Part 2: Writing

Bashar Alhafni, New York University

Roadmap

- **Introduction**
- Data and Evaluation
- Approaches
- LLMs for Grammatical Error Correction
- Challenges and Future Work

Some slides are adapted from the COLING 2020 tutorial on GEC and the NAACL 2022 tutorial on text-editing

Grammatical Error Correction

I think, that everybody deserve privacy, including famous people. They can barelly breathing with all those photographers around them. I don’t know why people love spying famous people. And magazines are full of those things.

Grammatical Error Correction

I think**,** that everybody **deserve** privacy, including famous people. They can **barelly breathing** with all those photographers around them. I don’t know why people love **spying** famous people. And magazines are full of those things.

↓

I think that everybody **deserves** privacy, including famous people. They can **barely breath** with all those photographers around them. I don’t know why people love **spying on** famous people. And magazines are full of those things.

Grammatical Error Correction: Challenges

1. Multiple corrections are acceptable.
   - Above all, life is more important than {**secret**→**secrets**|**secrecy**|**a secret**}. {**In conclude**→**In conclusion**|**To conclude**}, social media benefit people.

2. Multiple errors may occur in a single sentence.
   - 19-58% of sentences in ESL (English as Second Language) corpora contain more than one error

3. Long-distance dependencies, including cross-sentence dependencies.
   - A subtle scent of red sweet apples and cinnamon sticks {**are**→**is**} present in the wine .

Grammatical Error Correction: Challenges

4. Some error types are more difficult to correct than others.
   - Closed-class error types (e.g. articles) vs. open-class errors.

5. Low frequency of errors.
   - Depending on the corpus, ESL corpora contain 6-15% erroneous words.
   - These numbers are lower for texts written by native speakers.

6. Error types and error distributions vary significantly among writers and datasets.

... and more.

Roadmap

- Introduction
- **Data and Evaluation**
- Approaches
- LLMs for Grammatical Error Correction
- Challenges and Future Work

Data and Evaluation: Data Annotation

Annotation goals:

- To build a parallel corpus of errors and their corrections
- To let us analyze error patterns
- Training/test data for machine learning

Sample annotation

Dear Paul
I haven't written to you for ages ~~**but**~~**because** *I was very busy* ~~**because of**~~**with** *the exams at the University. What about you? What's new in* ~~**Brazil?As**~~**Brazil?** *As you know, my friend John asked me to help him with the organization* ~~**at**~~***of*** *the concert, which was* ~~**performed**~~**held** *last month.*

Data and Evaluation: Data Annotation

Annotation Challenges:

- **Minimal vs. Fluent:**

Original: I want **explain to** you some interesting **part from** my experience.

Minimal: I want **to explain** to you some interesting **parts of** my experience.

Fluent: I want **to tell you about** some interesting **parts of** my experience.

Data and Evaluation: Corpora

English Corpora:

1. FCE (Yannakoudakis et a., 2011)
2. NUCLE (Dahlmeier et al., 2013)
3. CoNLL-2013 (Ng et al., 2013)
4. CoNLL-2014 (Ng et al., 2014)
5. Lang-8 (Mizumoto et al., 2011; Tajiri et al., 2012)
6. WikEd (Grundkiewicz and Junczys-Dowmnut 2014)
7. W&I-LOCNESS (BEA-2019) (Bryant et al., 2019)
8. CLC (Nicholls 2003)
9. JFLEG (Napoles et al., 2017)
10. GMEG (Napoles et al., 2019)

Data and Evaluation: Corpora

Non-English Corpora:

- **Arabic:**
  - QALB-2014, QALB-2015, ZAEBUC (Mohit et al., 2014; Rozovskaya et al., 2015, Habash et al., 2022)
- **Chinese:**
  - NLPTEA-2020, MuCGEC (Rao et al., 2020; Zhang et al., 2022)
- **Czech:**
  - AKCES-GEC, GECCC (Náplava & Straka, 2019; Náplava et al., 2022)
- **German:**
  - Falko-MERLIN (Boyd, 2018)
- **Japanese:**
  - TEC-JL (Suzuki et al., 2022)
- **Russian:**
  - RULEC-GEC (Rozovskaya & Roth, 2019)
- **Ukrainian:**
  - UA-GEC (Syvokon et al., 2023)

