On the Equivalence of Tasks in (Sentential) Semantics

Ning Shi, Grzegorz Kondrak

Alberta Machine Intelligence Institute (Amii)
Department of Computing Science
University of Alberta, Edmonton, Canada

Outline

This presentation will cover our recent publications on computational semantics, especially focusing on the semantic/task relationships.

Motivation: Lexical Gap <-> Machine Translation (ACL 2024)

Word Level: Lexical Substitution <-> Causal Language Modeling (*SEM 2024)

Sentence Level: Paraphrase Identification <-> Textual Inference (*SEM 2024)

Cross Level: Reasoning <-> Language Modeling (?)

Translation-based Lexicalization Generation and Lexical Gap Detection: Application to Kinship Terms

Senyu Li, Bradley Hauer, **Ning Shi**, Grzegorz Kondrak

Alberta Machine Intelligence Institute (Amii)
Department of Computing Science
University of Alberta, Edmonton, Canada

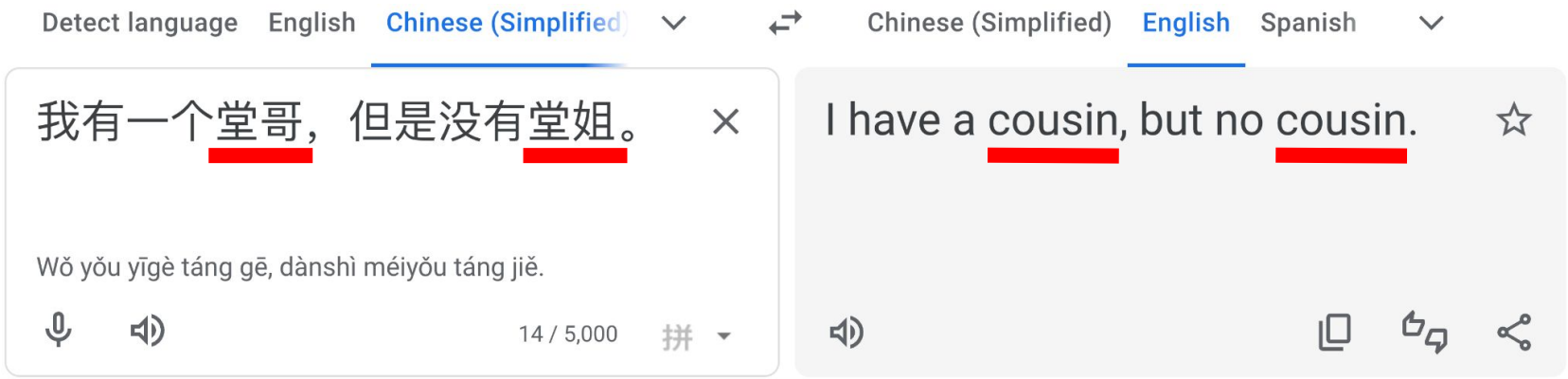
In Proceedings of ACL 2024

An Error Case: Google Translate

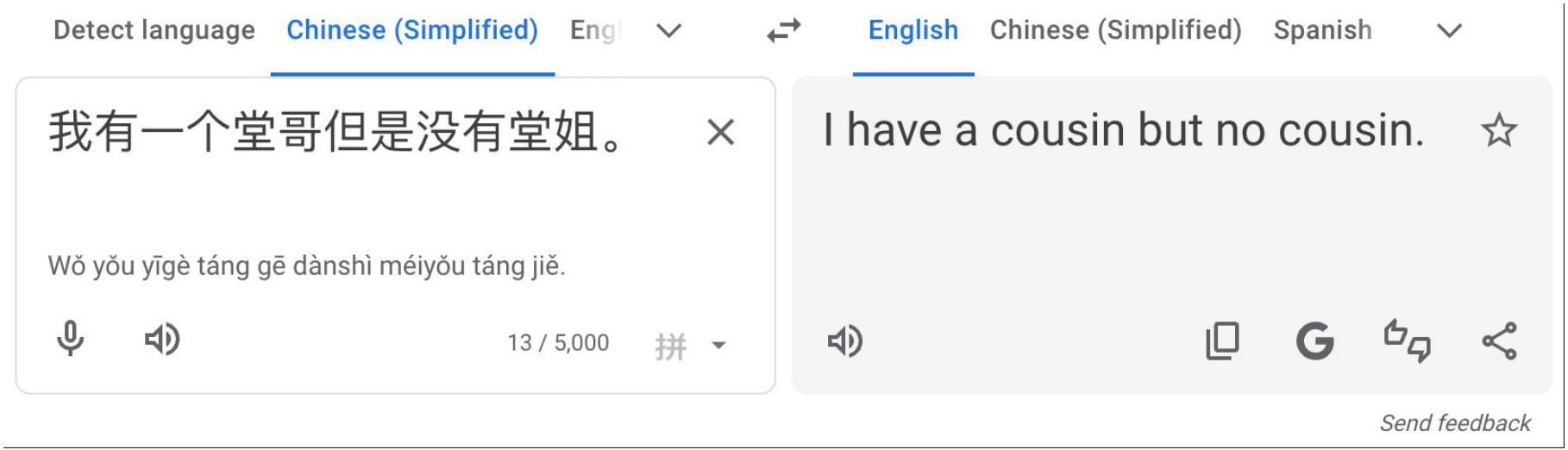
堂哥 “elder son of father’s brother” => "**cousin**"

