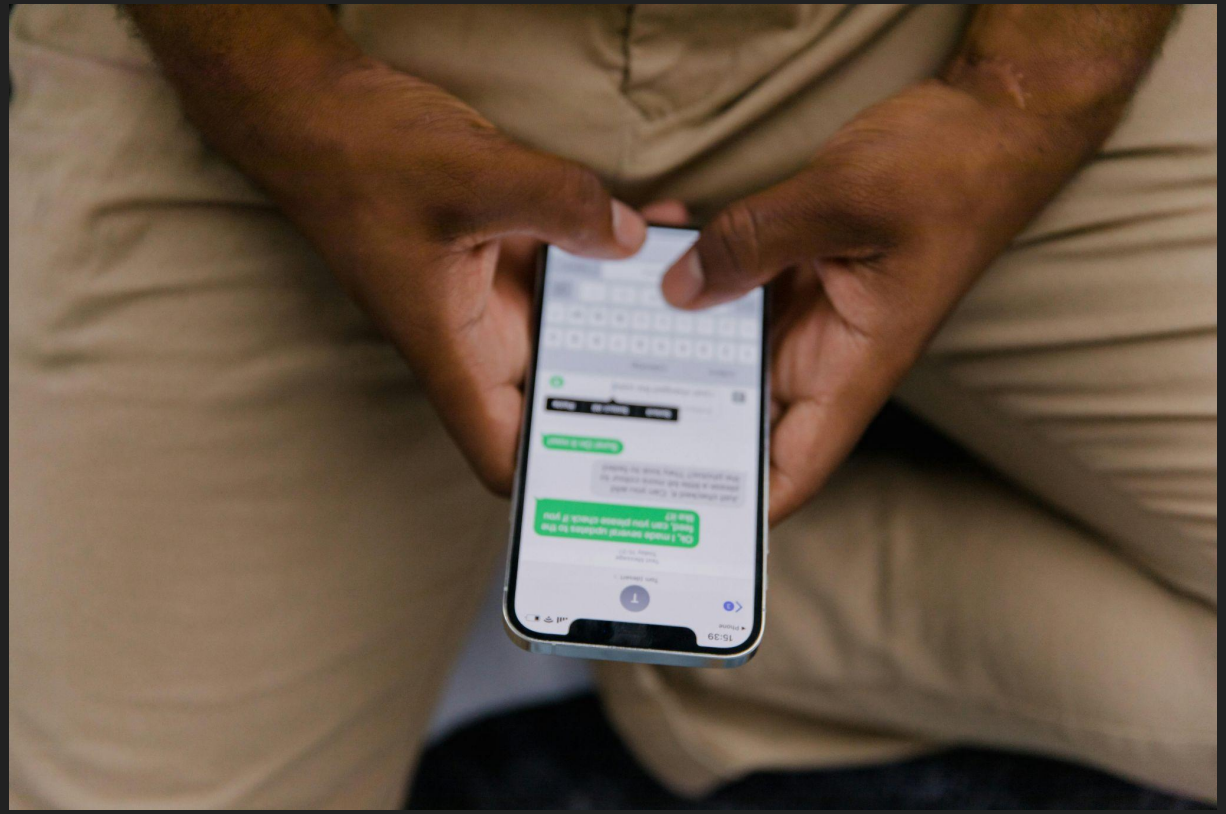
SkipWriter:

LLM-Powered Abbreviated Writing on Tablets

Zheer Xu[1], Shanqing Cai[2], Mukund Varma T[3], Subhashini Venugopalan[2], Shumin Zhai[2]

[1] DARTMOUTH [2] Google [3] UC San Diego



Conventional Predictive Interfaces

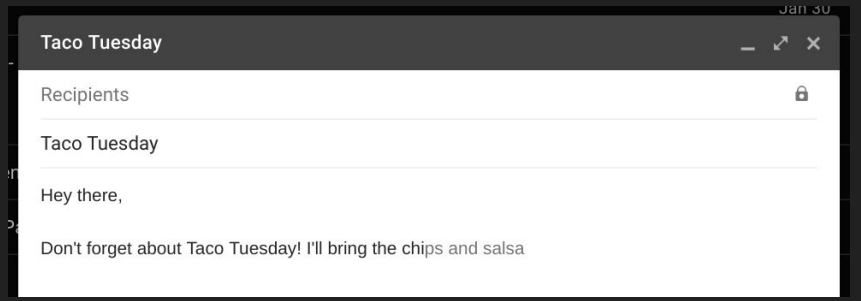
Word Forward Suggestion

N-gram LM

Conventional Predictive Interfaces

Word Forward Suggestion

N-gram LM



Phrase Forward Suggestion

Transformer-based LLM

Source: Chen et al. *Gmail Smart Compose: Real-Time Assisted Writing*. KDD 2019.