Data and Evaluation: Corpora

English Corpora:

1. FCE (Yannakoudakis et a., 2011)
2. NUCLE (Dahlmeier et al., 2013)
3. ***CoNLL-2013 (Ng et al., 2013)***
4. ***CoNLL-2014 (Ng et al., 2014)***
5. Lang-8 (Mizumoto et al., 2011; Tajiri et al., 2012)
6. WikEd (Grundkiewicz and Junczys-Dowmnut 2014)
7. ***W&I-LOCNESS (BEA-2019) (Bryant et al., 2019)***
8. CLC (Nicholls 2003)
9. ***JFLEG (Napoles et al., 2017)***
10. GMEG (Napoles et al., 2019)

Data and Evaluation: Corpora (CoNLL-2013/2014)

Name	**Conference on Natural Language Learning shared tasks**
Train	-
Dev	1.4k sentences (29k tokens) – CoNLL-2013
Test	1.3k sentences (30k tokens) – CoNLL-2014
Level	Upper Intermediate (C1)
Edits	Yes (28 types)
Domain	Essays
Authors	South-East Asian Undergraduates
Notes	CoNLL-2013 was originally a test set; CoNLL-2014 has 10 references (2 official, 8 extended); CoNLL-2014 is still a common benchmark; Very narrow domains: i) technology, ii) genetic testing;
Reference	Ng et al. (2013, 2014)

Data and Evaluation: Corpora (JFLEG)

Name	Johns Hopkins Fluency-Extended GUG Corpus
Train	-
Dev	754 sentences (14k tokens)
Test	747 sentences (14k tokens)
Level	Unknown
Edits	No
Domain	Essays
Authors	ESL learners
Notes	Advocated fluent over minimal corrections; 4 sets of references (both dev and test); Isolated sentences (not whole essays); Smallest test set;
Reference	Napoles et al. (2017)

Data and Evaluation: Corpora (W&I + LOCNESS)

Name	Cambridge English Write & Improve and LOCNESS
Train	34k sentences (628k tokens)
Dev	4.4k sentences (87k tokens)
Test	4.5k sentences (86k tokens)
Level	Beginner - Advanced (A1-C2), Native (LOCNESS)
Edits	Yes (55 types - automatic)
Domain	Short essays, letters, exams, web
Authors	International ESL learners
Notes	Native LOCNESS data only in dev and test; Balanced across all ability levels in terms of sentences; Released with the BEA-2019 shared task Official dev/test data of the BEA-2019 shared task; 5 sets of references in the test data;
Reference	Bryant et al. (2019)

Data and Evaluation: Evaluation Metrics

- Most commonly carried out in terms of edits

Original	I often look at TV	**Span-based**	**Span-based**	**Token-based**
Reference	[2, 4, watch]	**Correction**	**Detection**	**Detection**
Hypothesis-1	[2, 4, watch]	**Match**	**Match**	**Match**
Hypothesis-2	[2, 4, see]	**No match**	**Match**	**Match**
Hypothesis-3	[2, 3, watch]	**No match**	**No match**	**Match**

Data and Evaluation: Evaluation Metrics

- Most commonly carried out in terms of edits

Original	I often look at TV	Span-based	Span-based	Token-based
Reference	[2, 4, watch]	**Correction**	**Detection**	**Detection**
Hypothesis-1	[2, 4, watch]	Match	Match	Match
Hypothesis-2	[2, 4, see]	No match	Match	Match
Hypothesis-3	[2, 3, watch]	No match	No match	Match

Problem: unannotated hypothesis vs. annotated reference

Original	This is grammatical sentences .
Hypothesis	This **are** **a** grammatical sentences
Reference	This is a grammatical sentence .
Gold Edits	[2, 2, a], [3, 4, sentence]