堂姐 “elder daughter of father’s brother” => "**cousin**"

Other powerful translators make similar errors. (DeepL, Baidu, etc.)



An Error Case: Google Translate



Sample Output of ChatGPT

You

Given a word that means [father's younger brother] in Chinese is [叔叔], and a word that means [mother's brother] in Chinese is [舅舅]. Is there a word that means [elder brother] in [English]? If yes, give me that word. If no, say no.

ChatGPT

Yes, the word in English that means "elder brother" is "brother."

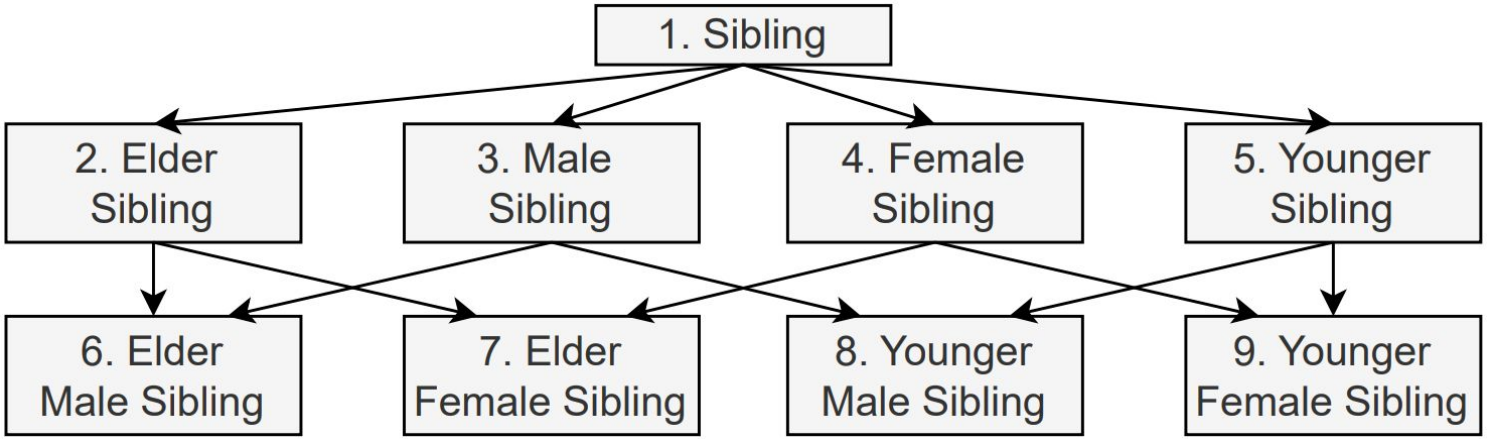
GPT 3.5, Feb. 15, 2024

Concepts

Concept: discrete word meaning

Kinship concepts have clear definitions and hierarchical structure

With well-studied, good gold-standard dataset (Khishigsuren et al, 2022)



Lexicalizations and Lexical Gaps

Lexicalization: a single word which can express (i.e. lexicalize) a concept.

Lexical Gap: a concept that has no lexicalization in a given language.