Phrase Abbreviation

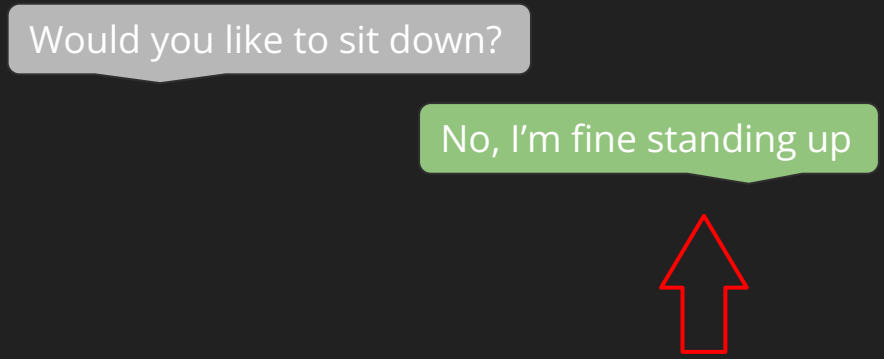
As far as I know	AFAIK
Sounds good to me	SGTM
See you later	CUL8R
Never mind	NVM
What are you doing	?RU doing
Forget it	4get it

Common phrase abbreviations in SMS Language

Phrase Abbreviation

As far as I know	AFAIK
Sounds good to me	SGTM
See you later	CUL8R
Never mind	NVM
What are you doing	?RU doing
Forget it	4get it

Common phrase abbreviations in SMS Language



Using LLM to expand Initial-only phrase abbreviation for dialogues

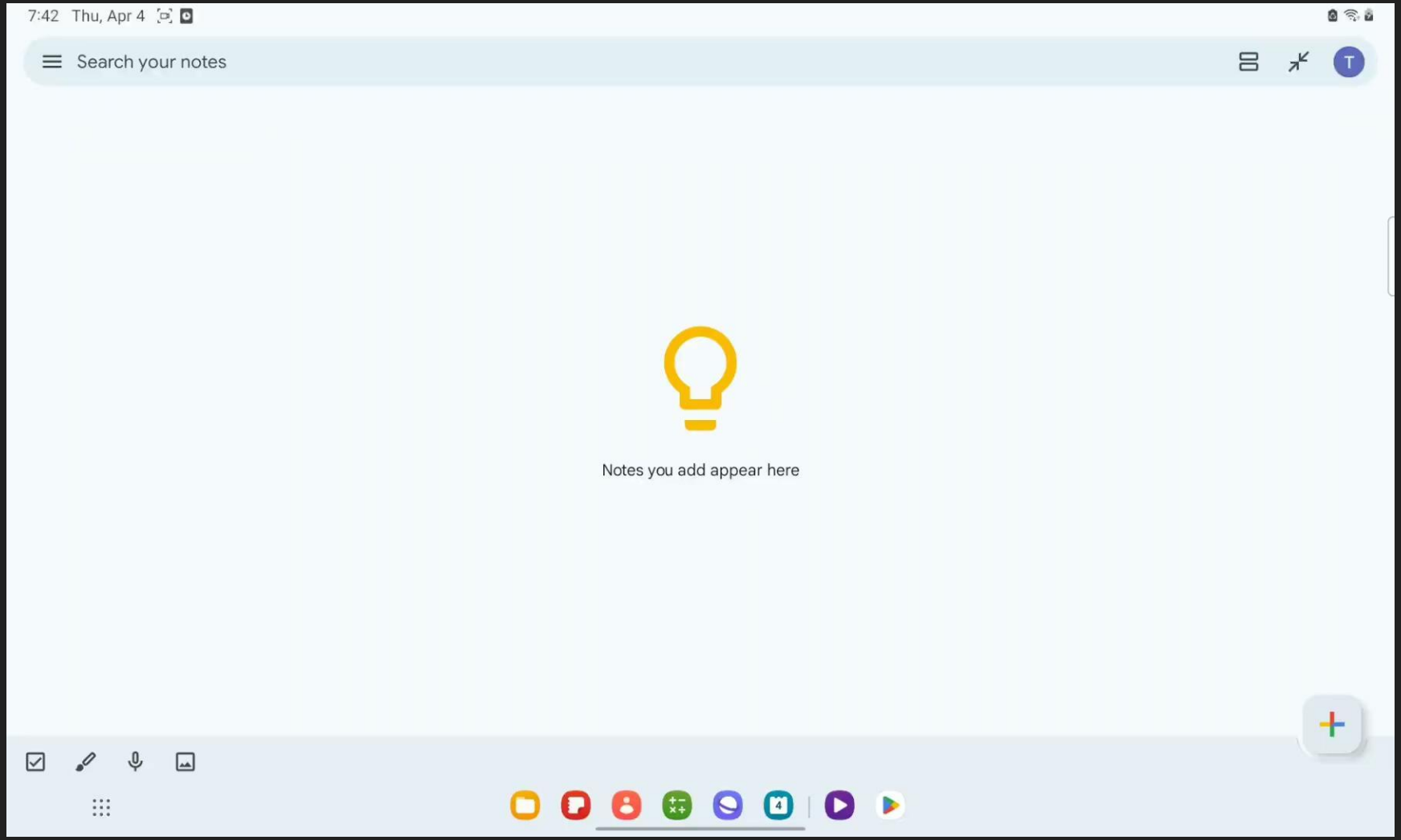
Source: Cai et al. *Context-Aware Abbreviation Expansion Using Large Language Models*. NAACL 2022.