Data and Evaluation: Evaluation Metrics – MaxMatch (M^2) Scorer

Reference: Dahlmeier and Ng (2012b)

Intuition:
- Align the original and hypothesis using Levenshtein
- Use TP, FP, FN to compute F-score

Official scorer of the CoNLL-2013/14 shared tasks.
Since CoNLL-2014, we use F0.5:
  - F0.5 weighs Precision twice as much as Recall
  - Still used today, notably on the CoNLL-2014 test set

Data and Evaluation: Evaluation Metrics – GLEU

Reference: Napoles et al. (2015)

Motivation: overcome the dependency on edits

Intuition:
- Inspired by BLEU n-gram matching
- Reward hyp n-grams that match ref, but not orig
- Penalize hyp n-grams that match orig, but not ref

Developed for fluency and JFLEG.
Often only reported on JFLEG.

Data and Evaluation: Evaluation Metrics – ERRANT

Reference: Bryant et al. (2017)

Motivation: facilitate error type scores

Intuition:

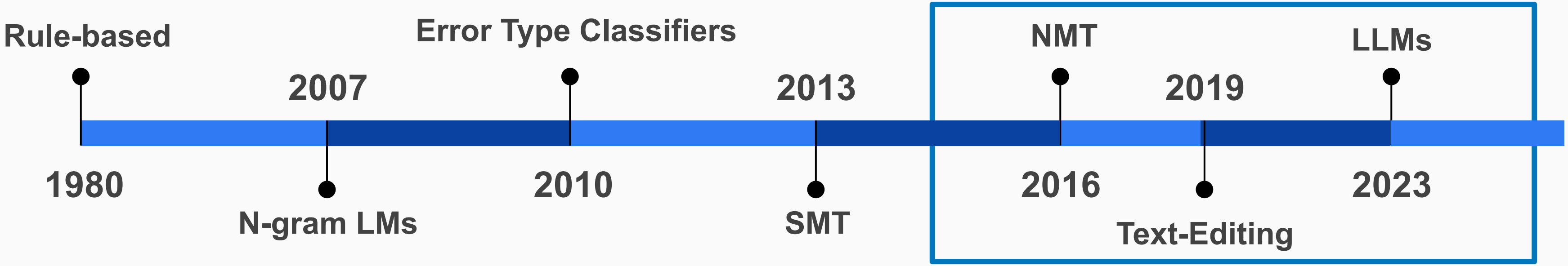
- Align orig and hyp using custom, linguistically-enhanced Damerau-Levenshtein (POS, lemma, chars)
- Use rules to automatically classify hyp edits
- Use TP, FP, FN to compute overall and error type F-scores

Official scorer of the BEA-2019 shared task.
Can also be used to standardize corpus annotation.

Roadmap

- Introduction
- Data and Evaluation
- **Approaches**
- LLMs for Grammatical Error Correction
- Challenges and Future Work

