouamor, and Nizar Habash. 2021b. [NADI 2021: The second nuanced Arabic dialect identification shared task](#). In *Proceedings of the Sixth Arabic Natural Language Processing Workshop*, pages 244–259, Kyiv, Ukraine (Virtual). Association for Computational Linguistics.

Muhammad Abdul-Mageed, Chiyu Zhang, AbdelRahim Elmadany, Houda Bouamor, and Nizar Habash. 2022. [NADI 2022: The third nuanced Arabic dialect identification shared task](#). In *Proceedings of the Seventh Arabic Natural Language Processing Workshop (WANLP)*, pages 85–97, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

- Sweta Agrawal and Marine Carpuat. 2019. [Controlling text complexity in neural machine translation](#). 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)*, pages 1549–1564, Hong Kong, China. Association for Computational Linguistics.
- Sweta Agrawal and Marine Carpuat. 2023. [Controlling pre-trained language models for grade-specific text simplification](#). In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 12807–12819, Singapore. Association for Computational Linguistics.
- Mohamed Al-Badrashiny and Mona Diab. 2016. LILI: A simple language independent approach for language identification. In *Proceedings of the International Conference on Computational Linguistics (COLING)*, Osaka, Japan.
- Mohamed Al-Badrashiny, Ramy Eskander, Nizar Habash, and Owen Rambow. 2014. [Automatic transliteration of Romanized dialectal Arabic](#). In *Proceedings of the Eighteenth Conference on Computational Natural Language Learning*, pages 30–38, Ann Arbor, Michigan. Association for Computational Linguistics.
- Rania Al-Sabbagh and Roxana Girju. 2012. [YADAC: Yet another dialectal Arabic corpus](#). In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, pages 2882–2889, Istanbul, Turkey. European Language Resources Association (ELRA).
- Nora Al-Twairesh, Rawan Al-Matham, Nora Madi, Nada Almgren, Al-Hanouf Al-Aljmi, Shahad Alshalan, Raghad Alshalan, Nafla Alrumayyan, Shams Al-Manea, Sumayah Bawazeer, Nourah Al-Mutlaq, Nada Almanea, Waad Bin Huwaymil, Dalal

- Alqusair, Reem Alotaibi, Suha Al-Senaydi, and Abeer Alfutamani. 2018. SUAR: Towards building a corpus for the Saudi dialect. In *Proceedings of the International Conference on Arabic Computational Linguistics (ACLing)*.
- Abdullah Alfaifi and Eric Atwell. 2012. Arabic learner corpora (alc): a taxonomy of coding errors. In *The 8th International Computing Conference in Arabic*.
- Abdullah Alfaifi, Eric Atwell, and Ghazi Abuhakema. 2013. Error annotation of the Arabic learner corpus. In *Language Processing and Knowledge in the Web*, pages 14–22. Springer.
- Bashar Alhafni, Sarah Al-Towaity, Ziyad Fawzy, Fatema Nassar, Fadhl Eryani, Houda Bouamor, and Nizar Habash. 2024a. [Exploiting dialect identification in automatic dialectal text normalization](#). In *Proceedings of The Second Arabic Natural Language Processing Conference*, pages 42–54, Bangkok, Thailand. Association for Computational Linguistics.
- Bashar Alhafni, Nizar Habash, and Houda Bouamor. 2020. [Gender-aware reinflection using linguistically enhanced neural models](#). In *Proceedings of the Second Workshop on Gender Bias in Natural Language Processing*, pages 139–150, Barcelona, Spain (Online). Association for Computational Linguistics.
- Bashar Alhafni, Nizar Habash, and Houda Bouamor. 2022a. [The Arabic parallel gender corpus 2.0: Extensions and analyses](#). In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 1870–1884, Marseille, France. European Language Resources Association.
- Bashar Alhafni, Nizar Habash, and Houda Bouamor. 2022b. [User-centric gender rewriting](#). In *Proceedings of the 2022 Conference of the North American Chapter of the*

- Association for Computational Linguistics: Human Language Technologies*, pages 618–631, Seattle, United States. Association for Computational Linguistics.
- Bashar Alhafni, Nizar Habash, Houda Bouamor, Ossama Obeid, Sultan Alrowili, Daliyah AlZeer, Kawla Mohmad Shnqiti, Ahmed Elbakry, Muhammad ElNokrashy, Mohamed Gabr, Abderrahmane Issam, Abdelrahim Qaddoumi, Vijay Shanker, and Mahmoud Zyate. 2022c. [The shared task on gender rewriting](#). In *Proceedings of the Seventh Arabic Natural Language Processing Workshop (WANLP)*, pages 98–107, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.
- Bashar Alhafni, Reem Hazim, Juan David Pineros Liberato, Muhamed Al Khalil, and Nizar Habash. 2024b. [The SAMER Arabic text simplification corpus](#). In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 16079–16093, Torino, Italia. ELRA and ICCL.
- Bashar Alhafni, Go Inoue, Christian Khairallah, and Nizar Habash. 2023a. [Advancements in Arabic grammatical error detection and correction: An empirical investigation](#). In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 6430–6448, Singapore. Association for Computational Linguistics.
- Bashar Alhafni, Vivek Kulkarni, Dhruv Kumar, and Vipul Raheja. 2024c. [Personalized text generation with fine-grained linguistic control](#). In *Proceedings of the 1st Workshop on Personalization of Generative AI Systems (PERSONALIZE 2024)*, pages 88–101, St. Julians, Malta. Association for Computational Linguistics.
- Bashar Alhafni, Ossama Obeid, and Nizar Habash. 2023b. [The user-aware Arabic gender rewriter](#). In *Proceedings of the First Workshop on Gender-Inclusive Translation*

- Technologies*, pages 3–11, Tampere, Finland. European Association for Machine Translation.
- Ahmed Ali, Salam Khalifa, and Nizar Habash. 2019. [Towards Variability Resistant Dialectal Speech Evaluation](#). In *Proc. Interspeech 2019*, pages 336–340.
- Manar Alkhatib, Azza Abdel Monem, and Khaled Shaalan. 2020. [Deep learning for Arabic error detection and correction](#). *ACM Trans. Asian Low-Resour. Lang. Inf. Process.*, 19(5).
- Ebtesam Almazrouei, Hamza Alobeidli, Abdulaziz Alshamsi, Alessandro Cappelli, Ruxandra Cojocaru, Mérouane Debbah, Étienne Goffinet, Daniel Hesslow, Julien Launay, Quentin Malartic, Daniele Mazzotta, Badreddine Noune, Baptiste Pannier, and Guilherme Penedo. 2023. [The falcon series of open language models](#). *Preprint*, arXiv:2311.16867.
- Khalid Alnajjar and Mika Härmäläinen. 2024. [Normalization of arabic dialects into modern standard arabic using](#). *Journal of Data Mining & Digital Humanities*, NLP4DH.
- Wissam Antoun, Fady Baly, and Hazem Hajj. 2020. [AraBERT: Transformer-based model for Arabic language understanding](#). In *Proceedings of the 4th Workshop on Open-Source Arabic Corpora and Processing Tools, with a Shared Task on Offensive Language Detection*, pages 9–15, Marseille, France. European Language Resource Association.
- Akari Asai, Sneha Kudugunta, Xinyan Yu, Terra Blevins, Hila Gonen, Machel Reid, Yulia Tsvetkov, Sebastian Ruder, and Hannaneh Hajishirzi. 2024. [BUFFET: Benchmarking large language models for few-shot cross-lingual transfer](#). In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational*

- Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 1771–1800, Mexico City, Mexico. Association for Computational Linguistics.
- Mohammed Attia, Mohamed Al-Badrashiny, and Mona Diab. 2014. [GWU-HASP: Hybrid Arabic spelling and punctuation corrector](#). In *Proceedings of the EMNLP 2014 Workshop on Arabic Natural Language Processing (ANLP)*, pages 148–154, Doha, Qatar. Association for Computational Linguistics.
- Dana Awad. 2013. La ponctuation en Arabe: histoire et règles. etude constrative avec le français et l’anglais.
- Abhijeet Awasthi, Sunita Sarawagi, Rasna Goyal, Sabyasachi Ghosh, and Vihari Piratla. 2019. [Parallel iterative edit models for local sequence transduction](#). In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4260–4270, Hong Kong, China. Association for Computational Linguistics.
- Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *CoRR*, abs/1409.0473.
- Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *Proceedings of the International Conference on Learning Representations (ICLR)*.
- Marion Bartl, Malvina Nissim, and Albert Gatt. 2020. [Unmasking contextual stereotypes: Measuring and mitigating BERT’s gender bias](#). In *Proceedings of the Second Workshop on Gender Bias in Natural Language Processing*, pages 1–16, Barcelona, Spain (Online). Association for Computational Linguistics.

- Christine Basta, Marta R. Costa-jussà, and José A. R. Fonollosa. 2020. [Towards mitigating gender bias in a decoder-based neural machine translation model by adding contextual information](#). In *Proceedings of the Fourth Widening Natural Language Processing Workshop*, pages 99–102, Seattle, USA. Association for Computational Linguistics.
- Kenneth R. Beesley. 1996. [Arabic finite-state morphological analysis and generation](#). In *COLING 1996 Volume 1: The 16th International Conference on Computational Linguistics*.
- Riadh Belkebir and Nizar Habash. 2021. [Automatic error type annotation for Arabic](#). In *Proceedings of the 25th Conference on Computational Natural Language Learning*, pages 596–606, Online. Association for Computational Linguistics.
- Samuel Bell, Helen Yannakoudakis, and Marek Rei. 2019. [Context is key: Grammatical error detection with contextual word representations](#). In *Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 103–115, Florence, Italy. Association for Computational Linguistics.
- Luisa Bentivogli, Beatrice Savoldi, Matteo Negri, Mattia A. Di Gangi, Roldano Cattoni, and Marco Turchi. 2020. [Gender in danger? evaluating speech translation technology on the MuST-SHE corpus](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6923–6933, Online. Association for Computational Linguistics.
- Su Lin Blodgett, Solon Barocas, Hal Daumé III, and Hanna Wallach. 2020. [Language \(technology\) is power: A critical survey of “bias” in NLP](#). In *Proceedings of the 58th*

- Annual Meeting of the Association for Computational Linguistics*, pages 5454–5476, Online. Association for Computational Linguistics.
- Tolga Bolukbasi, Kai-Wei Chang, James Y Zou, Venkatesh Saligrama, and Adam T Kalai. 2016. [Man is to computer programmer as woman is to homemaker? debiasing word embeddings](#). In *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc.
- Shikha Bordia and Samuel R. Bowman. 2019. [Identifying and reducing gender bias in word-level language models](#). In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Student Research Workshop*, pages 7–15, Minneapolis, Minnesota. Association for Computational Linguistics.
- Houda Bouamor, Nizar Habash, and Kemal Oflazer. 2014. [A multidialectal parallel corpus of Arabic](#). In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 1240–1245, Reykjavik, Iceland. European Language Resources Association (ELRA).
- Houda Bouamor, Nizar Habash, Mohammad Salameh, Wajdi Zaghouani, Owen Rambow, Dana Abdulrahim, Ossama Obeid, Salam Khalifa, Fadhl Eryani, Alexander Erdmann, and Kemal Oflazer. 2018. [The MADAR Arabic dialect corpus and lexicon](#). In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).
- Houda Bouamor, Sabit Hassan, and Nizar Habash. 2019. The MADAR Shared Task

on Arabic Fine-Grained Dialect Identification. In *Proceedings of the Fourth Arabic Natural Language Processing Workshop (WANLP19)*, Florence, Italy.