Research Goal

- Generic text input where text can be abbreviated in chunks

- Minimize human input efforts by harnessing a LLM decoder


Input Modality
Abbreviation Form
Decoder Training
Interface Design


7:42 Thu, Apr 4
Search your notes
T
Notes you add appear here

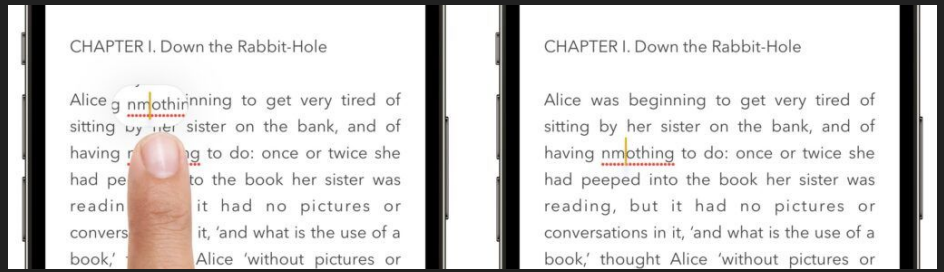
Input Modality: Handwriting

- Handwriting has prolonged stress on the hand and wrist.

Input Modality: Handwriting

- Handwriting has prolonged stress on the hand and wrist.

- Handwriting has the flexibility of pen-based interaction.
  - “Random Access” instead of an insertion cursor



Featured image by iStock.com/Ralf Geithe

Abbreviation Form

Variable-length Prefix-based Abbreviation

Example*: when would you come home*

Tianshi Li, Philip Quinn, and Shumin Zhai. *C-PAK: Correcting and completing variable-length Prefix-based Abbreviated Keystrokes. A*CM Trans. Comput. Hum. Interact. Jul 2022.

Abbreviation Form

- Flexibility in the abbreviation length
  - “would” can be written as: “w”, “wo”, “wou”, “woul”, “would”

Abbreviation Form

- Flexibility in the abbreviation length
  - “would” can be written as: “w”, “wo”, “wou”, “woul”, “would”
- High Ceiling of character savings (i.e., initial-only)
- Guaranteed successful entry (i.e., full format)

Abbreviation Form

- Flexibility in the abbreviation length
  - “would” can be written as: “w”, “wo”, “wou”, “woul”, “would”
- High Ceiling of character savings (i.e., initial-only)
- Guaranteed successful entry (i.e., full format)
- Easy to complete/extend in a progressive way

LLM-Powered Abbreviation Decoder

- In-production handwriting recognizer + LLM for abbreviation decoder (PaLM2)

LLM-Powered Abbreviation Decoder

- In-production handwriting recognizer + LLM for abbreviation decoder (PaLM2)

- Arbitrarily generate abbreviations for fine-tuning the LLM.
  - Full phrase: “have a way of”
  - Generated abbreviation: “ha a wa o”

LLM-Powered Abbreviation Decoder

- In-production handwriting recognizer + LLM for abbreviation decoder (PaLM2)

- Arbitrarily generate abbreviations for fine-tuning the LLM.
  - Full phrase: “have a way of”
  - Generated abbreviation: “ha a wa o”

- Preceding context is also utilized for decoding.