Concepts	En	Es	Fr	Ja	Fa	Zh	Pl
1	Sibling	ø	fratrie	ø	ø	同胞	ø
2	ø	ø	ø	ø	ø	ø	ø
3	Brother	hermano	frère	ø	برادر	兄弟	brat
4	Sister	hermana	sœur	ø	خواهر	姐妹	siostra
5	ø	ø	ø	ø	ø	ø	ø
6	ø	ø	ø	兄さん	ø	哥哥	ø
7	ø	ø	ø	姉ちゃん	ø	姐姐	ø
8	ø	tato	ø	おとうと	ø	弟弟	ø
9	ø	ø	ø	いもうと	ø	妹妹	ø

Data from *Using Linguistic Typology to Enrich Multilingual Lexicons: the Case of Lexical Gaps in Kinship* (Khishigsuren et al, 2022)

Task definition: LexGen and LexGap

LexGen: Lexicalization Generation

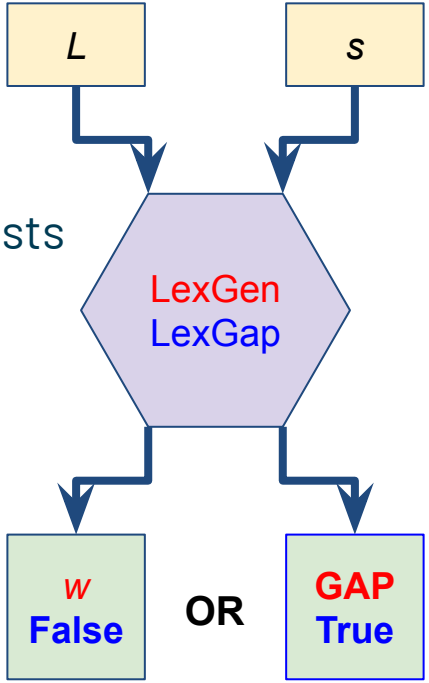
- Input: language *L*, concept *s*
- Output: word *w* in *L* s.t. *w* lexicalizes *s*,

  OR a special token **GAP** indicating that no such *w* exists

LexGap: Lexical Gap Detection

- Input: language *L*, concept *s*
- Output: True if no word in *L* lexicalizes *s*, False otherwise.

LexGen(*L*,*s*) = **GAP** if and only if LexGap(*L*,*s*) = True



Our Work

Problem: How to identify concept lexicalizations and lexical gaps efficiently?

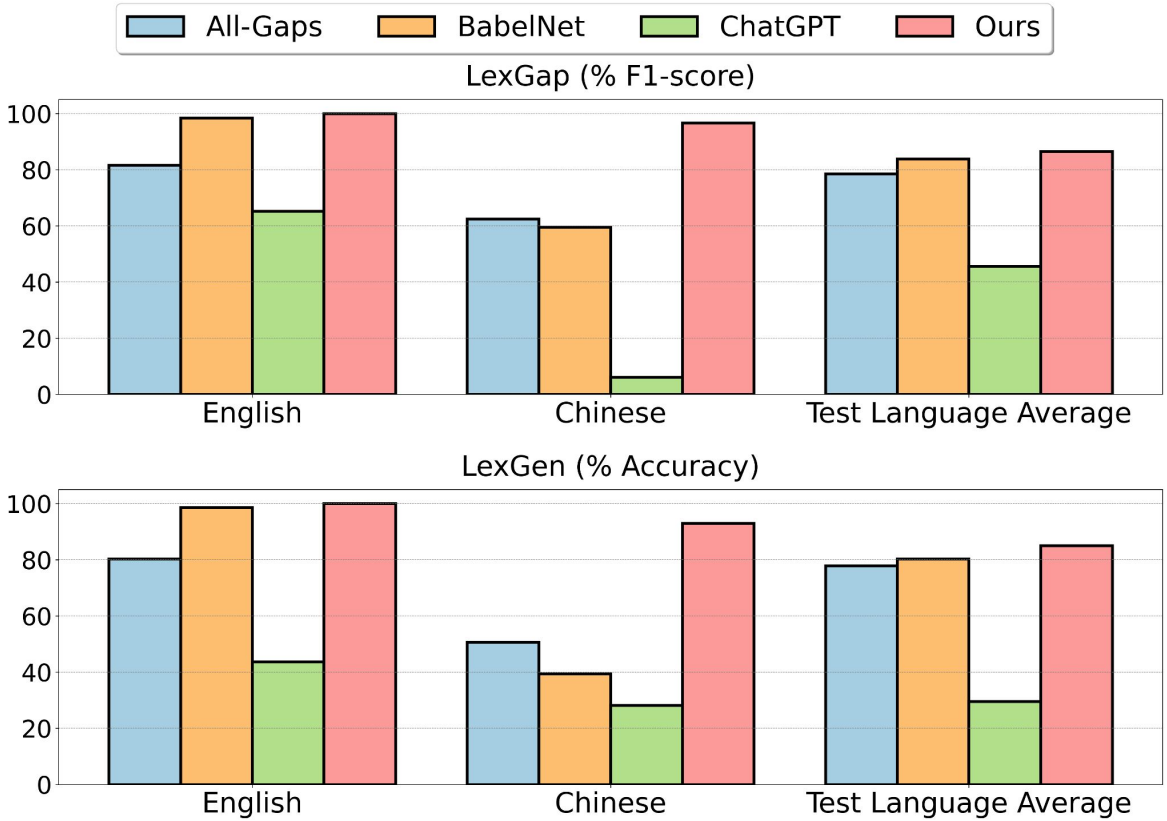
Idea: If a concept is an exclusive disjunction of its hyponym concepts then all three concepts should have different lexicalizations.