Fethi Bougares and Houda Bouamor. 2015. [UMMU@QALB-2015 shared task: Character and word level SMT pipeline for automatic error correction of Arabic text](#). In *Proceedings of the Second Workshop on Arabic Natural Language Processing*, pages 166–172, Beijing, China. Association for Computational Linguistics.

Adriane Boyd. 2018. [Using Wikipedia edits in low resource grammatical error correction](#). In *Proceedings of the 2018 EMNLP Workshop W-NUT: The 4th Workshop on Noisy User-generated Text*, pages 79–84, Brussels, Belgium. Association for Computational Linguistics.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, and 12 others. 2020. [Language models are few-shot learners](#). In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Christopher Bryant, Mariano Felice, Øistein E. Andersen, and Ted Briscoe. 2019. [The BEA-2019 shared task on grammatical error correction](#). In *Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 52–75, Florence, Italy. Association for Computational Linguistics.

Christopher Bryant, Mariano Felice, and Ted Briscoe. 2017. [Automatic annotation and evaluation of error types for grammatical error correction](#). In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1:*

- Long Papers*), pages 793–805, Vancouver, Canada. Association for Computational Linguistics.
- Christopher Bryant, Zheng Yuan, Muhammad Reza Qorib, Hannan Cao, Hwee Tou Ng, and Ted Briscoe. 2023. [Grammatical Error Correction: A Survey of the State of the Art](#). *Computational Linguistics*, 49(3):643–701.
- Tim Buckwalter. 2002. Buckwalter Arabic morphological analyzer version 1.0. Linguistic Data Consortium (LDC) catalog number LDC2002L49, ISBN 1-58563-257-0.
- Tim Buckwalter. 2004a. Issues in Arabic Orthography and Morphology Analysis. In *Proceedings of the Workshop on Computational Approaches to Arabic Script-based Languages (CAASL)*, pages 31–34, Geneva, Switzerland.
- Tim Buckwalter. 2004b. [Issues in Arabic orthography and morphology analysis](#). In *Proceedings of the Workshop on Computational Approaches to Arabic Script-based Languages*, pages 31–34, Geneva, Switzerland. COLING.
- Yang Trista Cao and III Daumé, Hal. 2021. [Toward gender-inclusive coreference resolution: An analysis of gender and bias throughout the machine learning lifecycle*](#). *Computational Linguistics*, 47(3):615–661.
- Amanda Cercas Curry, Judy Robertson, and Verena Rieser. 2020. [Conversational assistants and gender stereotypes: Public perceptions and desiderata for voice personas](#). In *Proceedings of the Second Workshop on Gender Bias in Natural Language Processing*, pages 72–78, Barcelona, Spain (Online). Association for Computational Linguistics.
- Junyi Chai, Reid Pryzant, Victor Ye Dong, Konstantin Golobokov, Chenguang Zhu, and Yi Liu. 2022. [Fast: Improving controllability for text generation with feedback aware self-training](#). *Preprint*, arXiv:2210.03167.

- Alvin Chan, Yew-Soon Ong, Bill Pung, Aston Zhang, and Jie Fu. 2022. [Cocon: A self-supervised approach for controlled text generation](#). *Preprint*, arXiv:2006.03535.
- Eugene Charniak, Kevin Knight, and Kenji Yamada. 2003. [Syntax-based language models for statistical machine translation](#). In *Proceedings of Machine Translation Summit IX: Papers*, New Orleans, USA.
- Xinlei Chen, Hao Fang, Tsung-Yi Lin, Ramakrishna Vedantam, Saurabh Gupta, Piotr Dollar, and C. Lawrence Zitnick. 2015. [Microsoft coco captions: Data collection and evaluation server](#). *Preprint*, arXiv:1504.00325.
- Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. [Learning phrase representations using RNN encoder–decoder for statistical machine translation](#). In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734, Doha, Qatar. Association for Computational Linguistics.
- Martin Chodorow, Joel Tetreault, and Na-Rae Han. 2007. [Detection of grammatical errors involving prepositions](#). In *Proceedings of the Fourth ACL-SIGSEM Workshop on Prepositions*, pages 25–30, Prague, Czech Republic. Association for Computational Linguistics.
- Ana Cimitan, Ana Alves Pinto, and Michaela Geierhos. 2024. [Curation of benchmark templates for measuring gender bias in named entity recognition models](#). In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 4238–4246, Torino, Italia. ELRA and ICCL.
- Chloe Ciora, Nur Iren, and Malihe Alikhani. 2021. [Examining covert gender bias: A case](#)

- study in Turkish and English machine translation models. In *Proceedings of the 14th International Conference on Natural Language Generation*, pages 55–63, Aberdeen, Scotland, UK. Association for Computational Linguistics.
- Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. 2020. [ELECTRA: Pre-training text encoders as discriminators rather than generators](#). In *International Conference on Learning Representations*.
- Marta R. Costa-jussà and Adrià de Jorge. 2020. [Fine-tuning neural machine translation on gender-balanced datasets](#). In *Proceedings of the Second Workshop on Gender Bias in Natural Language Processing*, pages 26–34, Barcelona, Spain (Online). Association for Computational Linguistics.
- Ryan Cotterell and Chris Callison-Burch. 2014. [A multi-dialect, multi-genre corpus of informal written Arabic](#). In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 241–245, Reykjavik, Iceland. European Language Resources Association (ELRA).
- C. o. E. Council of Europe. 2001. Common european framework of reference for languages: learning, teaching, assessment. Cambridge University Press.
- Steven Coyne, Keisuke Sakaguchi, Diana Galvan-Sosa, Michael Zock, and Kentaro Inui. 2023. [Analyzing the performance of gpt-3.5 and gpt-4 in grammatical error correction](#). Preprint, arXiv:2303.14342.
- Daniel Dahlmeier and Hwee Tou Ng. 2011. [Grammatical error correction with alternating structure optimization](#). In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 915–923, Portland, Oregon, USA. Association for Computational Linguistics.

- Daniel Dahlmeier and Hwee Tou Ng. 2012. [Better evaluation for grammatical error correction](#). In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 568–572, Montréal, Canada. Association for Computational Linguistics.
- Daniel Dahlmeier, Hwee Tou Ng, and Siew Mei Wu. 2013. [Building a large annotated corpus of learner English: The NUS corpus of learner English](#). In *Proceedings of the Eighth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 22–31, Atlanta, Georgia. Association for Computational Linguistics.
- Josef Dai, Xuehai Pan, Ruiyang Sun, Jiaming Ji, Xinbo Xu, Mickel Liu, Yizhou Wang, and Yaodong Yang. 2024. [Safe RLHF: Safe reinforcement learning from human feedback](#). In *The Twelfth International Conference on Learning Representations*.
- Robert Dale and Adam Kilgarriff. 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.
- Sumanth Dathathri, Andrea Madotto, Janice Lan, Jane Hung, Eric Frank, Piero Molino, Jason Yosinski, and Rosanne Liu. 2020. [Plug and play language models: A simple approach to controlled text generation](#). *Preprint*, arXiv:1912.02164.
- Christopher Davis, Andrew Caines, Øistein E. Andersen, Shiva Taslimipoor, Helen Yannakoudakis, Zheng Yuan, Christopher Bryant, Marek Rei, and Paula Buttery. 2024. [Prompting open-source and commercial language models for grammatical error correction of English learner text](#). In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 11952–11967, Bangkok, Thailand. Association for Computational Linguistics.

- Daniel de Vassimon Manela, David Errington, Thomas Fisher, Boris van Breugel, and Pasquale Minervini. 2021. [Stereotype and skew: Quantifying gender bias in pre-trained and fine-tuned language models](#). In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 2232–2242, Online. Association for Computational Linguistics.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. [BERT: Pre-training of deep bidirectional transformers for language understanding](#). In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.
- Mona Diab, Nizar Habash, Owen Rambow, Mohamed AlTantawy, and Yassine Benajiba. 2010. COLABA: Arabic Dialect Annotation and Processing. In *Proceedings of the Workshop on Language Resources and Human Language Technology for Semitic Languages*.
- Mona T Diab, Mohamed Al-Badrashiny, Maryam Aminian, Mohammed Attia, Heba Elfardy, Nizar Habash, Abdelati Hawwari, Wael Salloum, Pradeep Dasigi, and Ramy Eskander. 2014. Tharwa: A Large Scale Dialectal Arabic-Standard Arabic-English Lexicon. In *Proceedings of the Language Resources and Evaluation Conference (LREC)*, pages 3782–3789, Reykjavik, Iceland.
- Emily Dinan, Angela Fan, Adina Williams, Jack Urbanek, Douwe Kiela, and Jason Weston. 2020a. [Queens are powerful too: Mitigating gender bias in dialogue generation](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language*

- Processing (EMNLP)*, pages 8173–8188, Online. Association for Computational Linguistics.
- Emily Dinan, Angela Fan, Ledell Wu, Jason Weston, Douwe Kiela, and Adina Williams. 2020b. [Multi-dimensional gender bias classification](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 314–331, Online. Association for Computational Linguistics.
- Rotem Dror, Gili Baumer, Segev Shlomov, and Roi Reichart. 2018. [The hitchhiker’s guide to testing statistical significance in natural language processing](#). In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1383–1392, Melbourne, Australia. Association for Computational Linguistics.
- Mahmoud El-Haj. 2020. [Habibi - a multi dialect multi national Arabic song lyrics corpus](#). In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 1318–1326, Marseille, France. European Language Resources Association.
- Mostafa Elaraby, Ahmed Y. Tawfik, Mahmoud Khaled, Hany Hassan, and Aly Osama. 2018. [Gender aware spoken language translation applied to english-arabic](#). In *2018 2nd International Conference on Natural Language and Speech Processing (ICNLSP)*, pages 1–6.
- Heba Elfardy, Mohamed Al-Badrashiny, and Mona Diab. 2014. [AIDA: Identifying code switching in informal Arabic text](#). In *Proceedings of the First Workshop on Computational Approaches to Code Switching*, pages 94–101, Doha, Qatar. Association for Computational Linguistics.
- AbdelRahim Elmadany, El Moatez Billah Nagoudi, and Muhammad Abdul-Mageed.

2023. [Octopus: A multitask model and toolkit for Arabic natural language generation](#). In *Proceedings of ArabicNLP 2023*, pages 232–243, Singapore (Hybrid). Association for Computational Linguistics.
- Ali Emami, Paul Trichelair, Adam Trischler, Kaheer Suleman, Hannes Schulz, and Jackie Chi Kit Cheung. 2019. [The KnowRef coreference corpus: Removing gender and number cues for difficult pronominal anaphora resolution](#). In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3952–3961, Florence, Italy. Association for Computational Linguistics.
- Alexander Erdmann, Nizar Habash, Dima Taji, and Houda Bouamor. 2017. [Low resourced machine translation via morpho-syntactic modeling: The case of dialectal Arabic](#). In *Proceedings of Machine Translation Summit XVI: Research Track*, pages 185–200, Nagoya Japan.
- Fadhl Eryani, Nizar Habash, Houda Bouamor, and Salam Khalifa. 2020. [A spelling correction corpus for multiple Arabic dialects](#). In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 4130–4138, Marseille, France. European Language Resources Association.
- Joel Escudé Font and Marta R. Costa-jussà. 2019. [Equalizing gender bias in neural machine translation with word embeddings techniques](#). In *Proceedings of the First Workshop on Gender Bias in Natural Language Processing*, pages 147–154, Florence, Italy. Association for Computational Linguistics.
- Ramy Eskander, Nizar Habash, Owen Rambow, and Nadi Tomeh. 2013. Processing spontaneous orthography. In *Proceedings of the Conference of the North American*

- Chapter of the Association for Computational Linguistics (NAACL)*, pages 585–595, Atlanta, Georgia.
- Asma Etman and Louis Beex. 2015. Language and Dialect Identification: A Survey. In *Proceedings of the Intelligent Systems Conference (IntelliSys)*, London, UK.
- Tao Fang, Shu Yang, Kaixin Lan, Derek F. Wong, Jinpeng Hu, Lidia S. Chao, and Yue Zhang. 2023. [Is chatgpt a highly fluent grammatical error correction system? a comprehensive evaluation](#). *Preprint*, arXiv:2304.01746.
- Noura Farra, Nadi Tomeh, Alla Rozovskaya, and Nizar Habash. 2014. [Generalized character-level spelling error correction](#). In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 161–167, Baltimore, Maryland. Association for Computational Linguistics.
- Mariano Felice and Ted Briscoe. 2015. [Towards a standard evaluation method for grammatical error detection and correction](#). In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 578–587, Denver, Colorado. Association for Computational Linguistics.
- Mariano Felice, Zheng Yuan, Øistein E. Andersen, Helen Yannakoudakis, and Ekaterina Kochmar. 2014. [Grammatical error correction using hybrid systems and type filtering](#). In *Proceedings of the Eighteenth Conference on Computational Natural Language Learning: Shared Task*, pages 15–24, Baltimore, Maryland. Association for Computational Linguistics.
- Charles F Ferguson. 1959. Diglossia. *Word*, 15(2):325–340.