GEC Modeling Approaches



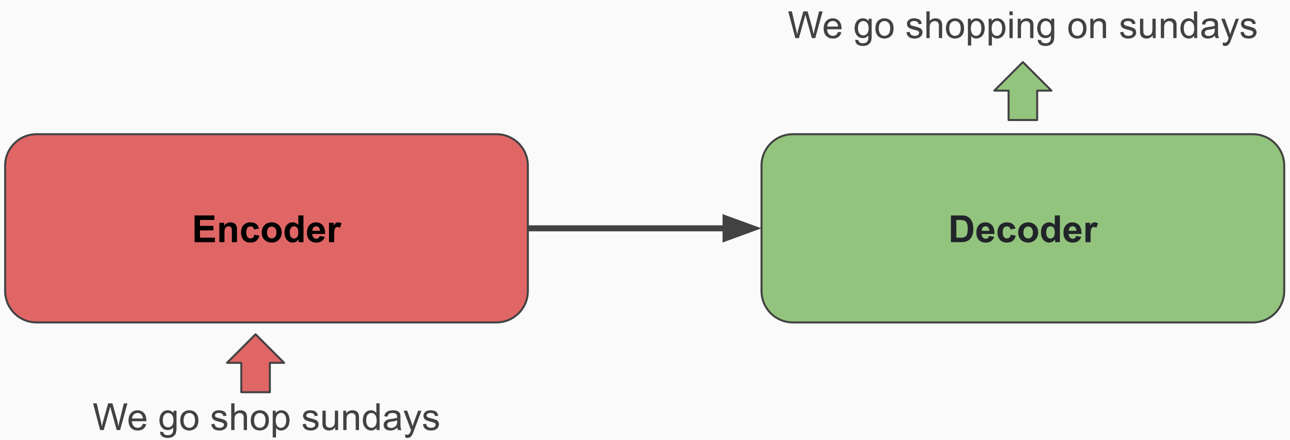
GEC Modeling Approaches: NMT

- GEC as neural machine translation (NMT)

“Incorrect” Text → “Correct” Text

- A large number of well-established methods from NMT have been applied to and adapted for GEC.

GEC Modeling Approaches: NMT



- Training on parallel sentence pairs using a gradient-based optimizer and cross-entropy loss; decoding with beam search

- Recurrent Neural Networks (RNN) (Bahdanau et al., 2015; Miceli Barone et al., 2017), Convolutional Neural Networks (CNN) (Gehring et al., 2017), Transformer (Vaswani et al., 2017).

GEC Modeling Approaches: NMT

GEC as low-resource NMT:

- A Multilayer Convolutional Encoder-Decoder Neural Network for Grammatical Error Correction (Chollampatt & Ng, 2018)
- Approaching Neural Grammatical Error Correction as a Low-Resource Machine Translation Task (Junczys-Dowmunt et al., 2018)
- Grammatical Error Correction in Low-Resource Scenarios (Náplava & Straka, 2019)
- Corpora Generation for Grammatical Error Correction (Lichtarge et al., 2019)

GEC Modeling Approaches: NMT

GEC as low-resource NMT:

- Subword Segmentation (e.g., BPE)

- Domain Adaptation:
  - Oversampling the in-domain data
  - Error rate adaptation

- Regularization:
  - Dropout over source words as a noising strategy

GEC Modeling Approaches: NMT

GEC as low-resource NMT:

- Model Ensembles:
    - Ensemble of independently trained models
    - Combining with a language model
    - Single models: averaging model checkpoints or exponential smoothing of model parameters

- Artificial Error Generation:
    - Random perturbations to clean monolingual texts (unsupervised).
    - Error generation based on the error distributions of annotated corpora.
    - Using other parallel corpora, e.g. Wikipedia revisions, machine translation corpora

GEC Modeling Approaches: Text-Editing

Observations:

- GEC is a monolingual task
  - Source and target often overlap
  - Generating the target from scratch is wasteful

- Can reconstruct the target from the source via basic ops like **KEEP**, **DELETE**, **INSERT, REPLACE**

GEC Modeling Approaches: Text-Editing

Observations:

- GEC is a monolingual task
  - Source and target often overlap
  - Generating the target from scratch is wasteful

- Can reconstruct the target from the source via basic ops like **KEEP**, **DELETE**, **INSERT, REPLACE**

After many years he still **dream** **to** **become** a superhero

GEC Modeling Approaches: Text-Editing

Observations:

- GEC is a monolingual task
  - Source and target often overlap
  - Generating the target from scratch is wasteful

- Can reconstruct the target from the source via basic ops like **KEEP**, **DELETE**, **INSERT, REPLACE**

After	many	years		he	still	**dream**	**to**	**become**	a	superhero
Keep	Keep	Keep	**Insert ,**	Keep	Keep	**Replace (dreams)**	**Replace (of)**	**Replace (becoming)**	Keep	Keep

GEC Modeling Approaches: Text-Editing

Observations:

- GEC is a monolingual task
  - Source and target often overlap
  - Generating the target from scratch is wasteful

- Can reconstruct the target from the source via basic ops like **KEEP**, **DELETE**, **INSERT, REPLACE**