LLM-Powered Abbreviation Decoder

- Handwriting recognizer (ML Kit) + LLM for abbreviation decoder (PaLM2)

- Arbitrarily generate abbreviations for fine-tuning the LLM.
  - Target text: “have a way of”
  - Generated abbreviation: “ha a wa o”

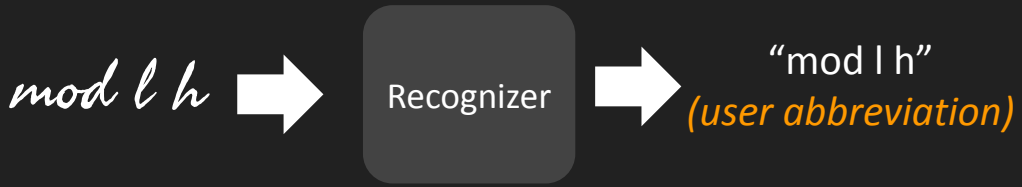
- Data source: 4 public datasets (papers, reviews, news, wiki)

LLM-Powered Abbreviation Decoder

Example Text: “Large Language Models (LLMs) may offer transformative opportunities for text input, especially for physically demanding *modalities like handwriting*”

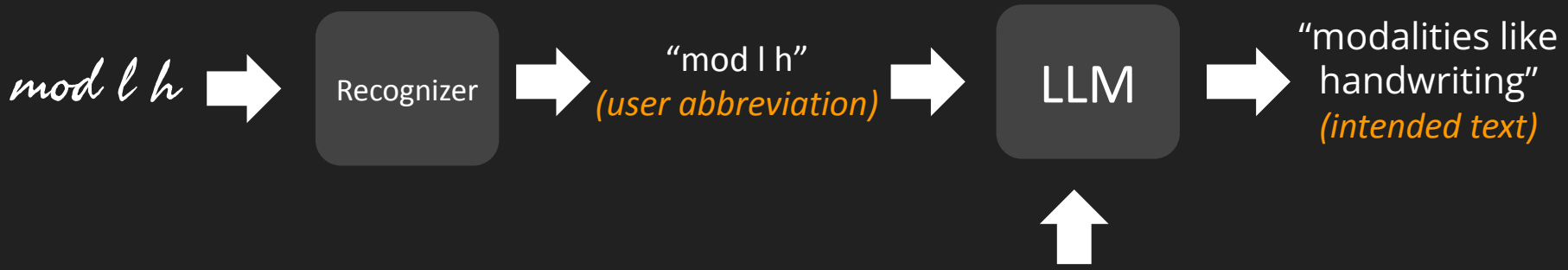
LLM-Powered Abbreviation Decoder

Example Text: “Large Language Models (LLMs) may offer transformative opportunities for text input, especially for physically demanding *modalities like handwriting*”



LLM-Powered Abbreviation Decoder

Example Text: “Large Language Models (LLMs) may offer transformative opportunities for text input, especially for physically demanding *modalities like handwriting*”



Abbreviation Generation

Question: How to generate the prefix abbreviation for each word?

- Random Sampling
  - E.g., “would” -> “w”, “wo”, “wou”, “woul”, “would”
  - Same probability for all options (i.e., uniform distribution).

Abbreviation Generation

Question: How to generate the prefix abbreviation for each word?

- Random Sampling
  - E.g., “would” -> “w”, “wo”, “wou”, “woul”, “would”
  - Same probability for all options (i.e., uniform distribution).
- Problem: each character is not equivalent in terms of disambiguation.
  - For the word “zebra”, “z” or “ze” is probably sufficient for disambiguation.
  - For the word “thank”, “th” is still too vague as it’s a very common prefix in English.

Abbreviation Generation

Question: How to generate the prefix abbreviation for each word?

- Random Sampling
  - E.g., “would” -> “w”, “wo”, “wou”, “woul”, “would”
  - Same probability for all options (i.e., uniform distribution).
- Problem: each character is not equivalent in terms of disambiguation.
  - For the word “zebra”, “z” or “ze” is probably sufficient for disambiguation.
  - For the word “thank”, “th” is still too vague as it’s a very common prefix in English.
- Solution: inversely tie the probability to the frequency of a prefix in English corpus

Abbreviation Generation

Question: How to generate the prefix abbreviation for a word?