Hassan Gadalla, Hanaa Kilany, Howaida Arram, Ashraf Yacoub, Alaa El-Habashi, Amr Shalaby, Krisjanis Karins, Everett Rowson, Robert MacIntyre, Paul Kingsbury, David Graff, and Cynthia McLemore. 1997. CALLHOME Egyptian Arabic transcripts LDC97T19. Web Download. Philadelphia: Linguistic Data Consortium.

Sarthak Garg, Mozhddeh Gheini, Clara Emmanuel, Tatiana Likhomanenko, Qin Gao, and Matthias Paulik. 2024. [Generating gender alternatives in machine translation](#). In *Proceedings of the 5th Workshop on Gender Bias in Natural Language Processing (GeBNLP)*, pages 237–254, Bangkok, Thailand. Association for Computational Linguistics.

Abbas Ghaddar, Yimeng Wu, Sunyam Bagga, Ahmad Rashid, Khalil Bibi, Mehdi Rezagholizadeh, Chao Xing, Yasheng Wang, Xinyu Duan, Zhefeng Wang, Baoxing Huai, Xin Jiang, Qun Liu, and Phillippe Langlais. 2022. [Revisiting pre-trained language models and their evaluation for Arabic natural language processing](#). In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 3135–3151, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Hila Gonen and Yoav Goldberg. 2019. [Lipstick on a pig: Debiasing methods cover up systematic gender biases in word embeddings but do not remove them](#). *Preprint*, arXiv:1903.03862.

Hila Gonen and Kellie Webster. 2020. [Automatically identifying gender issues in machine translation using perturbations](#). In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1991–1995, Online. Association for Computational Linguistics.

David Graff, Mohamed Maamouri, Basma Bouziri, Sondos Krouna, Seth Kulick, and Tim Buckwalter. 2009. Standard Arabic Morphological Analyzer (SAMA) Version 3.1. Linguistic Data Consortium LDC2009E73.

Yvette Graham, Nitika Mathur, and Timothy Baldwin. 2014. [Randomized significance tests in machine translation](#). In *Proceedings of the Ninth Workshop on Statistical Machine Translation*, pages 266–274, Baltimore, Maryland, USA. Association for Computational Linguistics.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, and 542 others. 2024. [The llama 3 herd of models](#). *Preprint*, arXiv:2407.21783.

Arthur Gretton, Karsten Borgwardt, Malte Rasch, Bernhard Schölkopf, and Alex Smola. 2006. [A kernel method for the two-sample-problem](#). In *Advances in Neural Information Processing Systems*, volume 19. MIT Press.

Roman Grundkiewicz, Marcin Junczys-Dowmunt, and Kenneth Heafield. 2019. [Neural grammatical error correction systems with unsupervised pre-training on synthetic data](#). In *Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 252–263, Florence, Italy. Association for Computational Linguistics.

Nizar Habash, Houda Bouamor, and Christine Chung. 2019. [Automatic gender identification and reinflection in Arabic](#). In *Proceedings of the First Workshop on Gender*

Bias in Natural Language Processing, pages 155–165, Florence, Italy. Association for Computational Linguistics.

Nizar Habash, Mona Diab, and Owen Rambow. 2012a. [Conventional orthography for dialectal Arabic](#). In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, pages 711–718, Istanbul, Turkey. European Language Resources Association (ELRA).

Nizar Habash, Fadhl Eryani, Salam Khalifa, Owen Rambow, Dana Abdulrahim, Alexander Erdmann, Reem Faraj, Wajdi Zaghrouani, Houda Bouamor, Nasser Zalmout, Sara Hassan, Faisal Al-Shargi, Sakhar Alkhereyf, Basma Abdulkareem, Ramy Eskander, Mohammad Salameh, and Hind Saddiki. 2018. [Unified guidelines and resources for Arabic dialect orthography](#). In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Nizar Habash, Ramy Eskander, and Abdelati Hawwari. 2012b. A Morphological Analyzer for Egyptian Arabic. In *Proceedings of the Workshop of the Special Interest Group on Computational Morphology and Phonology (SIGMORPHON)*, pages 1–9, Montréal, Canada.

Nizar Habash and David Palfreyman. 2022. [ZAEBUC: An annotated Arabic-English bilingual writer corpus](#). In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 79–88, Marseille, France. European Language Resources Association.

Nizar Habash, Owen Rambow, Mona Diab, and Reem Kanjawi-Faraj. 2008. Guidelines

for Annotation of Arabic Dialectness. In *Proceedings of the Workshop on HLT & NLP within the Arabic World*, Marrakech, Morocco.

Nizar Habash and Ryan Roth. 2011. Using deep morphology to improve automatic error detection in Arabic handwriting recognition. In *Proceedings of the Conference of the Association for Computational Linguistics (ACL)*, pages 875–884, Portland, Oregon, USA.

Nizar Habash, Abdelhadi Souidi, and Tim Buckwalter. 2007. On Arabic Transliteration. In A. van den Bosch and A. Souidi, editors, *Arabic Computational Morphology: Knowledge-based and Empirical Methods*, pages 15–22. Springer, Netherlands.

Nizar Y Habash. 2010. *Introduction to Arabic natural language processing*, volume 3. Morgan & Claypool Publishers.

Xingwei He. 2021. [Parallel refinements for lexically constrained text generation with BART](#). In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 8653–8666, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. [Long Short-Term Memory](#). *Neural Computation*, 9(8):1735–1780.

Xinyu Hua and Lu Wang. 2020. [PAIR: Planning and iterative refinement in pre-trained transformers for long text generation](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 781–793, Online. Association for Computational Linguistics.

John Hutchins. 2004. The first public demonstration of machine translation: the georgetown-ibm system, 7th january 1954.

- Go Inoue, Bashar Alhafni, Nurpeiis Baimukan, Houda Bouamor, and Nizar Habash. 2021. [The interplay of variant, size, and task type in Arabic pre-trained language models](#). In *Proceedings of the Sixth Arabic Natural Language Processing Workshop*, pages 92–104, Kyiv, Ukraine (Virtual). Association for Computational Linguistics.
- Go Inoue, Salam Khalifa, and Nizar Habash. 2022. [Morphosyntactic tagging with pre-trained language models for Arabic and its dialects](#). In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 1708–1719, Dublin, Ireland. Association for Computational Linguistics.
- Mustafa Jarrar, Nizar Habash, Faeq Alrimawi, Diyam Akra, and Nasser Zalmout. 2016. Curras: an annotated corpus for the Palestinian Arabic dialect. *Language Resources and Evaluation*, pages 1–31.
- Serena Jeblee, Weston Feely, Houda Bouamor, Alon Lavie, Nizar Habash, and Kemal Oflazer. 2014. Domain and dialect adaptation for machine translation into egyptian Arabic. In *Proceedings of the Workshop for Arabic Natural Language Processing (WANLP)*, pages 196–206, Doha, Qatar.
- Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. 2023. [Survey of hallucination in natural language generation](#). *ACM Comput. Surv.*, 55(12).
- Renlong Jie, Xiaojun Meng, Lifeng Shang, Xin Jiang, and Qun Liu. 2024. [Prompt-based length controlled generation with multiple control types](#). In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 1067–1085, Bangkok, Thailand. Association for Computational Linguistics.

Melvin Johnson. 2020. [A scalable approach to reducing gender bias in google translate](#). Google AI Blog.

Melvin Johnson, Mike Schuster, Quoc V Le, Maxim Krikun, Yonghui Wu, Zhifeng Chen, Nikhil Thorat, Fernanda Viégas, Martin Wattenberg, Greg Corrado, and 1 others. 2016. Google’s multilingual neural machine translation system: Enabling zero-shot translation. *arXiv preprint arXiv:1611.04558*.

Pratik Joshi, Sebastin Santy, Amar Budhiraja, Kalika Bali, and Monojit Choudhury. 2020. [The state and fate of linguistic diversity and inclusion in the NLP world](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6282–6293, Online. Association for Computational Linguistics.

Marcin Junczys-Dowmunt and Roman Grundkiewicz. 2014. [The AMU system in the CoNLL-2014 shared task: Grammatical error correction by data-intensive and feature-rich statistical machine translation](#). In *Proceedings of the Eighteenth Conference on Computational Natural Language Learning: Shared Task*, pages 25–33, Baltimore, Maryland. Association for Computational Linguistics.

Marcin Junczys-Dowmunt and Roman Grundkiewicz. 2016. [Phrase-based machine translation is state-of-the-art for automatic grammatical error correction](#). In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1546–1556, Austin, Texas. Association for Computational Linguistics.

Marcin Junczys-Dowmunt, Roman Grundkiewicz, Shubha Guha, and Kenneth Heafield. 2018. [Approaching neural grammatical error correction as a low-resource machine translation task](#). 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 595–606, New Orleans, Louisiana. Association for Computational Linguistics.

Juseon-Do, Jingun Kwon, Hidetaka Kamigaito, and Manabu Okumura. 2024. [InstructCMP: Length control in sentence compression through instruction-based large language models](#). In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 8980–8996, Bangkok, Thailand. Association for Computational Linguistics.

Nal Kalchbrenner, Lasse Espeholt, Karen Simonyan, Aaron van den Oord, Alex Graves, and Koray Kavukcuoglu. 2017. [Neural machine translation in linear time](#). *Preprint*, arXiv:1610.10099.

Moussa Kamal Eddine, Nadi Tomeh, Nizar Habash, Joseph Le Roux, and Michalis Vazirgiannis. 2022. [AraBART: a pretrained Arabic sequence-to-sequence model for abstractive summarization](#). In *Proceedings of the Seventh Arabic Natural Language Processing Workshop (WANLP)*, pages 31–42, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

Masahiro Kaneko, Aizhan Imankulova, Danushka Bollegala, and Naoaki Okazaki. 2022. [Gender bias in masked language models for multiple languages](#). In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2740–2750, Seattle, United States. Association for Computational Linguistics.

Masahiro Kaneko and Mamoru Komachi. 2019. [Multi-head multi-layer attention to deep language representations for grammatical error detection](#). *Preprint*, arXiv:1904.07334.

Masahiro Kaneko, Masato Mita, Shun Kiyono, Jun Suzuki, and Kentaro Inui. 2020.