After	many	years		he	still	**dream**	**to**	**become**	a	superhero
Keep	Keep	Keep	**Insert ,**	Keep	Keep	**Replace (dreams)**	**Replace (of)**	**Replace (becoming)**	Keep	Keep
After	many	years	**,**	he	still	**dreams**	**of**	**becoming**	a	superhero

GEC Modeling Approaches: Text-Editing

- Text-editing models generate natural language by applying edit operations to the input text to produce the target text

- Key benefits:
  - **Data Efficiency:** text editing models need less training data
  - **Latency:** can be >10x faster inference
  - **Control:** control over what the model generates

GEC Modeling Approaches: Text-Editing

- **Key Ingredients:**
  - **Convert** training target texts into **target tag** sequences (i.e., Edit Operations)
    - <u>KEEP</u>: Keeps the current token
    - <u>DELETE</u>: Deletes the current token
    - <u>REPLACE</u>:
      - REPLACE_X: Replace with a specific token/phrase X (e.g. LaserTagger, GECToR)
      - REPLACE: Replace with a placeholder and use a separate insertion component to fill the blank (e.g., Felix)
    - <u>APPEND</u>/<u>PREPEND</u>: Inserts new token(s) next to the current token

  - **Tagging Model:**
    - PLM Encoder: BERT, XLNET, etc.

GEC Modeling Approaches: Text-Editing

AutoRegressive Tagging	Non-AutoRegressive Tagging
Seq2Edits (Stahlberg and Kumar, 2020)	LaserTagger (Malmi et al., 2019)
	PIE (Awasthi et al., 2019)
	LevT (Gu et al., 2019)
	Felix (Mallinson et al., 2020)
	Masker (Malmi et al., 2020)
	GECToR (Omelianchuk et al., 2020)

Roadmap

- Introduction
- Data and Evaluation
- Approaches
- **LLMs for Grammatical Error Correction**
- Challenges and Future Work

LLMs for GEC

Observations:

- Supervised methods are data hungry
  - Collecting large-scale in-domain data is challenging
  - SOTA models rely on synthetic data for pretraining

- Prompt-based LLMs excel in various tasks, what about GEC?

LLMs for GEC

Observations:

- Supervised methods are data hungry
  - Collecting large-scale in-domain data is challenging
  - SOTA models rely on synthetic data for pretraining

- Prompt-based LLMs excel in various tasks, what about GEC?

Refine with proper English:
He **enjoying to play** the guitar. → **LLM** → He **enjoys playing** the guitar.

LLMs for GEC

- Is ChatGPT a Highly Fluent Grammatical Error Correction System? A Comprehensive Evaluation (Fang et al., 2023)
- Analyzing the Performance of GPT-3.5 and GPT-4 in Grammatical Error Correction (Coyne et al., 2023)
- ChatGPT or Grammarly? Evaluating ChatGPT on Grammatical Error Correction Benchmark (Wu et al., 2023)
- Exploring Effectiveness of GPT-3 in Grammatical Error Correction: A Study on Performance and Controllability in Prompt-Based Methods (Loem et al., 2023)
- GPT-3.5 for Grammatical Error Correction (Katinskaia et al., 2024)
- Prompting open-source and commercial language models for grammatical error correction of English learner text (Davis et al., 2024)

LLMs for GEC: Prompting Strategies

- **References:** Fang et al., 2023, Coyne et al., 2023, Wu et al., 2023, Loem et al., 2023, Katinskaia et al., 2024

- Improve the overall grammaticality and fluency of the input text

- Make minimal edits and stay as close as possible to the input text

LLMs for GEC: Prompting Strategies

- **Zero-shot:**

  - Reply with a corrected version of the input sentence with all grammatical and spelling errors fixed. If there are no errors, reply with a copy of the original sentence.

    Input sentence: {x}
    Corrected sentence:

  - Provide a grammatical correction for the following sentence indicated by <input> ERROR </input> tag, making only necessary changes. If the input text is already correct, return it unchanged. Output the corrected version directly without any comments and explanations. Remember to format your corrected output with the tag <output> Your Corrected Version </output>. Please start: <input> ERROR </input>

  - Do grammatical error correction on all the following sentences I type in the conversation.