Method:

1. Generate a candidate lexicalization for each concept by translating an unambiguous lexicalization into the target language in the context of the concept gloss.
2. Then filter out incorrect translations using the above idea.

Results: Empirical evaluations demonstrate that our approach yields higher accuracy than BabelNet and ChatGPT.

Results


All-Gaps
BabelNet
ChatGPT
Ours
LexGap (% F1-score)
100
80
60
40
20
0
English
Chinese
Test Language Average
LexGen (% Accuracy)
100
80
60
40
20
0
English
Chinese
Test Language Average

Conclusion

- Novel translate-and-filter method for:
  - Generating **lexicalizations**
  - Detecting **lexical gaps**
- Grounded in linguistic theory, with clear definitions and and propositions
- Leverages translation and hypernym/hyponym relations
- Future work: Beyond kinship to other domains

github.com/UAlberta-NLP/KinshipAutoLex

Word (Token) Level Lexical Substitution as Causal Language Modeling

Ning Shi, Bradley Hauer, Grzegorz Kondrak

Alberta Machine Intelligence Institute (Amii)
Department of Computing Science
University of Alberta, Edmonton, Canada

In Proceedings of *SEM 2024

Lexical Substitution Task (LST)

LST is to identify suitable replacements for a target word while preserving the contextual meaning of the sentence.

LST(S, w_x) = y, for example:

Sentence (S) = "Let me begin again."

Target Word (w_x) = "begin"

Substitutes (y) = ["start", "commence", "open", ...]

Limitations of Prior Work

- Predicted substitutes may align with the context **BUT** change the original meaning of the sentence. Consider **M**asked **L**anguage **M**odeling (MLM):

Input S_0: "Let me begin again." → MLM → "Let me [MASK] again."

Output S_1: "Let me start again."
WordNet: (Verb) take the first step or steps in carrying out an action.

VS

Output S_2: "Let me originate again."
WordNet: (Verb) bring into being.

- Pipeline approaches, depending on defined **heuristics**, tuned **thresholds**, extensive **post-processing** steps, and external **resources**.
- A **GAP** between pre-training (language modeling) and fine-tuning (LST).

WordNet: WordNet 3.1 & Open English WordNet.

Task Definition

Lexical **Sub**stitution, **LexSub**(S, w_x, w_y) := "the word w_x can be replaced by the word w_y in the sentence S without altering its meaning"

LexSub("Let me begin again.", "begin", "start") = TRUE

Word **P**rediction, **WP**(S, w) := "the word w has the same meaning as the masked word in the sentence S"

WP("Let me [~~begin~~] again.", "start") = TRUE

Task Reduction

A **P-to-Q** reduction solves an instance of a problem **P** by combining the solutions of one or more instances of **Q**.

A **mutual** reductions of two problems to one another demonstrate their **equivalence**.