- Encoder-decoder models can benefit from pre-trained masked language models in grammatical error correction. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4248–4254, Online. Association for Computational Linguistics.
- Masahiro Kaneko and Naoaki Okazaki. 2023. Reducing sequence length by predicting edit spans with large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 10017–10029, Singapore. Association for Computational Linguistics.
- Masahiro Kaneko and Naoaki Okazaki. 2024. Controlled generation with prompt insertion for natural language explanations in grammatical error correction. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 3955–3961, Torino, Italia. ELRA and ICCL.
- Anisia Katinskaia and Roman Yangarber. 2024. GPT-3.5 for grammatical error correction. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 7831–7843, Torino, Italia. ELRA and ICCL.
- Styliani Katsarou, Borja Rodríguez-Gálvez, and Jesse Shanahan. 2022. Measuring gender bias in contextualized embeddings. In *Computer Sciences and Mathematics Forum*, volume 3, page 3. MDPI.
- Satoru Katsumata and Mamoru Komachi. 2020. Stronger baselines for grammatical error correction using a pretrained encoder-decoder model. In *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics*

and the 10th International Joint Conference on Natural Language Processing, pages 827–832, Suzhou, China. Association for Computational Linguistics.

Amr Keleg, Sharon Goldwater, and Walid Magdy. 2023. [ALDi: Quantifying the Arabic level of dialectness of text](#). In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 10597–10611, Singapore. Association for Computational Linguistics.

Amr Keleg and Walid Magdy. 2023. [Arabic dialect identification under scrutiny: Limitations of single-label classification](#). In *Proceedings of ArabicNLP 2023*, pages 385–398, Singapore (Hybrid). Association for Computational Linguistics.

Nitish Shirish Keskar, Bryan McCann, Lav R. Varshney, Caiming Xiong, and Richard Socher. 2019. [Ctrl: A conditional transformer language model for controllable generation](#). *Preprint*, arXiv:1909.05858.

Salam Khalifa, Nizar Habash, Dana Abdulrahim, and Sara Hassan. 2016a. A Large Scale Corpus of Gulf Arabic. In *Proceedings of the Language Resources and Evaluation Conference (LREC)*, Portorož, Slovenia.

Salam Khalifa, Nizar Habash, Dana Abdulrahim, and Sara Hassan. 2016b. [A large scale corpus of Gulf Arabic](#). In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 4282–4289, Portorož, Slovenia. European Language Resources Association (ELRA).

Salam Khalifa, Nizar Habash, Fadhl Eryani, Ossama Obeid, Dana Abdulrahim, and Meera Al Kaabi. 2018. A morphologically annotated corpus of emirati Arabic. In *Proceedings of the Language Resources and Evaluation Conference (LREC)*, Miyazaki, Japan.

- Salam Khalifa, Nasser Zalmout, and Nizar Habash. 2020. [Morphological analysis and disambiguation for Gulf Arabic: The interplay between resources and methods](#). In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 3895–3904, Marseille, France. European Language Resources Association.
- Ahmed El Kholy and Nizar Habash. 2010. Techniques for Arabic Morphological Detokenization and Orthographic Denormalization. In *Proceedings of the Workshop on Language Resources and Human Language Technology for Semitic Languages*, Valletta, Malta.
- Sheldon Klein. 1965. [Control of style with a generative grammar](#). *Language*, 41(4):619–631.
- Sheldon Klein and Robert F. Simmons. 1963. [Syntactic dependence and the computer generation of coherent discourse](#). *Mech. Transl. Comput. Linguistics*, 7:50–61.
- Ekaterina Kochmar, Øistein Andersen, and Ted Briscoe. 2012. [HOO 2012 error recognition and correction shared task: Cambridge University submission report](#). In *Proceedings of the Seventh Workshop on Building Educational Applications Using NLP*, pages 242–250, Montréal, Canada. Association for Computational Linguistics.
- Philipp Koehn, Franz J. Och, and Daniel Marcu. 2003. [Statistical phrase-based translation](#). In *Proceedings of the 2003 Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*, pages 127–133.
- Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2023. [Large language models are zero-shot reasoners](#). *Preprint*, arXiv:2205.11916.
- Hadas Kotek, Rikker Dockum, and David Sun. 2023. [Gender bias and stereotypes in](#)

large language models. In *Proceedings of The ACM Collective Intelligence Conference*, CI '23, page 12–24. ACM.

Aomi Koyama, Tomoshige Kiyuna, Kenji Kobayashi, Mio Arai, and Mamoru Komachi. 2020. Construction of an evaluation corpus for grammatical error correction for learners of Japanese as a second language. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 204–211, Marseille, France. European Language Resources Association.

Ben Krause, Akhilesh Deepak Gotmare, Bryan McCann, Nitish Shirish Keskar, Shafiq Joty, Richard Socher, and Nazneen Fatema Rajani. 2021. GeDi: Generative discriminator guided sequence generation. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 4929–4952, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Olli Kuparinen, Aleksandra Miletić, and Yves Scherrer. 2023. Dialect-to-standard normalization: A large-scale multilingual evaluation. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 13814–13828, Singapore. Association for Computational Linguistics.

Sang Kwon, Gagan Bhatia, El Moatez Billah Nagoudi, and Muhammad Abdul-Mageed. 2023. Beyond English: Evaluating LLMs for Arabic grammatical error correction. In *Proceedings of ArabicNLP 2023*, pages 101–119, Singapore (Hybrid). Association for Computational Linguistics.

Wuwei Lan, Yang Chen, Wei Xu, and Alan Ritter. 2020. An empirical study of pre-trained transformers for Arabic information extraction. In *Proceedings of the 2020*

Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 4727–4734, Online. Association for Computational Linguistics.

Anne Lauscher, Rafik Takieddin, Simone Paolo Ponzetto, and Goran Glavaš. 2020. [AraWEAT: Multidimensional analysis of biases in Arabic word embeddings](#). In *Proceedings of the Fifth Arabic Natural Language Processing Workshop*, pages 192–199, Barcelona, Spain (Online). Association for Computational Linguistics.

Brian Lester, Rami Al-Rfou, and Noah Constant. 2021. [The power of scale for parameter-efficient prompt tuning](#). In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3045–3059, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

V. I. Levenshtein. 1966. Binary Codes Capable of Correcting Deletions, Insertions and Reversals. *Soviet Physics Doklady*, 10:707.

Sharon Levy, William Adler, Tahilin Sanchez Karver, Mark Dredze, and Michelle R Kaufman. 2024. [Gender bias in decision-making with large language models: A study of relationship conflicts](#). In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 5777–5800, Miami, Florida, USA. Association for Computational Linguistics.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. [BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

- Jiwei Li, Will Monroe, Alan Ritter, Dan Jurafsky, Michel Galley, and Jianfeng Gao. 2016. [Deep reinforcement learning for dialogue generation](#). In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1192–1202, Austin, Texas. Association for Computational Linguistics.
- Junyi Li, Tianyi Tang, Wayne Xin Zhao, and Ji-Rong Wen. 2021. [Pretrained language model for text generation: A survey](#). In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21*, pages 4492–4499. International Joint Conferences on Artificial Intelligence Organization. Survey Track.
- Xiang Lisa Li and Percy Liang. 2021. [Prefix-tuning: Optimizing continuous prompts for generation](#). 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 4582–4597, Online. Association for Computational Linguistics.
- Xun Liang, Hanyu Wang, Yezhaohui Wang, Shichao Song, Jiawei Yang, Simin Niu, Jie Hu, Dan Liu, Shunyu Yao, Feiyu Xiong, and Zhiyu Li. 2024. [Controllable text generation for large language models: A survey](#). *Preprint*, arXiv:2408.12599.
- Zhiyu Lin and Mark Riedl. 2021. [Plug-and-blend: A framework for controllable story generation with blended control codes](#). In *Proceedings of the Third Workshop on Narrative Understanding*, pages 62–71, Virtual. Association for Computational Linguistics.
- Pierre Lison and Jörg Tiedemann. 2016. [OpenSubtitles2016: Extracting large parallel corpora from movie and TV subtitles](#). In *Proceedings of the Tenth International Con-*

- ference on Language Resources and Evaluation (LREC'16)*, pages 923–929, Portorož, Slovenia. European Language Resources Association (ELRA).
- Haochen Liu, Jamell Dacon, Wenqi Fan, Hui Liu, Zitao Liu, and Jiliang Tang. 2020a. [Does gender matter? towards fairness in dialogue systems](#). In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 4403–4416, Barcelona, Spain (Online). International Committee on Computational Linguistics.
- Haochen Liu, Wentao Wang, Yiqi Wang, Hui Liu, Zitao Liu, and Jiliang Tang. 2020b. [Mitigating gender bias for neural dialogue generation with adversarial learning](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 893–903, Online. Association for Computational Linguistics.
- Sheng Liu, Haotian Ye, Lei Xing, and James Zou. 2024. [In-context vectors: Making in context learning more effective and controllable through latent space steering](#). *Preprint*, arXiv:2311.06668.
- Xiao Liu, Yanan Zheng, Zhengxiao Du, Ming Ding, Yujie Qian, Zhilin Yang, and Jie Tang. 2023. [Gpt understands, too](#). *Preprint*, arXiv:2103.10385.
- Yinhan Liu, Jiatao Gu, Naman Goyal, Xian Li, Sergey Edunov, Marjan Ghazvininejad, Mike Lewis, and Luke Zettlemoyer. 2020c. [Multilingual denoising pre-training for neural machine translation](#). *Transactions of the Association for Computational Linguistics*, 8:726–742.
- Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. [Roberta: A robustly optimized bert pretraining approach](#). *Preprint*, arXiv:1907.11692.

- Yinhong Liu, Yixuan Su, Ehsan Shareghi, and Nigel Collier. 2022. [Plug-and-play recipe generation with content planning](#). In *Proceedings of the 2nd Workshop on Natural Language Generation, Evaluation, and Metrics (GEM)*, pages 223–234, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.
- Mengsay Loem, Masahiro Kaneko, Sho Takase, and Naoaki Okazaki. 2023. [Exploring effectiveness of GPT-3 in grammatical error correction: A study on performance and controllability in prompt-based methods](#). In *Proceedings of the 18th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2023)*, pages 205–219, Toronto, Canada. Association for Computational Linguistics.
- Xiang Long, Chuang Gan, and Gerard de Melo. 2018. [Video captioning with multi-faceted attention](#). *Transactions of the Association for Computational Linguistics*, 6:173–184.
- Kaiji Lu, Piotr Mardziel, Fangjing Wu, Preetam Amancharla, and Anupam Datta. 2018. [Gender bias in neural natural language processing](#). *Preprint*, arXiv:1807.11714.
- Ximing Lu, Sean Welleck, Peter West, Liwei Jiang, Jungo Kasai, Daniel Khashabi, Ronan Le Bras, Lianhui Qin, Youngjae Yu, Rowan Zellers, Noah A. Smith, and Yejin Choi. 2022. [NeuroLogic a*esque decoding: Constrained text generation with lookahead heuristics](#). In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 780–799, Seattle, United States. Association for Computational Linguistics.
- Agnes Luhtaru, Elizaveta Korotkova, and Mark Fishel. 2024. [No error left behind: Multilingual grammatical error correction with pre-trained translation models](#). In *Proceedings of the 18th Conference of the European Chapter of the Association for*

Computational Linguistics (Volume 1: Long Papers), pages 1209–1222, St. Julian's, Malta. Association for Computational Linguistics.

Yan Ma, Yu Qiao, and Pengfei Liu. 2024. [MoPS: Modular story premise synthesis for open-ended automatic story generation](#). In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2135–2169, Bangkok, Thailand. Association for Computational Linguistics.

Mohamed Maamouri, Ann Bies, Seth Kulick, Michael Ciul, Nizar Habash, and Ramy Eskander. 2014. [Developing an egyptian arabic treebank: Impact of dialectal morphology on annotation and tool development](#). In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC-2014)*. European Language Resources Association (ELRA).