LLMs for GEC: Prompting Strategies

- **Zero-shot chain-of-thought (CoT):**

- Please identify and correct any grammatical errors in the following sentence indicated by <input> ERROR </input>, you need to comprehend the sentence as a whole before identifying and correcting any errors step by step. Afterward, output the corrected version directly without any explanations. Remember to format your corrected output results with the tag <output> Your Corrected Version </output>. Please start: <input> ERROR </input>

LLMs for GEC: Prompting Strategies

- **Few-shot chain-of-thought (CoT):**

  - `Please identify and correct any grammatical errors in the following sentence indicated by <input> ERROR </input>, you need to comprehend the sentence as a whole before identifying and correcting any errors step by step. Afterward, output the corrected version directly without any explanations. Here are some incontext examples:`
    `(1)<input> SRC-1 </input>: <output> TGT-1 </output>;`
    `(2)<input> SRC-2 </input>: <output> TGT-2 </output>;`
    `(3)<input> SRC-3 </input>: <output> TGT-3 </output>.`
    `Please feel free to refer to these examples.`
    `Remember to format your corrected outputs results with the tag <output> Your Corrected Version </output>. Please start: <input> ERROR </input>`

LLMs for GEC: Prompting Strategies

- **Few-shot chain-of-thought (CoT) – Minimal Edits:**

  - ```
    You are an English language teacher. A student has sent you the
    following text.
    {text}
    Provide a grammatical correction for the text, making only necessary
    changes. Do not provide any additional comments or explanations. If
    the input text is already correct, return it unchanged.
    ```
    ```

LLMs for GEC: Results

	CoNLL-14 (Test)			BEA-19 (Test)			JFLEG (Test)
	P	R	$F_{0.5}$	P	R	$F_{0.5}$	GLEU
Transformer	60.1	36.6	53.3	60.9	48.3	57.9	55.4
TagGEC (Stahlberg and Kumar, 2021)	72.8	49.5	66.6	72.1	64.4	70.4	64.7
GECToR (Omelianchuk et al., 2020)	75.6	44.5	66.3	76.7	57.8	71.9	58.6
T5-xxl (Rothe et al., 2021)	-	-	**68.9**	-	-	**75.9**	-
GPT-3.5 Turbo (0-shot + CoT) (Fang et al., 2023)	50.2	59.0	51.7	32.1	70.5	36.1	61.4
GPT-3.5 Turbo (3-shot + CoT) (Fang et al., 2023)	51.3	**62.4**	53.2	34.0	**70.2**	37.9	63.5
GPT-3.5 text-davinci-003 (16-shot) (Loem et al., 2023)	-	-	-	-	-	57.4	**67.0**
GPT-4 (2-shot) (Coyne et al., 2023)	-	-	-	-	-	52.8	65.0
GPT-4 (0-shot) (Omelianchuk et al., 2024)	59.0	55.4	58.2	-	-	-	-
Chat-LLaMa-2-13B (0-shot) (Omelianchuk et al., 2024)	49.1	56.1	50.4	-	-	-	-
Chat-LLaMa-2-13B + FT (Omelianchuk et al., 2024)	**77.3**	45.6	67.9	74.6	67.8	73.1	-

LLMs for GEC: Results

	CoNLL-14 (Test)			BEA-19 (Test)			JFLEG (Test)
	P	R	$F_{0.5}$	P	R	$F_{0.5}$	GLEU
Transformer	60.1	36.6	53.3	60.9	48.3	57.9	55.4
TagGEC (Stahlberg and Kumar, 2021)	72.8	49.5	66.6	72.1	64.4	70.4	64.7
GECToR (Omelianchuk et al., 2020)	75.6	44.5	66.3	76.7	57.8	71.9	58.6
T5-xxl (Rothe et al., 2021)	-	-	**68.9**	-	-	**75.9**	-
GPT-3.5 Turbo (0-shot + CoT) (Fang et al., 2023)	50.2	59.0	51.7	32.1	70.5	36.1	61.4
GPT-3.5 Turbo (3-shot + CoT) (Fang et al., 2023)	51.3	**62.4**	53.2	34.0	**70.2**	37.9	63.5