- Prefix Entropy

$$H_{\text{prefix}} = -\sum_{w \in W} p(w) \log(p(w))$$

prefix: a character sequence

W: the set of all words start with the sequence *prefix*

Abbreviation Generation

Question: How to generate the prefix abbreviation for a word?

- Progressively determine each character

$$P_i = \frac{H(c_1, c_2, \ldots, c_{i-1})}{H_0}$$

P_i : the probability of the i^{th} character being included

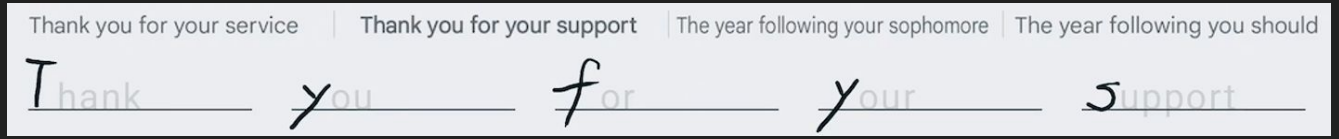
$H(c_1, c_2, \ldots c_{i-1})$: the prefix entropy of the sequence before the i^{th} character

H_0: the entropy of an empty sequence

Interface Design

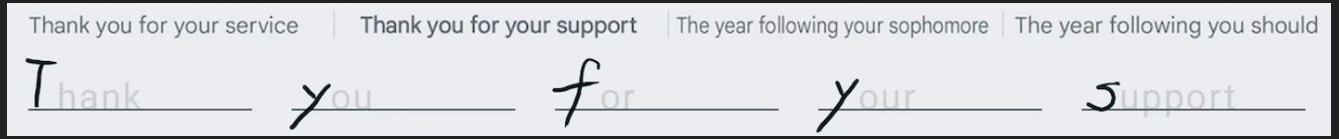
Goal: encourage the user to employ more efficient (i.e., shorter but effective) abbreviations

Interface Design



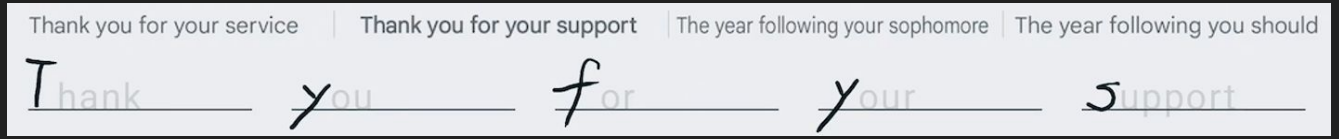
- Easy editing
  - Ambient reminder for users to leave space for future completion

Interface Design



- Easy editing
  - Ambient reminder for users to leave space for future completion
  - Reduces the overhead of completion and encourages users to try aggressive abbreviation

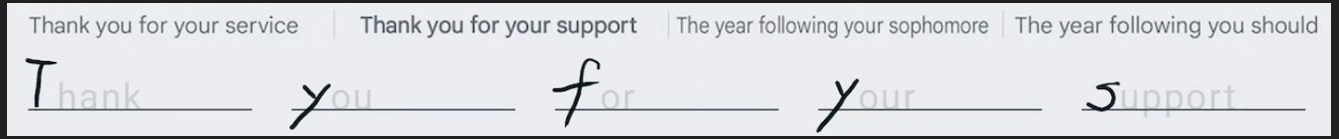
Interface Design: Segmented Rule



- Easy retroactive editing

Intended text: “when would you come home”

Interface Design: Segmented Rule



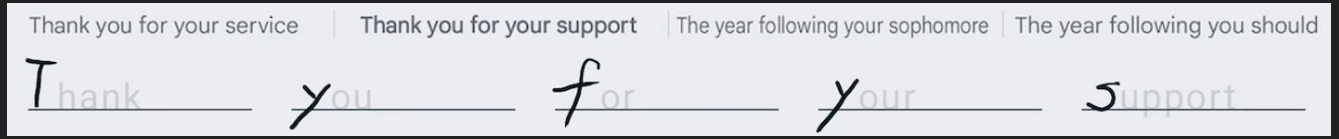
- Easy retroactive editing

Intended text: “when would you come home”

User writes: *w w y c h*

Prediction: When will you come here

Interface Design: Segmented Rule



- Easy retroactive editing

Intended text: “when would you come home”