Anne Maass and Luciano Arcuri. 1996. Language and stereotyping. *Stereotypes and stereotyping*, pages 193–226.

Nishtha Madaan, Sameep Mehta, Taneeja Agrawaal, Vrinda Malhotra, Aditi Aggarwal, Yatin Gupta, and Mayank Saxena. 2018. [Analyze, detect and remove gender stereotyping from bollywood movies](#). In *Proceedings of the 1st Conference on Fairness, Accountability and Transparency*, volume 81 of *Proceedings of Machine Learning Research*, pages 92–105. PMLR.

Nora Madi and Hend Al-Khalifa. 2020. [Error detection for Arabic text using neural sequence labeling](#). *Applied Sciences*, 10(15).

Jonathan Mallinson, Jakub Adamek, Eric Malmi, and Aliaksei Severyn. 2022. [EdiT5: Semi-autoregressive text editing with t5 warm-start](#). In *Findings of the Association*

- for Computational Linguistics: EMNLP 2022*, pages 2126–2138, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.
- Jonathan Mallinson, Aliaksei Severyn, Eric Malmi, and Guillermo Garrido. 2020. [FELIX: Flexible text editing through tagging and insertion](#). In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1244–1255, Online. Association for Computational Linguistics.
- Shervin Malmasi, Marcos Zampieri, Nikola Ljubešić, Preslav Nakov, Ahmed Ali, and Jörg Tiedemann. 2016. [Discriminating between similar languages and Arabic dialect identification: A report on the third DSL shared task](#). In *Proceedings of the Third Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial3)*, pages 1–14, Osaka, Japan. The COLING 2016 Organizing Committee.
- Eric Malmi, Sebastian Krause, Sascha Rothe, Daniil Mirylenka, and Aliaksei Severyn. 2019. [Encode, tag, realize: High-precision text editing](#). 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)*, pages 5054–5065, Hong Kong, China. Association for Computational Linguistics.
- Thomas Manzini, Lim Yao Chong, Alan W Black, and Yulia Tsvetkov. 2019. [Black is to criminal as Caucasian is to police: Detecting and removing multiclass bias in word embeddings](#). 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 615–621, Minneapolis, Minnesota. Association for Computational Linguistics.
- Rowan Hall Maudslay, Hila Gonen, Ryan Cotterell, and Simone Teufel. 2019. [It’s all in](#)

- the name: Mitigating gender bias with name-based counterfactual data substitution. 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)*, pages 5267–5275, Hong Kong, China. Association for Computational Linguistics.
- Kathleen McKeown. 1985. Text generation - using discourse strategies and focus constraints to generate natural language text. In *Studies in natural language processing*.
- Karen McNeil and Miled Faiza. 2011. Tunisian Arabic Corpus : Creating a Written Corpus of an "Unwritten" Language. In *Proceedings of the Workshop on Arabic Corpus Linguistics (WACL)*.
- Quinn McNemar. 1947. Note on the sampling error of the difference between correlated proportions or percentages. *Psychometrika*, 12(2):153–157.
- Ninareh Mehrabi, Thamme Gowda, Fred Morstatter, Nanyun Peng, and Aram Galstyan. 2019. Man is to person as woman is to location: Measuring gender bias in named entity recognition. *Preprint*, arXiv:1910.10872.
- Michela Menegatti and Monica Rubini. 2017. Gender bias and sexism in language. In *Oxford Research Encyclopedia of Communication*. Oxford University Press.
- Stuart Mesham, Christopher Bryant, Marek Rei, and Zheng Yuan. 2023. An extended sequence tagging vocabulary for grammatical error correction. In *Findings of the Association for Computational Linguistics: EACL 2023*, pages 1608–1619, Dubrovnik, Croatia. Association for Computational Linguistics.
- Rada Mihalcea and Ted Pedersen. 2003. An evaluation exercise for word alignment. In

Proceedings of the HLT-NAACL 2003 Workshop on Building and Using Parallel Texts: Data Driven Machine Translation and Beyond, pages 1–10.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. [Efficient estimation of word representations in vector space](#). *Preprint*, arXiv:1301.3781.

Masato Mita, Keisuke Sakaguchi, Masato Hagiwara, Tomoya Mizumoto, Jun Suzuki, and Kentaro Inui. 2024. [Towards automated document revision: Grammatical error correction, fluency edits, and beyond](#). In *Proceedings of the 19th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2024)*, pages 251–265, Mexico City, Mexico. Association for Computational Linguistics.

Behrang Mohit, Alla Rozovskaya, Nizar Habash, Wajdi Zaghoulani, and Ossama Obeid. 2014a. The first QALB shared task on automatic text correction for Arabic. In *Proceedings of the Workshop for Arabic Natural Language Processing (WANLP)*, pages 39–47, Doha, Qatar.

Behrang Mohit, Alla Rozovskaya, Nizar Habash, Wajdi Zaghoulani, and Ossama Obeid. 2014b. [The first QALB shared task on automatic text correction for Arabic](#). In *Proceedings of the EMNLP 2014 Workshop on Arabic Natural Language Processing (ANLP)*, pages 39–47, Doha, Qatar. Association for Computational Linguistics.

Giovanni Molina, Fahad AlGhamdi, Mahmoud Ghoneim, Abdelati Hawwari, Nicolas Rey-Villamizar, Mona Diab, and Tamar Solorio. 2016. [Overview for the second shared task on language identification in code-switched data](#). In *Proceedings of the Second Workshop on Computational Approaches to Code Switching*, pages 40–49, Austin, Texas. Association for Computational Linguistics.

Amit Moryossef, Roei Aharoni, and Yoav Goldberg. 2019. [Filling gender & number](#)

gaps in neural machine translation with black-box context injection. In *Proceedings of the First Workshop on Gender Bias in Natural Language Processing*, pages 49–54, Florence, Italy. Association for Computational Linguistics.

Niklas Muennighoff, Thomas Wang, Lintang Sutawika, Adam Roberts, Stella Biderman, Teven Le Scao, M Saiful Bari, Sheng Shen, Zheng Xin Yong, Hailey Schoelkopf, Xiangru Tang, Dragomir Radev, Alham Fikri Aji, Khalid Almubarak, Samuel Albanie, Zaid Alyafeai, Albert Webson, Edward Raff, and Colin Raffel. 2023. [Crosslingual generalization through multitask finetuning](#). In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15991–16111, Toronto, Canada. Association for Computational Linguistics.

Benjamin Muller, Belen Alastruey, Prangthip Hansanti, Elahe Kalbassi, Christophe Ropers, Eric Smith, Adina Williams, Luke Zettlemoyer, Pierre Andrews, and Marta R. Costa-jussà. 2023. [The gender-GAP pipeline: A gender-aware polyglot pipeline for gender characterisation in 55 languages](#). In *Proceedings of the Eighth Conference on Machine Translation*, pages 536–550, Singapore. Association for Computational Linguistics.

El Moatez Billah Nagoudi, AbdelRahim Elmadany, and Muhammad Abdul-Mageed. 2022. [AraT5: Text-to-text transformers for Arabic language generation](#). In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 628–647, Dublin, Ireland. Association for Computational Linguistics.

Jakub Náplava and Milan Straka. 2019. [Grammatical error correction in low-resource scenarios](#). In *Proceedings of the 5th Workshop on Noisy User-generated Text (W-NUT 2019)*, pages 346–356, Hong Kong, China. Association for Computational Linguistics.

- Jakub Náplava, Milan Straka, Jana Straková, and Alexandr Rosen. 2022. [Czech grammar error correction with a large and diverse corpus](#). *Transactions of the Association for Computational Linguistics*, 10:452–467.
- Courtney Napoles, Keisuke Sakaguchi, Matt Post, and Joel Tetreault. 2015. [Ground truth for grammatical error correction metrics](#). In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 588–593, Beijing, China. Association for Computational Linguistics.
- Michael Nawar. 2015. [CUFE@QALB-2015 shared task: Arabic error correction system](#). In *Proceedings of the Second Workshop on Arabic Natural Language Processing*, pages 133–137, Beijing, China. Association for Computational Linguistics.
- Hwee Tou Ng, Siew Mei Wu, Ted Briscoe, Christian Hadiwinoto, Raymond Hendy Susanto, and Christopher Bryant. 2014. [The CoNLL-2014 shared task on grammatical error correction](#). In *Proceedings of the Eighteenth Conference on Computational Natural Language Learning: Shared Task*, pages 1–14, Baltimore, Maryland. Association for Computational Linguistics.
- Hwee Tou Ng, Siew Mei Wu, Yuanbin Wu, Christian Hadiwinoto, and Joel Tetreault. 2013. [The CoNLL-2013 shared task on grammatical error correction](#). In *Proceedings of the Seventeenth Conference on Computational Natural Language Learning: Shared Task*, pages 1–12, Sofia, Bulgaria. Association for Computational Linguistics.
- Xing Niu, Marianna Martindale, and Marine Carpuat. 2017. [A study of style in machine translation: Controlling the formality of machine translation output](#). In *Proceedings of*

the 2017 Conference on Empirical Methods in Natural Language Processing, pages 2814–2819, Copenhagen, Denmark. Association for Computational Linguistics.

Xing Niu, Sudha Rao, and Marine Carpuat. 2018. [Multi-task neural models for translating between styles within and across languages](#). In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1008–1021, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Jekaterina Novikova, Ondřej Dušek, and Verena Rieser. 2017. [The E2E dataset: New challenges for end-to-end generation](#). In *Proceedings of the 18th Annual SIGdial Meeting on Discourse and Dialogue*, pages 201–206, Saarbrücken, Germany. Association for Computational Linguistics.

Debora Nozza, Federico Bianchi, and Dirk Hovy. 2021. [HONEST: Measuring hurtful sentence completion in language models](#). In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2398–2406, Online. Association for Computational Linguistics.

Ossama Obeid, Go Inoue, and Nizar Habash. 2022. [Camelira: An Arabic multi-dialect morphological disambiguator](#). In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 319–326, Abu Dhabi, UAE. Association for Computational Linguistics.

Ossama Obeid, Nasser Zalmout, Salam Khalifa, Dima Taji, Mai Oudah, Bashar Alhafni, Go Inoue, Fadhl Eryani, Alexander Erdmann, and Nizar Habash. 2020. [CAMEL tools: An open source python toolkit for Arabic natural language processing](#). In *Proceedings*

of the Twelfth Language Resources and Evaluation Conference, pages 7022–7032, Marseille, France. European Language Resources Association.

Franz Josef Och and Hermann Ney. 2003. [A systematic comparison of various statistical alignment models](#). *Computational Linguistics*, 29(1):19–51.

Kostiantyn Omelianchuk, Vitaliy Atrasevych, Artem Chernodub, and Oleksandr Skurzhashkyi. 2020. [GECToR – grammatical error correction: Tag, not rewrite](#). In *Proceedings of the Fifteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 163–170, Seattle, WA, USA → Online. Association for Computational Linguistics.

Kostiantyn Omelianchuk, Andrii Liubonko, Oleksandr Skurzhashkyi, Artem Chernodub, Oleksandr Korniienko, and Igor Samokhin. 2024. [Pillars of grammatical error correction: Comprehensive inspection of contemporary approaches in the era of large language models](#). In *Proceedings of the 19th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2024)*, pages 17–33, Mexico City, Mexico. Association for Computational Linguistics.

OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mohammad Bavarian, Jeff Belgum, and 262 others. 2024. [Gpt-4 technical report](#). *Preprint*, arXiv:2303.08774.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: a Method for Automatic Evaluation of Machine Translation. In *Proceedings of the Conference*

of the Association for Computational Linguistics (ACL), pages 311–318, Philadelphia, Pennsylvania, USA.