GPT-3.5 text-davinci-003 (16-shot) (Loem et al., 2023)	-	-	-	-	-	57.4	**67.0**
GPT-4 (2-shot) (Coyne et al., 2023)	-	-	-	-	-	52.8	65.0
GPT-4 (0-shot) (Omelianchuk et al., 2024)	59.0	55.4	58.2	-	-	-	-
Chat-LLaMa-2-13B (0-shot) (Omelianchuk et al., 2024)	49.1	56.1	50.4	-	-	-	-
Chat-LLaMa-2-13B + FT (Omelianchuk et al., 2024)	**77.3**	45.6	67.9	74.6	67.8	73.1	-

LLMs for GEC: Results

	CoNLL-14 (Test)			BEA-19 (Test)			JFLEG (Test)
	P	R	$F_{0.5}$	P	R	$F_{0.5}$	GLEU
Transformer	60.1	36.6	53.3	60.9	48.3	57.9	55.4
TagGEC (Stahlberg and Kumar, 2021)	72.8	49.5	66.6	72.1	64.4	70.4	64.7
GECToR (Omelianchuk et al., 2020)	75.6	44.5	66.3	76.7	57.8	71.9	58.6
T5-xxl (Rothe et al., 2021)	-	-	**68.9**	-	-	**75.9**	-
GPT-3.5 Turbo (0-shot + CoT) (Fang et al., 2023)	50.2	59.0	51.7	32.1	70.5	36.1	61.4
GPT-3.5 Turbo (3-shot + CoT) (Fang et al., 2023)	51.3	**62.4**	53.2	34.0	**70.2**	37.9	63.5
GPT-3.5 text-davinci-003 (16-shot) (Loem et al., 2023)	-	-	-	-	-	57.4	**67.0**
GPT-4 (2-shot) (Coyne et al., 2023)	-	-	-	-	-	52.8	65.0
GPT-4 (0-shot) (Omelianchuk et al., 2024)	59.0	55.4	58.2	-	-	-	-
Chat-LLaMa-2-13B (0-shot) (Omelianchuk et al., 2024)	49.1	56.1	50.4	-	-	-	-
Chat-LLaMa-2-13B + FT (Omelianchuk et al., 2024)	**77.3**	45.6	67.9	74.6	67.8	73.1	-

LLMs for GEC: Results

	CoNLL-14 (Test)			BEA-19 (Test)			JFLEG (Test)
	P	R	$F_{0.5}$	P	R	$F_{0.5}$	GLEU
Transformer	60.1	36.6	53.3	60.9	48.3	57.9	55.4
TagGEC (Stahlberg and Kumar, 2021)	72.8	49.5	66.6	72.1	64.4	70.4	64.7
GECToR (Omelianchuk et al., 2020)	75.6	44.5	66.3	76.7	57.8	71.9	58.6
T5-xxl (Rothe et al., 2021)	-	-	**68.9**	-	-	**75.9**	-
GPT-3.5 Turbo (0-shot + CoT) (Fang et al., 2023)	50.2	59.0	51.7	32.1	70.5	36.1	61.4
GPT-3.5 Turbo (3-shot + CoT) (Fang et al., 2023)	51.3	**62.4**	53.2	34.0	**70.2**	37.9	63.5
GPT-3.5 text-davinci-003 (16-shot) (Loem et al., 2023)	-	-	-	-	-	57.4	**67.0**
GPT-4 (2-shot) (Coyne et al., 2023)	-	-	-	-	-	52.8	65.0
GPT-4 (0-shot) (Omelianchuk et al., 2024)	59.0	55.4	58.2	-	-	-	-
Chat-LLaMa-2-13B (0-shot) (Omelianchuk et al., 2024)	49.1	56.1	50.4	-	-	-	-
Chat-LLaMa-2-13B + FT (Omelianchuk et al., 2024)	**77.3**	45.6	67.9	74.6	67.8	73.1	-

LLMs for GEC: Results (Human Evaluation)

- **References:** Fang et al., 2023, Wu et al., 2023, Coyne et al., 2023
- Human evaluation of GPT* outputs against gold references on small samples (~100 sentences)
- GPT* outputs preferred by human raters for higher fluency
- LLMs identified and corrected errors missed by human annotators (i.e., under-corrections in gold references)

LLMs for GEC: Recommendations

- Fluency vs. Minimal Edits Prompts:
  - Performance varies based on the dataset:
    - JFLEG → Fluency prompts
    - CoNLL-14 and BEA-19 → Minimal edits prompts

- Few-Shot Prompting Outperforms Zero-Shot:
  - Performance improves as the number of examples increases (e.g., 1-shot < 3-shot < 5-shot)

- Chain-of-Thought (CoT) doesn't always lead to improvements

LLMs for GEC: Takeaways

- Recent LLM-powered methods do not outperform other available approaches to date (e.g., text-editing, seq2seq)

- However, being properly set, they can perform on par with other methods and lead to more powerful ensembles (Omelianchuk et al., 2024)