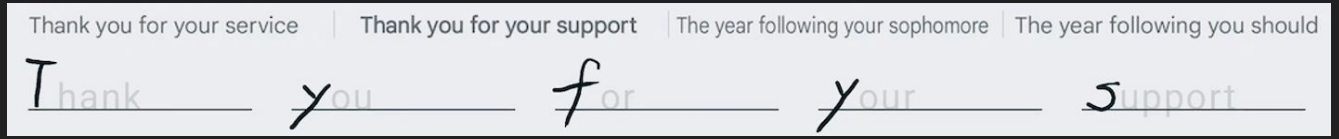
User writes: *w w y c h*

Prediction: When will you come here

User edits: *w wo y c ho*

Prediction: When would you come home

Interface Design: Segmented Rule

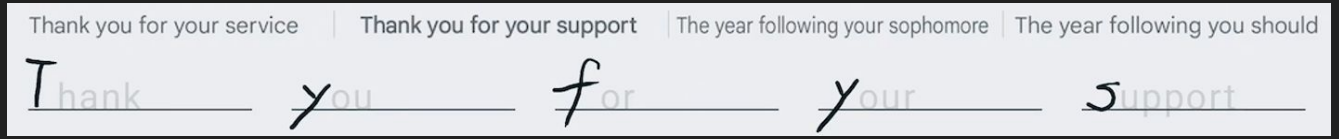


- Easy retroactive editing

Otherwise, the user may “charge ahead” to avoid subsequent edits:

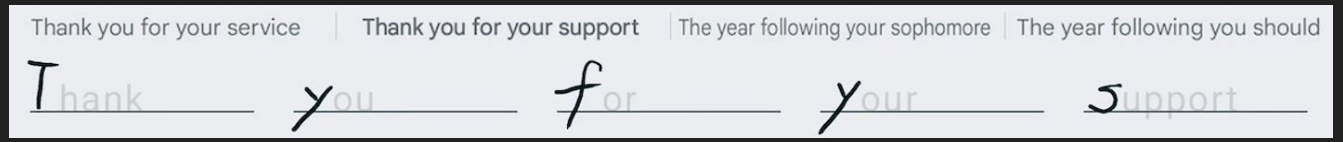
wh wo yo co ho

Interface Design



- Easy editing
- Low-cost word delimiters for robust recognition

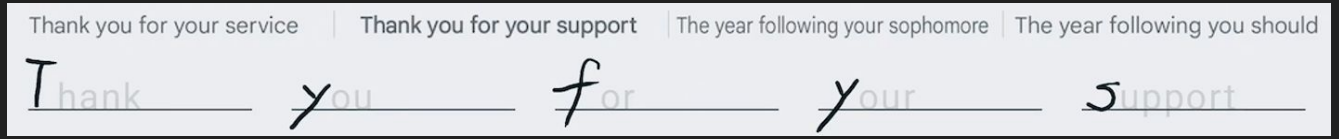
Interface Design: Segmented Rule



- Easy retroactive editing
- Low-cost delimiters for robust recognition

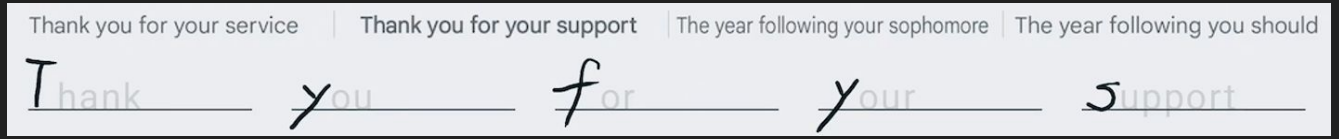
"*w wo y c ho*" can be mis-recognized as "w woye ho"