Ankur Parikh, Xuezhi Wang, Sebastian Gehrmann, Manaal Faruqui, Bhuwan Dhingra, Diyi Yang, and Dipanjan Das. 2020. [ToTTo: A controlled table-to-text generation dataset](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1173–1186, Online. Association for Computational Linguistics.

Ji Ho Park, Jamin Shin, and Pascale Fung. 2018. [Reducing gender bias in abusive language detection](#). In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2799–2804, Brussels, Belgium. Association for Computational Linguistics.

Arfath Pasha, Mohamed Al-Badrashiny, Mona Diab, Ahmed El Kholy, Ramy Eskander, Nizar Habash, Manoj Pooleery, Owen Rambow, and Ryan Roth. 2014a. [MADAMIRA: A fast, comprehensive tool for morphological analysis and disambiguation of Arabic](#). In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 1094–1101, Reykjavik, Iceland. European Language Resources Association (ELRA).

Arfath Pasha, Mohamed Al-Badrashiny, Mona Diab, Ahmed El Kholy, Ramy Eskander, Nizar Habash, Manoj Pooleery, Owen Rambow, and Ryan Roth. 2014b. Madamira: A fast, comprehensive tool for morphological analysis and disambiguation of Arabic. In *Proceedings of the Language Resources and Evaluation Conference (LREC)*, pages 1094–1101, Reykjavik, Iceland.

Barun Patra, Saksham Singhal, Shaohan Huang, Zewen Chi, Li Dong, Furu Wei, Vishrav

- Chaudhary, and Xia Song. 2023. [Beyond English-centric bitexts for better multilingual language representation learning](#). In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15354–15373, Toronto, Canada. Association for Computational Linguistics.
- Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. [GloVe: Global vectors for word representation](#). In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.
- Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. [Deep contextualized word representations](#). 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 2227–2237, New Orleans, Louisiana. Association for Computational Linguistics.
- Jonas Pfeiffer, Ivan Vulić, Iryna Gurevych, and Sebastian Ruder. 2020. [MAD-X: An Adapter-Based Framework for Multi-Task Cross-Lingual Transfer](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7654–7673, Online. Association for Computational Linguistics.
- Andrea Piergentili, Beatrice Savoldi, Dennis Fucci, Matteo Negri, and Luisa Bentivogli. 2023. [Hi guys or hi folks? benchmarking gender-neutral machine translation with the GeNTE corpus](#). In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 14124–14140, Singapore. Association for Computational Linguistics.

Matt Post. 2018. [A call for clarity in reporting BLEU scores](#). In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Brussels, Belgium. Association for Computational Linguistics.

Ella Rabinovich, Raj Nath Patel, Shachar Mirkin, Lucia Specia, and Shuly Wintner. 2017. [Personalized machine translation: Preserving original author traits](#). In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 1074–1084, Valencia, Spain. Association for Computational Linguistics.

Dragomir R. Radev, Eduard Hovy, and Kathleen McKeown. 2002. [Introduction to the special issue on summarization](#). *Computational Linguistics*, 28(4):399–408.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. [Language models are unsupervised multitask learners](#). *OpenAI*. Accessed: 2024-11-15.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. [Exploring the limits of transfer learning with a unified text-to-text transformer](#). *Journal of Machine Learning Research*, 21(140):1–67.

Vipul Raheja, Dimitris Alikaniotis, Vivek Kulkarni, Bashar Alhafni, and Dhruv Kumar. 2024. [mEdit: Multilingual text editing via instruction tuning](#). In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 979–1001, Mexico City, Mexico. Association for Computational Linguistics.

Vipul Raheja, Dhruv Kumar, Ryan Koo, and Dongyeop Kang. 2023. [CoEdit: Text](#)

- editing by task-specific instruction tuning. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 5274–5291, Singapore. Association for Computational Linguistics.
- Ahad Rana. 2010. [Common crawl—building an open web-scale crawl using hadoop](#). Accessed on January, 2025.
- Francisco Rangel, Paolo Rosso, Anis Charfi, Wajdi Zaghoulani, Bilal Ghanem, and Javier Sánchez-Junquera. 2019. On the author profiling and deception detection in Arabic shared task at FIRE. In *Proceedings of the 11th Annual Meeting of the Forum for Information Retrieval Evaluation*, pages 7–9.
- Gaoqi Rao, Erhong Yang, and Baolin Zhang. 2020. [Overview of NLPTEA-2020 shared task for Chinese grammatical error diagnosis](#). In *Proceedings of the 6th Workshop on Natural Language Processing Techniques for Educational Applications*, pages 25–35, Suzhou, China. Association for Computational Linguistics.
- Marek Rei and Helen Yannakoudakis. 2016. [Compositional sequence labeling models for error detection in learner writing](#). In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1181–1191, Berlin, Germany. Association for Computational Linguistics.
- Ehud Reiter and Robert Dale. 1997. [Building applied natural language generation systems](#). *Natural Language Engineering*, 3(1):57–87.
- Sascha Rothe, Jonathan Mallinson, Eric Malmi, Sebastian Krause, and Aliaksei Severyn. 2021. [A simple recipe for multilingual grammatical error correction](#). In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the*

11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers), pages 702–707, Online. Association for Computational Linguistics.

Alla Rozovskaya, Houda Bouamor, Nizar Habash, Wajdi Zaghoulani, Ossama Obeid, and Behrang Mohit. 2015a. [The second QALB shared task on automatic text correction for Arabic](#). In *Proceedings of the Second Workshop on Arabic Natural Language Processing*, pages 26–35, Beijing, China. Association for Computational Linguistics.

Alla Rozovskaya, Houda Bouamor, Nizar Habash, Wajdi Zaghoulani, Ossama Obeid, and Behrang Mohit. 2015b. The second QALB shared task on automatic text correction for arabic. In *Proceedings of the Workshop for Arabic Natural Language Processing (WANLP)*, pages 26–35, Beijing, China.

Alla Rozovskaya, Nizar Habash, Ramy Eskander, Noura Farra, and Wael Salloum. 2014. [The Columbia system in the QALB-2014 shared task on Arabic error correction](#). In *Proceedings of the EMNLP 2014 Workshop on Arabic Natural Language Processing (ANLP)*, pages 160–164, Doha, Qatar. Association for Computational Linguistics.

Alla Rozovskaya and Dan Roth. 2013. [Joint learning and inference for grammatical error correction](#). In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 791–802, Seattle, Washington, USA. Association for Computational Linguistics.

Alla Rozovskaya and Dan Roth. 2019. [Grammar error correction in morphologically rich languages: The case of Russian](#). *Transactions of the Association for Computational Linguistics*, 7:1–17.

Rachel Rudinger, Jason Naradowsky, Brian Leonard, and Benjamin Van Durme. 2018. [Gender bias in coreference resolution](#). In *Proceedings of the 2018 Conference of the*

North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers), pages 8–14, New Orleans, Louisiana. Association for Computational Linguistics.

Alexander M. Rush, Sumit Chopra, and Jason Weston. 2015. [A neural attention model for abstractive sentence summarization](#). In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 379–389, Lisbon, Portugal. Association for Computational Linguistics.

Houda Saadane and Nizar Habash. 2015. A Conventional Orthography for Algerian Arabic. In *Proceedings of the Second Workshop on Arabic Natural Language Processing*, page 69, Beijing, China.

Fatiha Sadat, Farzindar Kazemi, and Atefeh Farzindar. 2014. [Automatic identification of Arabic language varieties and dialects in social media](#). In *Proceedings of the Second Workshop on Natural Language Processing for Social Media (SocialNLP)*, pages 22–27, Dublin, Ireland. Association for Computational Linguistics and Dublin City University.

Ali Safaya, Moutasem Abdullatif, and Deniz Yuret. 2020. [KUISAIL at SemEval-2020 task 12: BERT-CNN for offensive speech identification in social media](#). In *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, pages 2054–2059, Barcelona (online). International Committee for Computational Linguistics.

Abdulwahab Sahyoun and Shady Shehata. 2023. [AraDiaWER: An explainable metric for dialectal Arabic ASR](#). In *Proceedings of the Second Workshop on NLP Applications to Field Linguistics*, pages 64–73, Dubrovnik, Croatia. Association for Computational Linguistics.

Ahmed Salama, Houda Bouamor, Behrang Mohit, and Kemal Oflazer. 2014. [YouDACC: the Youtube dialectal Arabic comment corpus](#). In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 1246–1251, Reykjavik, Iceland. European Language Resources Association (ELRA).

Mohammad Salameh, Houda Bouamor, and Nizar Habash. 2018. [Fine-grained Arabic dialect identification](#). In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1332–1344, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Julian Salazar, Davis Liang, Toan Q. Nguyen, and Katrin Kirchhoff. 2020. [Masked language model scoring](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2699–2712, Online. Association for Computational Linguistics.

Wael Salloum, Heba Elfardy, Linda Alamir-Salloum, Nizar Habash, and Mona Diab. 2014. [Sentence level dialect identification for machine translation system selection](#). In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 772–778, Baltimore, Maryland. Association for Computational Linguistics.

Wael Salloum and Nizar Habash. 2011. Dialectal to Standard Arabic Paraphrasing to Improve Arabic-English Statistical Machine Translation. In *Proceedings of the First Workshop on Algorithms and Resources for Modelling of Dialects and Language Varieties*, pages 10–21, Edinburgh, Scotland.

Wael Salloum and Nizar Habash. 2012. Elissa: A Dialectal to Standard Arabic Machine

- Translation System. In *Proceedings of the International Conference on Computational Linguistics (COLING)*, pages 385–392, Mumbai, India.
- Eduardo Sánchez, Pierre Andrews, Pontus Stenetorp, Mikel Artetxe, and Marta R. Costa-jussà. 2024. [Gender-specific machine translation with large language models](#). In *Proceedings of the Fourth Workshop on Multilingual Representation Learning (MRL 2024)*, pages 148–158, Miami, Florida, USA. Association for Computational Linguistics.
- Danielle Saunders and Bill Byrne. 2020. [Reducing gender bias in neural machine translation as a domain adaptation problem](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7724–7736, Online. Association for Computational Linguistics.
- Danielle Saunders, Rosie Sallis, and Bill Byrne. 2020. [Neural machine translation doesn’t translate gender coreference right unless you make it](#). In *Proceedings of the Second Workshop on Gender Bias in Natural Language Processing*, pages 35–43, Barcelona, Spain (Online). Association for Computational Linguistics.
- Danielle Saunders, Rosie Sallis, and Bill Byrne. 2021. [First the worst: Finding better gender translations during beam search](#). *Preprint*, arXiv:2104.07429.
- Beatrice Savoldi, Eleonora Cupin, Manjinder Thind, Anne Lauscher, Andrea Piergentili, Matteo Negri, and Luisa Bentivogli. 2025. [mgente: A multilingual resource for gender-neutral language and translation](#). *Preprint*, arXiv:2501.09409.
- Beatrice Savoldi, Marco Gaido, Luisa Bentivogli, Matteo Negri, and Marco Turchi. 2021. [Gender Bias in Machine Translation](#). *Transactions of the Association for Computational Linguistics*, 9:845–874.

Beatrice Savoldi, Marco Gaido, Matteo Negri, and Luisa Bentivogli. 2023. [Test suites task: Evaluation of gender fairness in MT with MuST-SHE and INES](#). In *Proceedings of the Eighth Conference on Machine Translation*, pages 252–262, Singapore. Association for Computational Linguistics.

Beatrice Savoldi, Sara Papi, Matteo Negri, Ana Guerbero-f-Arenas, and Luisa Bentivogli. 2024. [What the harm? quantifying the tangible impact of gender bias in machine translation with a human-centered study](#). In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 18048–18076, Miami, Florida, USA. Association for Computational Linguistics.