Task Reduction from LexSub to WP:

$$\textbf{LexSub}(S, w_x, w_y) \Leftrightarrow \textbf{WP}(S, w_x) \wedge \textbf{WP}(S, w_y)$$

(Hauer and Kondrak, 2023)

Method – PromptSub
Lexical Substitution via Prompt-aware Fine-tuning

InfoPrompt incorporate three additional attributes of the target word:

- Lemma form (**Target**)
- Part of Speech tag (**PoS**)
- Position in the **Context** (**Position**)

Exclusively from the task data thus **NO** reliance on external resources.

Target	PoS	Position	Context	Substitute::Frequency
begin	verb	3	Let me begin again.	start::6, commence::2, open::2, initiate::1, introduce::1, ...

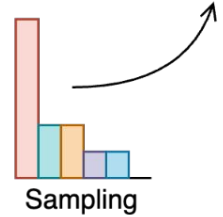
Method – PromptSub
Lexical Substitution via Prompt-aware Fine-tuning

FreqSub exploits the frequency information associated with each gold **substitute**.

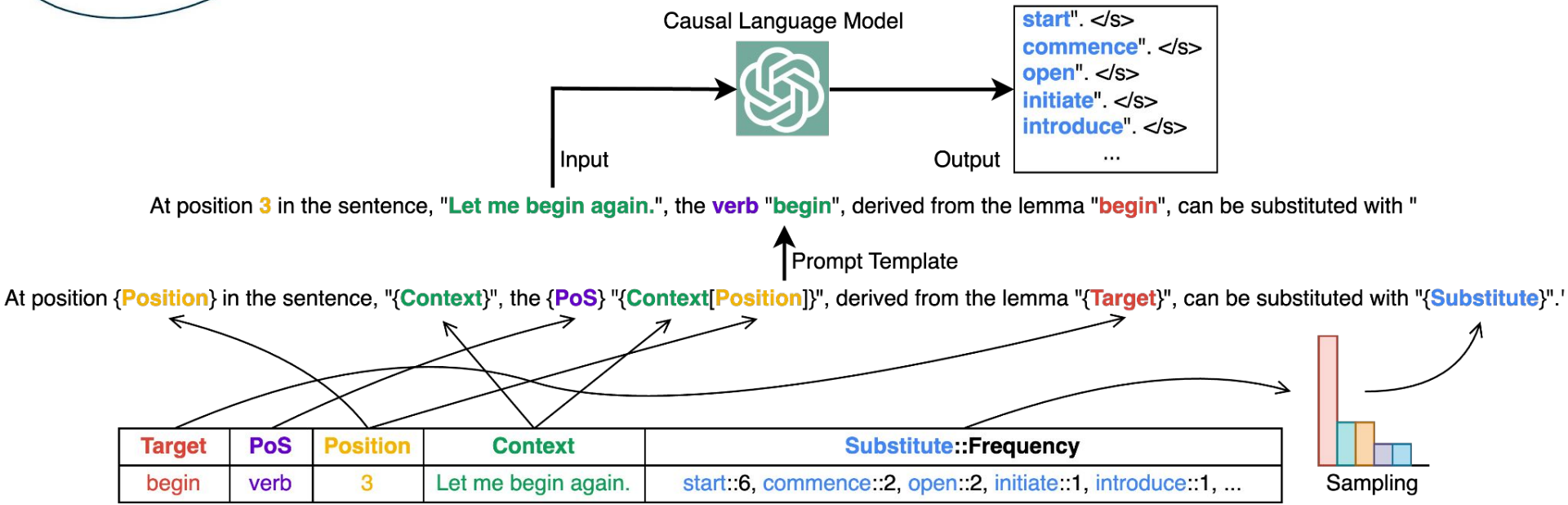
Frequency -> Softmax -> Probability Distribution -> Sampling

At position {Position} in the sentence, "{Context}", the {PoS} "{Context[Position]}", derived from the lemma "{Target}", can be substituted with "{Substitute}".'

Target	PoS	Position	Context	Substitute::Frequency
begin	verb	3	Let me begin again.	start::6, commence::2, open::2, initiate::1, introduce::1, ...