Interface Design: Segmented Rule

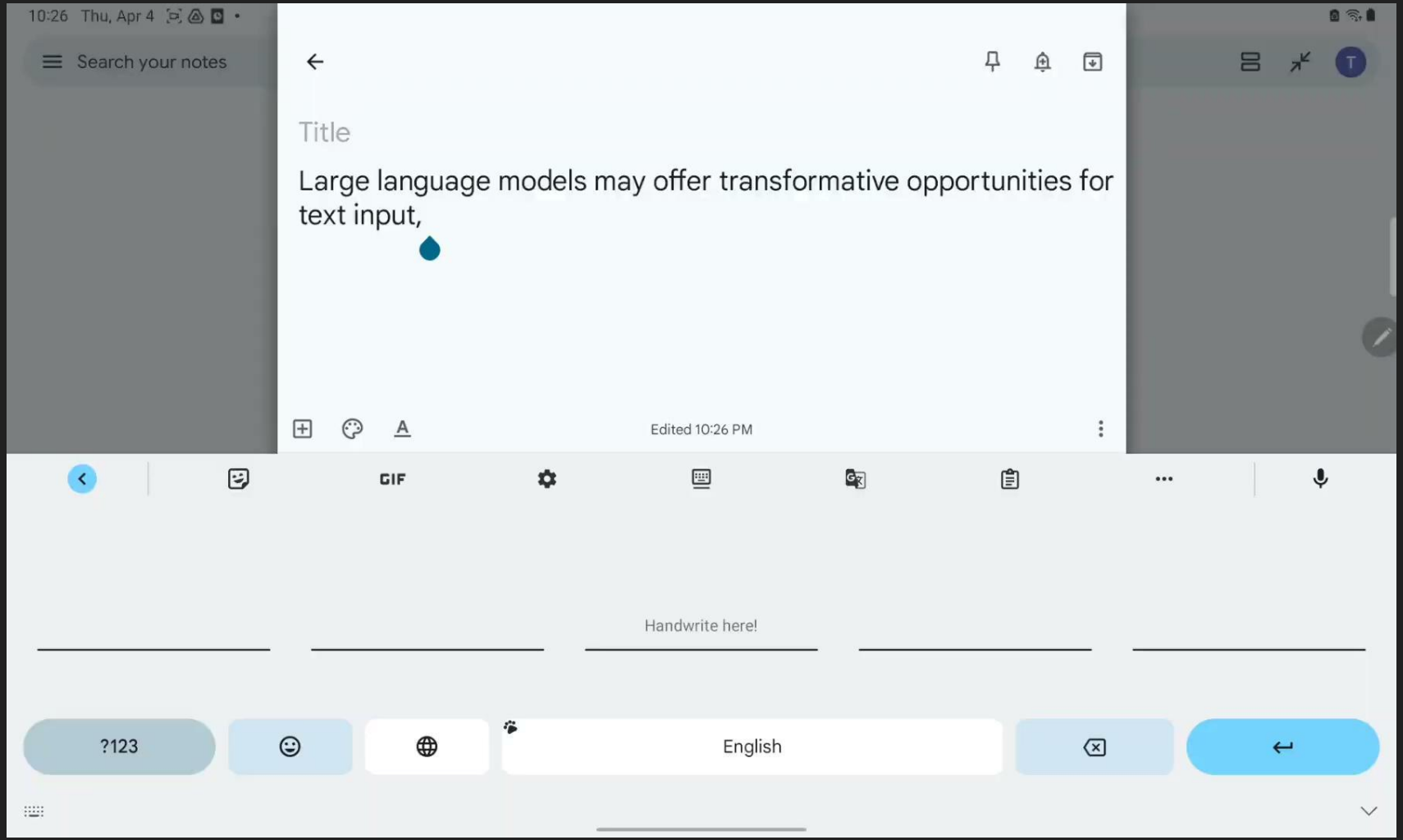


- Easy retroactive editing
- Low-cost delimiters for robust recognition
  - Separate handwriting recognition in each segment

Interface Design



- Easy editing
- Low-cost word delimiters for robust recognition
- Inline visualization of top candidate to minimize attention switch


10:26 Thu, Apr 4
Search your notes
Title
Large language models may offer transformative opportunities for text input,
Edited 10:26 PM
GIF
Handwrite here!
?123
English

User Evaluation

	Abbreviated Handwriting	Conventional Handwriting
Speed (Word Per Minute)	25.78 WPM	24.18 WPM
Word Error Rate (%)	2.08%	4.05%
Traversal Distance per Character *(Metric for physical efforts)*	11.41 mm **(60.19% ↓)**	18.74 mm

Offline Simulation

- Question: How efficient are users’ abbreviations in the study?

Offline Simulation

- Question: How efficient are users' abbreviations in the study?

- Simulate the most aggressive abbreviating behavior
  - Step 1: Start with the initials
  - Step 2: If the target not in the candidates - append one more character to the first wrong word
  - Step 3: Repeat Step 2 until the target appears.
  - Step 4: Get the final abbreviation.