Indira Sen, Mattia Samory, Fabian Flöck, Claudia Wagner, and Isabelle Augenstein. 2021. [How does counterfactually augmented data impact models for social computing constructs?](#) In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 325–344, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Neha Sengupta, Sunil Kumar Sahu, Bokang Jia, Satheesh Katipomu, Haonan Li, Fajri Koto, William Marshall, Gurpreet Gosal, Cynthia Liu, Zhiming Chen, Osama Mohammed Afzal, Samta Kamboj, Onkar Pandit, Rahul Pal, Lalit Pradhan, Zain Muhammad Mujahid, Massa Baali, Xudong Han, Sondas Mahmoud Bsharat, and 13 others. 2023. [Jais and jais-chat: Arabic-centric foundation and instruction-tuned open generative large language models](#). *Preprint*, arXiv:2308.16149.

Rico Sennrich and Barry Haddow. 2016. Linguistic input features improve neural machine translation. In *Proceedings of the First Conference on Machine Translation: Volume 1, Research Papers*, volume 1, pages 83–91.

- Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016a. [Controlling politeness in neural machine translation via side constraints](#). In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 35–40, San Diego, California. Association for Computational Linguistics.
- Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016b. [Neural machine translation of rare words with subword units](#). In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725. Association for Computational Linguistics.
- K. Shaalan, H.M.A. Bakr, and I. Ziedan. 2007. Transferring Egyptian colloquial dialect into modern standard Arabic. In *Proceedings of the Conference on Recent Advances in Natural Language Processing (RANLP)*, pages 525–529.
- Ali Shazal, Aiza Usman, and Nizar Habash. 2020. [A unified model for Arabizi detection and transliteration using sequence-to-sequence models](#). In *Proceedings of the Fifth Arabic Natural Language Processing Workshop*, pages 167–177, Barcelona, Spain (Online). Association for Computational Linguistics.
- Emily Sheng, Josh Arnold, Zhou Yu, Kai-Wei Chang, and Nanyun Peng. 2021a. [Revealing persona biases in dialogue systems](#). *Preprint*, arXiv:2104.08728.
- Emily Sheng, Kai-Wei Chang, Premkumar Natarajan, and Nanyun Peng. 2019. [The woman worked as a babysitter: On biases in language generation](#). 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)*, pages 3407–3412, Hong Kong, China. Association for Computational Linguistics.

- Emily Sheng, Kai-Wei Chang, Premkumar Natarajan, and Nanyun Peng. 2021b. "[nice try, kiddo](#)": Investigating ad hominem in dialogue responses. *Preprint*, arXiv:2010.12820.
- Stuart M. Shieber, Gertjan van Noord, Robert C. Moore, and Fernando C. N. Pereira. 1989. [A semantic-head-driven generation algorithm for unification-based formalisms](#). In *27th Annual Meeting of the Association for Computational Linguistics*, pages 7–17, Vancouver, British Columbia, Canada. Association for Computational Linguistics.
- Frank Smadja and Kathleen McKeown. 1994. [Translating collocations for use in bilingual lexicons](#). In *Human Language Technology: Proceedings of a Workshop held at Plainsboro, New Jersey, March 8-11, 1994*.
- Kamel Smaïli, Mourad Abbas, Karima Meftouh, and Salima Harrat. 2014. Building resources for Algerian Arabic dialects. In *Proceedings of the Conference of the International Speech Communication Association (Interspeech)*.
- Thamar Solorio, Elizabeth Blair, Suraj Maharjan, Steven Bethard, Mona Diab, Mahmoud Ghoneim, Abdelati Hawwari, Fahad AlGhamdi, Julia Hirschberg, Alison Chang, and Pascale Fung. 2014. [Overview for the first shared task on language identification in code-switched data](#). In *Proceedings of the First Workshop on Computational Approaches to Code Switching*, pages 62–72, Doha, Qatar. Association for Computational Linguistics.
- Aiman Solyman, Zhenyu Wang, Qian Tao, Arafat Abdulgader Mohammed Elhag, Rui Zhang, and Zeinab Mahmoud. 2022. [Automatic Arabic grammatical error correction based on expectation-maximization routing and target-bidirectional agreement](#). *Knowledge-Based Systems*, 241:108180.
- Aiman Solyman, Marco Zappatore, Wang Zhenyu, Zeinab Mahmoud, Ali Alfatemi,

- Ashraf Osman Ibrahim, and Lubna Abdelkareim Gabralla. 2023. [Optimizing the impact of data augmentation for low-resource grammatical error correction](#). *Journal of King Saud University - Computer and Information Sciences*, 35(6):101572.
- Aiman Solyman, Wang Zhenyu, Tao Qian, Arafat Abdulgader Mohammed Elhag, Muhammad Toseef, and Zeinab Aleibeid. 2021. [Synthetic data with neural machine translation for automatic correction in Arabic grammar](#). *Egyptian Informatics Journal*, 22(3):303–315.
- Artūrs Stāfanovičs, Toms Bergmanis, and Mārcis Pinnis. 2020. [Mitigating gender bias in machine translation with target gender annotations](#). *Preprint*, arXiv:2010.06203.
- Felix Stahlberg and Shankar Kumar. 2020. [Seq2Edits: Sequence transduction using span-level edit operations](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5147–5159, Online. Association for Computational Linguistics.
- Felix Stahlberg and Shankar Kumar. 2021. [Synthetic data generation for grammatical error correction with tagged corruption models](#). In *Proceedings of the 16th Workshop on Innovative Use of NLP for Building Educational Applications*, pages 37–47, Online. Association for Computational Linguistics.
- Karolina Stanczak and Isabelle Augenstein. 2021. [A survey on gender bias in natural language processing](#). *Preprint*, arXiv:2112.14168.
- Gabriel Stanovsky, Noah A. Smith, and Luke Zettlemoyer. 2019. [Evaluating gender bias in machine translation](#). In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1679–1684, Florence, Italy. Association for Computational Linguistics.

- Milan Straka, Jakub Náplava, and Jana Straková. 2021. [Character transformations for non-autoregressive GEC tagging](#). In *Proceedings of the Seventh Workshop on Noisy User-generated Text (W-NUT 2021)*, pages 417–422, Online. Association for Computational Linguistics.
- Nishant Subramani, Nivedita Suresh, and Matthew Peters. 2022. [Extracting latent steering vectors from pretrained language models](#). In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 566–581, Dublin, Ireland. Association for Computational Linguistics.
- Jiao Sun, Yufei Tian, Wangchunshu Zhou, Nan Xu, Qian Hu, Rahul Gupta, John Wieting, Nanyun Peng, and Xuezhe Ma. 2023. [Evaluating large language models on controlled generation tasks](#). In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 3155–3168, Singapore. Association for Computational Linguistics.
- Tony Sun, Andrew Gaut, Shirlyn Tang, Yuxin Huang, Mai ElSherief, Jieyu Zhao, Diba Mirza, Elizabeth Belding, Kai-Wei Chang, and William Yang Wang. 2019. [Mitigating gender bias in natural language processing: Literature review](#). In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1630–1640, Florence, Italy. Association for Computational Linguistics.
- Tony Sun, Kellie Webster, Apu Shah, William Yang Wang, and Melvin Johnson. 2021. [They, them, theirs: Rewriting with gender-neutral english](#). *Preprint*, arXiv:2102.06788.
- Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. [Sequence to sequence learning with neural networks](#). In *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc.

- Oleksiy Syvokon, Olena Nahorna, Pavlo Kuchmiichuk, and Nastasiia Osidach. 2023. [UA-GEC: Grammatical error correction and fluency corpus for the Ukrainian language](#). In *Proceedings of the Second Ukrainian Natural Language Processing Workshop (UNLP)*, pages 96–102, Dubrovnik, Croatia. Association for Computational Linguistics.
- Dima Taji, Jamila El Gizuli, and Nizar Habash. 2018a. An Arabic dependency treebank in the travel domain. In *Proceedings of the Workshop on Open-Source Arabic Corpora and Processing Tools (OSACT)*, Miyazaki, Japan.
- Dima Taji, Salam Khalifa, Ossama Obeid, Fadhl Eryani, and Nizar Habash. 2018b. An Arabic Morphological Analyzer and Generator with Copious Features. In *Proceedings of the Fifteenth Workshop on Computational Research in Phonetics, Phonology, and Morphology (SIGMORPHON)*, pages 140–150.
- Maksym Tarnavskiy, Artem Chernodub, and Kostiantyn Omelianchuk. 2022. [Ensembling and knowledge distilling of large sequence taggers for grammatical error correction](#). In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3842–3852, Dublin, Ireland. Association for Computational Linguistics.
- Fanar Team, Ummar Abbas, Mohammad Shahmeer Ahmad, Firoj Alam, Enes Altinisik, Ehsannedin Asgari, Yazan Boshmaf, Sabri Boughorbel, Sanjay Chawla, Shammur Chowdhury, Fahim Dalvi, Kareem Darwish, Nadir Durrani, Mohamed Elfeky, Ahmed Elmagarmid, Mohamed Eltabakh, Masoomali Fatehkia, Anastasios Fragkopoulos, Maram Hasanain, and 23 others. 2025. [Fanar: An arabic-centric multimodal generative ai platform](#). *Preprint*, arXiv:2501.13944.
- Joel R. Tetreault and Martin Chodorow. 2008. [The ups and downs of preposition error](#)

- detection in ESL writing. In *Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008)*, pages 865–872, Manchester, UK. Coling 2008 Organizing Committee.
- Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, and 49 others. 2023. [Llama 2: Open foundation and fine-tuned chat models](#). *Preprint*, arXiv:2307.09288.
- Severino Trotta, Lucie Flek, and Charles Welch. 2022. [Nearest neighbor language models for stylistic controllable generation](#). In *Proceedings of the 2nd Workshop on Natural Language Generation, Evaluation, and Metrics (GEM)*, pages 295–305, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.
- A. M. Turing. 1950. [Computing machinery and intelligence](#). *Mind*, LIX(236):433–460.
- Houcemeddine Turki, Emad Adel, Tariq Daouda, and Nassim Regragui. 2016. A Conventional Orthography for Maghrebi Arabic. In *Proceedings of the Language Resources and Evaluation Conference (LREC)*, Portorož, Slovenia.
- Alexander Matt Turner, Lisa Thiergart, Gavin Leech, David Udell, Juan J. Vazquez, Ulisse Mini, and Monte MacDiarmid. 2024. [Steering language models with activation engineering](#). *Preprint*, arXiv:2308.10248.
- Bhargav Upadhyay, Akhilesh Sudhakar, and Arjun Maheswaran. 2022. [Efficient reinforcement learning for unsupervised controlled text generation](#). *Preprint*, arXiv:2204.07696.
- Eva Vanmassenhove, Chris Emmery, and Dimitar Shterionov. 2021. [NeuTral Rewriter: A rule-based and neural approach to automatic rewriting into gender neutral alter-](#)

- natives. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 8940–8948, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Eva Vanmassenhove, Christian Hardmeier, and Andy Way. 2018. [Getting gender right in neural machine translation](#). In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3003–3008, Brussels, Belgium. Association for Computational Linguistics.
- Eva Vanmassenhove and Johanna Monti. 2021. [gENDER-IT: An annotated English-Italian parallel challenge set for cross-linguistic natural gender phenomena](#). In *Proceedings of the 3rd Workshop on Gender Bias in Natural Language Processing*, pages 1–7, Online. Association for Computational Linguistics.
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. [Attention is all you need](#). In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.
- Jesse Vig, Sebastian Gehrmann, Yonatan Belinkov, Sharon Qian, Daniel Nevo, Yaron Singer, and Stuart Shieber. 2020. [Investigating gender bias in language models using causal mediation analysis](#). In *Advances in Neural Information Processing Systems*, volume 33, pages 12388–12401. Curran Associates, Inc.
- Zhaohong Wan, Xiaojun Wan, and Wenguang Wang. 2020. [Improving grammatical error correction with data augmentation by editing latent representation](#). In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 2202–2212, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Daniel Watson, Nasser Zalmout, and Nizar Habash. 2018a. [Utilizing character and word embeddings for text normalization with sequence-to-sequence models](#). In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 837–843, Brussels, Belgium. Association for Computational Linguistics.