- The 10–50x increase in model size leads to rather small improvements

- LLM outputs preferred by human raters for higher fluency, but:
  - Minimal corrections are prioritized in educational applications to guide learners on how to amend errors effectively

  - GEC guidelines emphasize minimal edits to help learners express what they’re trying to say (Nicholls, 2003)

LLMs for GEC: Grammatical Error Explanation

Original: After many years he still **dream to become** a superhero
Corrected: After many years he still **dreams of becoming** a superhero

LLMs for GEC: Grammatical Error Explanation

Original: After many years he still **dream to become** a superhero
Corrected: After many years he still **dreams of becoming** a superhero

Error type: subject-verb agreement

Error type: verb-preposition usage

LLMs for GEC: Grammatical Error Explanation

Original: After many years he still **dream to become** a superhero
Corrected: After many years he still **dreams of becoming** a superhero

↓

Error type: subject-verb agreement
Error explanation: The verb “dream” is corrected to “dreams” to agree with the third-person singular subject “he” in the present tense.

Error type: verb-preposition usage
Error explanation: The phrase “to become” is replaced with “of becoming“ because “dream of“ is the correct collocation in English, and the gerund “becoming“ is required after the preposition “of“.

LLMs for GEC: Grammatical Error Explanation

- **Recent Datasets:**
  - **XGEC** (Kaneko et al., 2024):
    - 888 manually annotated samples from GEC datasets (NUCLE, CoNLL2013, and CoNLL2014) with edit explanations
  - **GMEG-EXP** (López Cortez et al., 2024)
    - 6K sentences from GMEG
    - Explanations generated either by human experts or GPT3.5
- **Other Datasets:**
  - **EXPECT** (Fei et al., 2023)
  - **ICNALE + Explanations** (Nagata et al., 2019)

LLMs for GEC: Grammatical Error Explanation

- **LLMs Performance on GEE:**
  - GEE! Grammar Error Explanation with Large Language Models (Song et al., 2024):
  - Controlled Generation with Prompt Insertion for Natural Language Explanations in Grammatical Error Correction (Kaneko et al., 2024)
    - Evaluation of error detection and explanations quality
    - GPT-4 (one-shot prompting) → struggles to identify and explain errors. Detects onlys 60% errors and correctly explains 68% of the errors it detects (via human-evaluation)
    - Providing extracted edits in the prompt leads to better explanations

Roadmap

- Introduction
- Data and Evaluation
- Approaches
- LLMs for Grammatical Error Correction
- **Challenges and Future Work**

Challenges and Future Work

- Evaluation
  - Robust evaluation of GEC system output is still an unsolved problem
  - Learning benefits of minimal edit feedback vs. fluency rewrites
- Personalized Systems
  - System performance is also to the profiles of the users in the training data
- Feedback Comment Generation (i.e., Explainable GEC)
  - Crucial in educational contexts
- Multilingual and Spoken GEC
  - More research is needed
- Synthetic Data Generation using LLMs
  - More research is needed