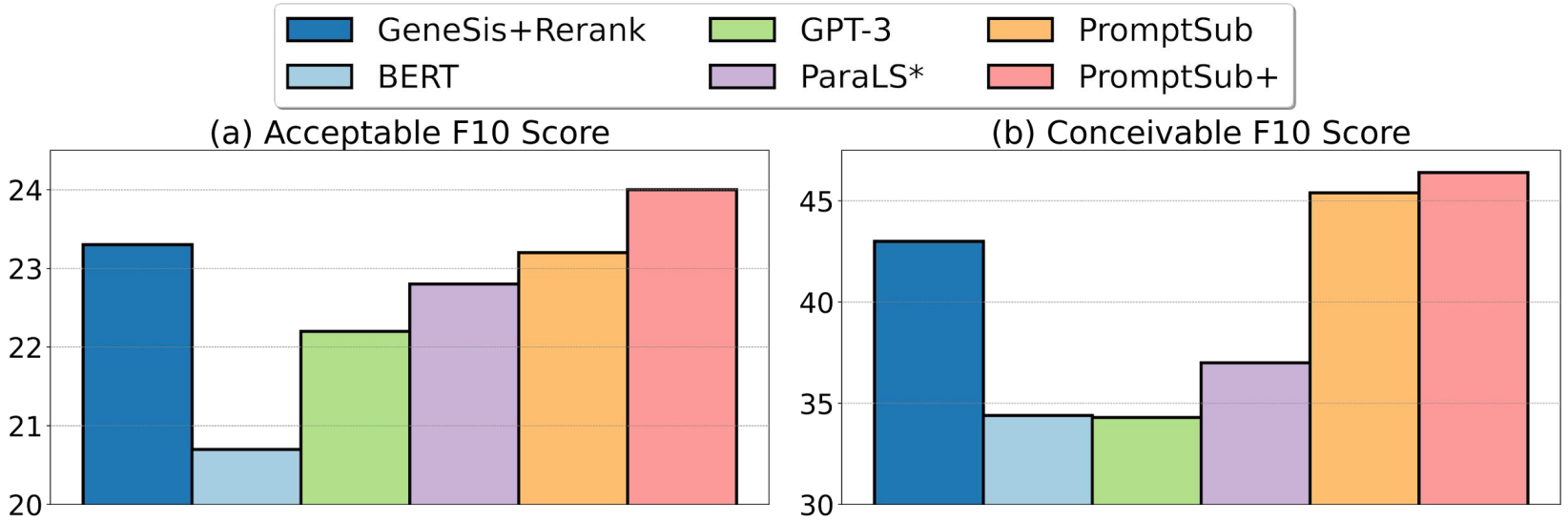
Method – PromptSub
Lexical Substitution via Prompt-aware Fine-tuning



Results – LS21

PromptSub and **PromptSub+** take GPT-2 Medium as its backbone.

GeneSis+Rerank incorporates post-processing to refine its results.



LS21 (Lee et al., 2021)

Train: SWORDS Train, LST and TWSI; Test: SWORDS Test

Conclusion

We have presented **PromptSub**, a framework reducing LST to CLM.

- **Bridges the gap** between pre-training and fine-tuning.
- Takes advantage of **greater** model capacity.
- Leverages a **broad** array of resources.
- Establishes a new overall **state of the art**, particularly LS21.

We expect to extend our approach to other semantic tasks in the future.

github.com/ShiningLab/PromptSub

Sentence Level Paraphrase Identification via Textual Inference

Ning Shi, Bradley Hauer, Jai Riley, Greg Kondrak

Alberta Machine Intelligence Institute (Amii)
Department of Computing Science
University of Alberta, Edmonton, Canada

In Proceedings of *SEM 2024

amii

Natural Language Inference

Natural **L**anguage **I**nference (NLI) involves three labels that describe the relationship between two sentences.

Entailment, Contradiction, Neutral

For example:

S_1: "This man is surfing."
S_2: "A man is on water."

Surfing: an aquatic activity or website browsing?

Paraphrase Identification

Paraphrase **I**dentification (PI) is the task of deciding whether two sentences convey the same meaning.