Daniel Watson, Nasser Zalmout, and Nizar Habash. 2018b. [Utilizing character and word embeddings for text normalization with sequence-to-sequence models](#). In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 837–843, Brussels, Belgium. Association for Computational Linguistics.

Janet CE Watson. 2007. *The Phonology and Morphology of Arabic*. Oxford University Press.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. 2023. [Chain-of-thought prompting elicits reasoning in large language models](#). *Preprint*, arXiv:2201.11903.

Joseph Weizenbaum. 1966. [Eliza—a computer program for the study of natural language communication between man and machine](#). *Communications of the ACM*, 9:36 – 45.

Sam Wiseman, Stuart Shieber, and Alexander Rush. 2017. [Challenges in data-to-document generation](#). In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2253–2263, Copenhagen, Denmark. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, and 3 others. 2020. [Transformers: State-of-the-art](#)

- natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.
- Haoran Wu, Wenxuan Wang, Yuxuan Wan, Wenxiang Jiao, and Michael Lyu. 2023. [Chatgpt or grammarly? evaluating chatgpt on grammatical error correction benchmark](#). *Preprint*, arXiv:2303.13648.
- Ziang Xie, Anand Avati, Naveen Arivazhagan, Dan Jurafsky, and Andrew Y. Ng. 2016. [Neural language correction with character-based attention](#). *Preprint*, arXiv:1603.09727.
- Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2021. [mT5: A massively multilingual pre-trained text-to-text transformer](#). In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 483–498, Online. Association for Computational Linguistics.
- Hemant Yadav and Sunayana Sitaram. 2022. [A survey of multilingual models for automatic speech recognition](#). In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 5071–5079, Marseille, France. European Language Resources Association.
- Daiki Yanamoto, Tomoki Ikawa, Tomoyuki Kajiwara, Takashi Ninomiya, Satoru Uchida, and Yuki Arase. 2022. [Controllable text simplification with deep reinforcement learning](#). In *Proceedings of the 2nd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 12th International Joint Conference*

on Natural Language Processing (Volume 2: Short Papers), pages 398–404, Online only. Association for Computational Linguistics.

Kevin Yang and Dan Klein. 2021. [FUDGE: Controlled text generation with future discriminators](#). In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3511–3535, Online. Association for Computational Linguistics.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019a. [Xlnet: Generalized autoregressive pretraining for language understanding](#). In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019b. [Xlnet: Generalized autoregressive pretraining for language understanding](#). In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Helen Yannakoudakis, Ted Briscoe, and Ben Medlock. 2011. [A new dataset and method for automatically grading ESOL texts](#). In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 180–189, Portland, Oregon, USA. Association for Computational Linguistics.

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

Zheng Yuan and Christopher Bryant. 2021. [Document-level grammatical error correction](#).

- In *Proceedings of the 16th Workshop on Innovative Use of NLP for Building Educational Applications*, pages 75–84, Online. Association for Computational Linguistics.
- Zheng Yuan, Felix Stahlberg, Marek Rei, Bill Byrne, and Helen Yannakoudakis. 2019. [Neural and FST-based approaches to grammatical error correction](#). In *Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 228–239, Florence, Italy. Association for Computational Linguistics.
- Zheng Yuan, Shiva Taslimipoor, Christopher Davis, and Christopher Bryant. 2021. [Multi-class grammatical error detection for correction: A tale of two systems](#). In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 8722–8736, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Wajdi Zaghouani and Dana Awad. 2016. Toward an Arabic punctuated corpus: Annotation guidelines and evaluation. In *The 2nd Workshop on Arabic Corpora and Processing Tools 2016 Theme: Social Media*, page 22.
- Wajdi Zaghouani and Anis Charfi. 2018a. [Arap-tweet: A large multi-dialect Twitter corpus for gender, age and language variety identification](#). In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).
- Wajdi Zaghouani and Anis Charfi. 2018b. ArapTweet: A Large Multi-Dialect Twitter Corpus for Gender, Age and Language Variety Identification. In *Proceedings of the Language Resources and Evaluation Conference (LREC)*, Miyazaki, Japan.
- Wajdi Zaghouani, Nizar Habash, Houda Bouamor, Alla Rozovskaya, Behrang Mohit, Abeer Heider, and Kemal Oflazer. 2015a. Correction annotation for non-native arabic

- texts: Guidelines and corpus. In *Proceedings of the Linguistic Annotation Workshop (LAW)*, pages 129–139.
- Wajdi Zaghoulani, Nizar Habash, Houda Bouamor, Alla Rozovskaya, Behrang Mohit, Abeer Heider, and Kemal Oflazer. 2015b. [Correction annotation for non-native Arabic texts: Guidelines and corpus](#). In *Proceedings of the 9th Linguistic Annotation Workshop*, pages 129–139, Denver, Colorado, USA. Association for Computational Linguistics.
- Wajdi Zaghoulani, Behrang Mohit, Nizar Habash, Ossama Obeid, Nadi Tomeh, Alla Rozovskaya, Noura Farra, Sarah Alkuhlani, and Kemal Oflazer. 2014. [Large scale Arabic error annotation: Guidelines and framework](#). In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, Reykjavik, Iceland. European Language Resources Association (ELRA).
- Omar F. Zaidan and Chris Callison-Burch. 2011. [The Arabic online commentary dataset: an annotated dataset of informal Arabic with high dialectal content](#). In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 37–41, Portland, Oregon, USA. Association for Computational Linguistics.
- Omar F. Zaidan and Chris Callison-Burch. 2014. [Arabic dialect identification](#). *Computational Linguistics*, 40(1):171–202.
- Nasser Zalmout. 2020. [Morphological Tagging and Disambiguation in Dialectal Arabic Using Deep Learning Architectures](#). Ph.D. thesis. Copyright - Database copyright ProQuest LLC; ProQuest does not claim copyright in the individual underlying works; Last updated - 2023-06-21.

- Nasser Zalmout, Alexander Erdmann, and Nizar Habash. 2018. [Noise-robust morphological disambiguation for dialectal Arabic](#). 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 953–964, New Orleans, Louisiana. Association for Computational Linguistics.
- Nasser Zalmout and Nizar Habash. 2019. [Adversarial multitask learning for joint multi-feature and multi-dialect morphological modeling](#). In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1775–1786, Florence, Italy. Association for Computational Linguistics.
- Nasser Zalmout and Nizar Habash. 2020. [Joint diacritization, lemmatization, normalization, and fine-grained morphological tagging](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8297–8307, Online. Association for Computational Linguistics.
- Marcos Zampieri, Shervin Malmasi, Nikola Ljubešić, Preslav Nakov, Ahmed Ali, Jörg Tiedemann, Yves Scherrer, and Noëmi Aeppli. 2017. [Findings of the VarDial evaluation campaign 2017](#). In *Proceedings of the Fourth Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial)*, pages 1–15, Valencia, Spain. Association for Computational Linguistics.
- Marcos Zampieri, Shervin Malmasi, Preslav Nakov, Ahmed Ali, Suwon Shon, James Glass, Yves Scherrer, Tanja Samardžić, Nikola Ljubešić, Jörg Tiedemann, Chris van der Lee, Stefan Grondelaers, Nelleke Oostdijk, Dirk Speelman, Antal van den Bosch, Ritesh Kumar, Bornini Lahiri, and Mayank Jain. 2018. [Language identification and morphosyntactic tagging: The second VarDial evaluation campaign](#). In *Proceedings of the Fifth Workshop on NLP for Similar Languages, Varieties and*

- Dialects (VarDial 2018)*, pages 1–17, Santa Fe, New Mexico, USA. Association for Computational Linguistics.
- Rabih Zbib, Erika Malchiodi, Jacob Devlin, David Stallard, Spyros Matsoukas, Richard Schwartz, John Makhoul, Omar F. Zaidan, and Chris Callison-Burch. 2012. Machine Translation of Arabic Dialects. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, pages 49–59, Montréal, Canada.
- Hanqing Zhang and Dawei Song. 2022. [DisCup: Discriminator cooperative unlikelihood prompt-tuning for controllable text generation](#). In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 3392–3406, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.
- Yu Zhang, Yue Zhang, Leyang Cui, and Guohong Fu. 2023. [Non-autoregressive text editing with copy-aware latent alignments](#). In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 7075–7085, Singapore. Association for Computational Linguistics.
- Zhehao Zhang, Ryan A. Rossi, Branislav Kveton, Yijia Shao, Diyi Yang, Hamed Zamani, Franck Dernoncourt, Joe Barrow, Tong Yu, Sungchul Kim, Ruiyi Zhang, Jiuxiang Gu, Tyler Derr, Hongjie Chen, Junda Wu, Xiang Chen, Zichao Wang, Subrata Mitra, Nedim Lipka, and 2 others. 2024. [Personalization of large language models: A survey](#). *Preprint*, arXiv:2411.00027.
- Jieyu Zhao, Subhabrata Mukherjee, Saghar Hosseini, Kai-Wei Chang, and Ahmed Hassan Awadallah. 2020. [Gender bias in multilingual embeddings and cross-lingual transfer](#). In *Proceedings of the 58th Annual Meeting of the Association for Com-*

- putational Linguistics*, pages 2896–2907, Online. Association for Computational Linguistics.
- Jieyu Zhao, Tianlu Wang, Mark Yatskar, Vicente Ordonez, and Kai-Wei Chang. 2018a. [Gender bias in coreference resolution: Evaluation and debiasing methods](#). In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 15–20, New Orleans, Louisiana. Association for Computational Linguistics.
- Jieyu Zhao, Yichao Zhou, Zeyu Li, Wei Wang, and Kai-Wei Chang. 2018b. [Learning gender-neutral word embeddings](#). In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4847–4853, Brussels, Belgium. Association for Computational Linguistics.
- Wei Zhao, Liang Wang, Kewei Shen, Ruoyu Jia, and Jingming Liu. 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.
- Yuanyuan Zhao, Nan Jiang, Weiwei Sun, and Xiaojun Wan. 2018c. Overview of the nlpcc 2018 shared task: Grammatical error correction. In *Natural Language Processing and Chinese Computing*.
- Houquan Zhou, Yumeng Liu, Zhenghua Li, Min Zhang, Bo Zhang, Chen Li, Ji Zhang, and Fei Huang. 2023a. [Improving Seq2Seq grammatical error correction via decoding](#)

[interventions](#). In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 7393–7405, Singapore. Association for Computational Linguistics.

Wangchunshu Zhou, Yuchen Eleanor Jiang, Ethan Wilcox, Ryan Cotterell, and Mrinmaya Sachan. 2023b. [Controlled text generation with natural language instructions](#). In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 42602–42613. PMLR.

Ran Zmigrod, Sabrina J. Mielke, Hanna Wallach, and Ryan Cotterell. 2019. [Counterfactual data augmentation for mitigating gender stereotypes in languages with rich morphology](#). In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1651–1661, Florence, Italy. Association for Computational Linguistics.

Inès Zribi, Rahma Boujelbane, Abir Masmoudi, Mariem Ellouze, Lamia Belguith, and Nizar Habash. 2014. A conventional orthography for Tunisian Arabic. In *Proceedings of the Language Resources and Evaluation Conference (LREC)*, Reykjavik, Iceland.