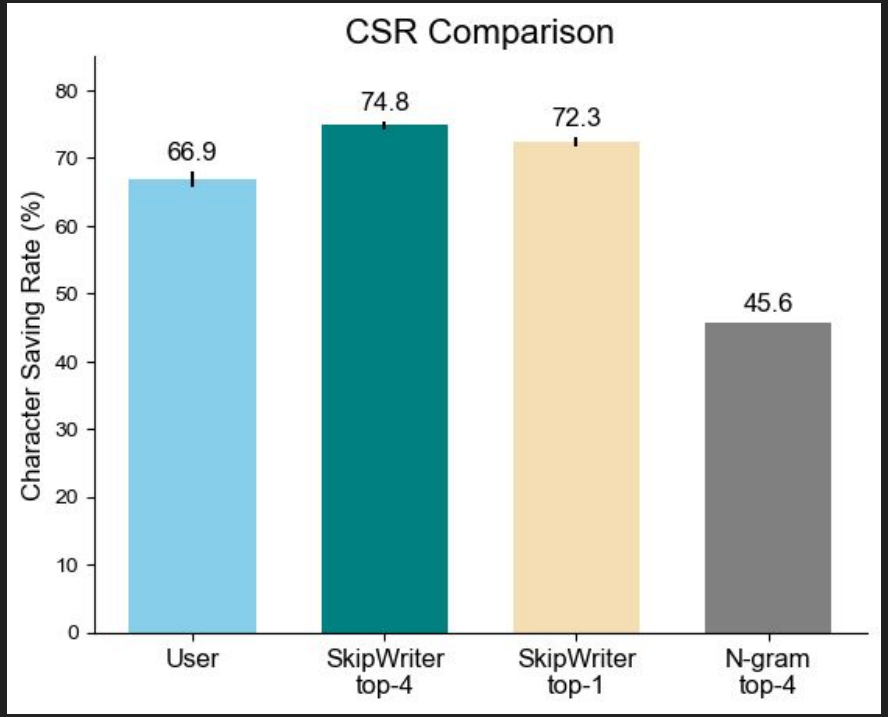
Offline Simulation

- Question: How efficient are abbreviation users adopted in the study?

- Simulate the most aggressive abbreviating behavior
  - Step 1: Start with the initials
  - Step 2: If the target not in the candidates - append one more character to the first wrong word
  - Step 3: Repeat Step 2 until the target appears.
  - Step 4: Get the final abbreviation.

- Compare the User CSR v.s. Simulated CSR
  - Character Saving Rate (CSR): percentage of characters skipped in the abbreviation.

Offline Evaluation



Take-away Messages

- LLM can effectively decode phrase abbreviations and achieve high character savings.

Take-away Messages

- LLM can effectively decode phrase abbreviations and achieve high character savings.

- Users can utilize the LLM’s power to write and abbreviate efficiently on SkipWriter interface with diminished physical efforts.

Thank you!

Discussion on Cognitive Load

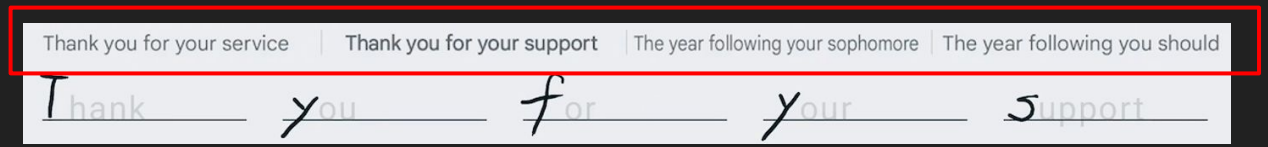
- Observed higher cognitive load in the user study

Discussion on Cognitive Load

- Observed higher cognitive load in the user study
- Common problem in phrase-level input:
  - compare, search, and complete a phrase of multiple words

Discussion on Cognitive Load

- Observed higher cognitive load in the user study
- Common problem in phrase-level input:
  - compare, search, and complete a phrase of multiple words
- Future improvements
  - Decoder: latest foundational models + human data to train an end-to-end decoder
  - Interface: highlight the differences among the candidates



Contributions

AI-Generated

Text Creation

(ChatGPT etc.)

Contributions

AI-Assisted

Text Input

AI-Generated

Text Creation

(ChatGPT etc.)

Future Directions

- On-device reference
- Personalization
- Mitigating cognitive load
- Real word deployment and evaluation

Abbreviations for other languages

Text input is fundamentally a sequence of target acquisition tasks. (i.e., abbreviated input -> partial sequence)

- Adaptation should be based on primary units of that language
  - Word-Based Languages (e.g., English)
  - Syllable-Based Languages (e.g., Chinese)
  - Morphological or Agglutinative Languages (e.g., Turkish, Japanese)
  - Root-Based or Semitic Languages (e.g., Arabic, Hebrew)
  - Polysynthetic Languages (e.g., Inuktitut, Mohawk)