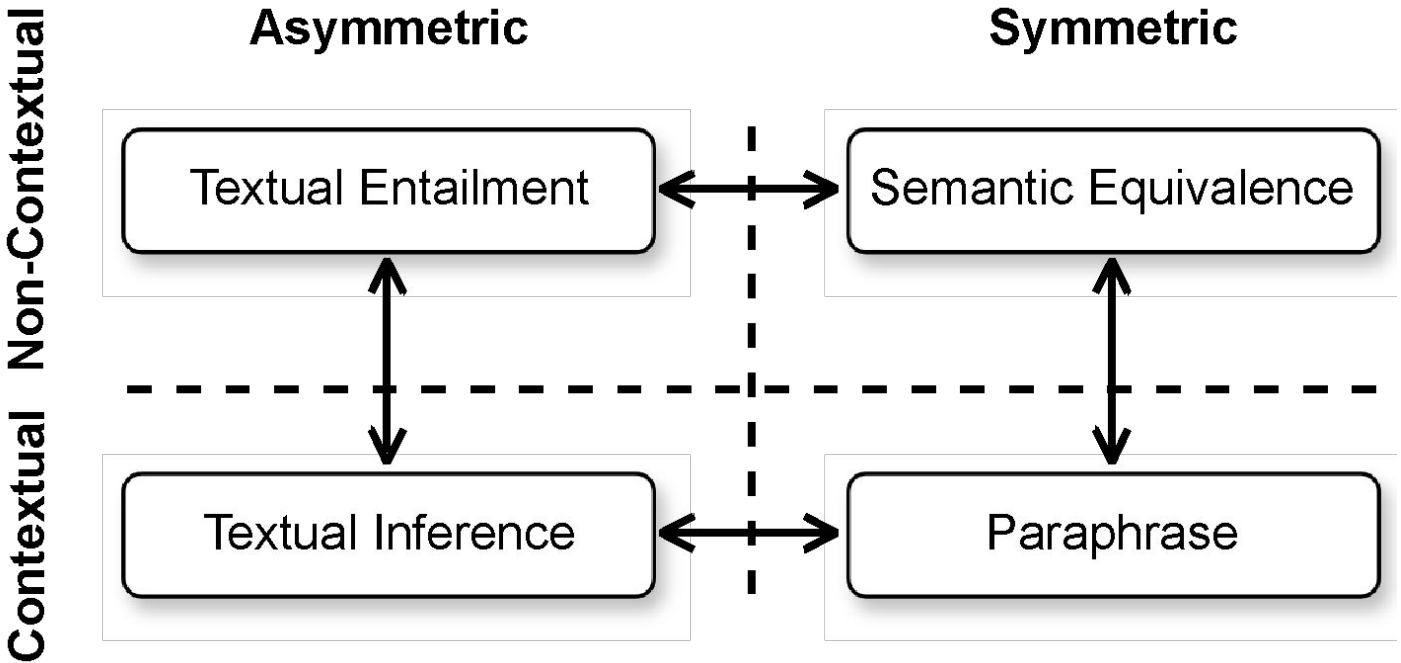
Hypothesis:

Paraphrasing corresponds to **bidirectional textual entailment**.

Prior work:

- A blend of modules complicates the analysis
- Bias to traditional PI methods
- Lacks any theoretical formalization

Sentence-level Relations



Equivalence and Paraphrasing

Semantic **Eq**uivalence relation**, SEQ(S_1, S_2)** :=
"the sentences S_1 and S_2 convey the same meaning"

PR(C, S_1, S_2) := "the sentences S_1 and S_2 convey the same meaning given the context C"

The relationship in between:

$$\mathbf{SEQ(S_1, S_2) \Leftrightarrow \forall C : PR(C, S_1, S_2)}$$

Example:

S_1: "We must work hard to win this election."
S_2: "The Democrats must work hard to win this election."

Entailment and Inference

Textual **E**ntailment, $TE(S_1, S_2)$:=
"the sentence S_2 can be inferred from the sentence S_1"

Textual **I**nference, $TI(C, S_1, S_2)$:= "the sentence S_2 can be inferred from the sentence S_1 given the context C"

The relationship in between:

$$\mathbf{TE(S_1, S_2) \Leftrightarrow \forall C : TI(C, S_1, S_2)}$$

Example:
S_1: "This man is surfing."
S_2: "A man is on water."

Proposition

Given context C, sentences S_1 and S_2 are paraphrases if and only if they can be mutually inferred from each other.

Formally:

$$\mathbf{PR(C, S_1, S_2) \Leftrightarrow TI(C, S_1, S_2) \wedge TI(C, S_2, S_1)}$$

Context:

- Context includes common sense and world knowledge.
- In practice, context is embedded in the data distribution.

Data Adaptation

Positive PI instances:

We convert each positive PI instance into two distinct NLI positive instances, one in each direction.

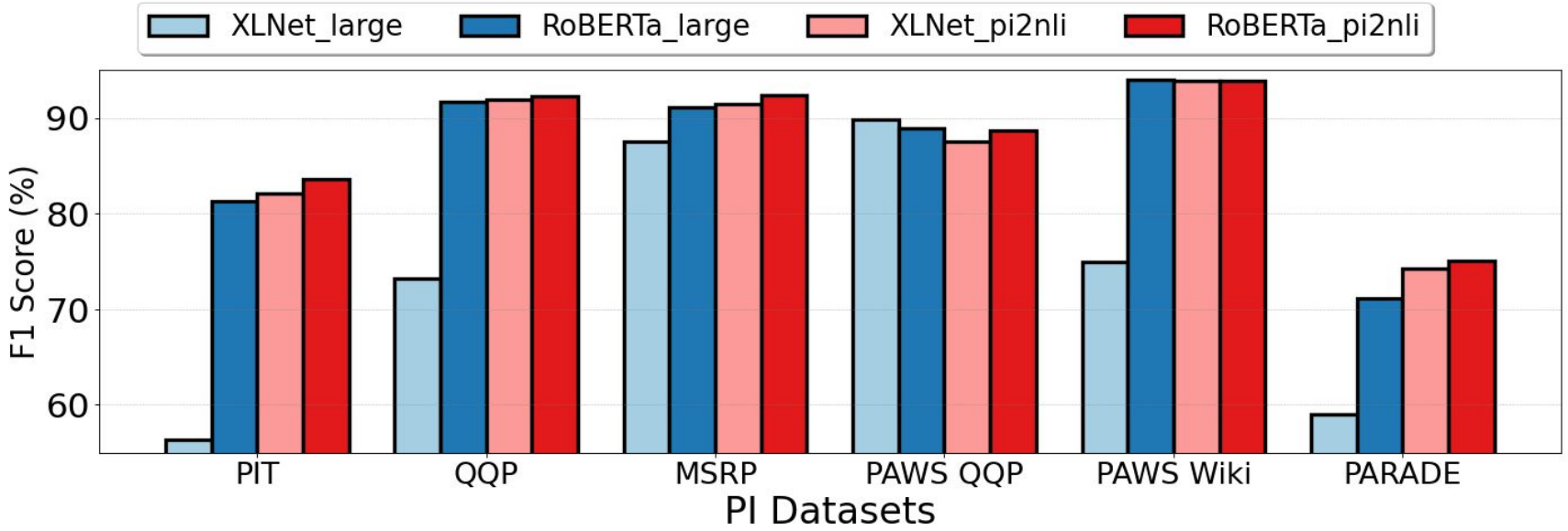
PI(S_1, S_2, True) -> NLI(S_1, S_2, Entailment) **AND** NLI(S_2, S_1, Entailment)

Negative PI instances:

We generate a negative NLI instance (randomly selected as either Contradiction or Neutral) in one randomly selected direction.

PI(S_1, S_2, False) -> NLI(S_1, S_2, Contradiction) **OR** NLI(S_2, S_1, Contradiction)
NLI(S_1, S_2, Neutral) **OR** NLI(S_2, S_1, Neutral)

Results



XLNet_large (Yang et al., 2019); RoBERTa_large (Liu et al., 2019)

PAWS QQP and PAWS Wiki include **adversarial** example created by word scrambling and back-translation.

Conclusion

We have presented PI2NLI, the first attempt to reduce PI to NLI.

- PI can be reduced to NLI **theoretically** and **empirically**.
- Fine-tuned NLI models can **outperform** PI models on PI datasets.
- Applying PI2NLI in a zero-shot setting show the **limitations** in the current PI datasets.

Future:

- Using NLI to refine PI data?
- Using PI models to solve NLI?
- Using one single model to solve both PI and NLI?

github.com/ShiningLab/PI2NLI

Cross Level?
Contradiction Detection at the Lexical Level

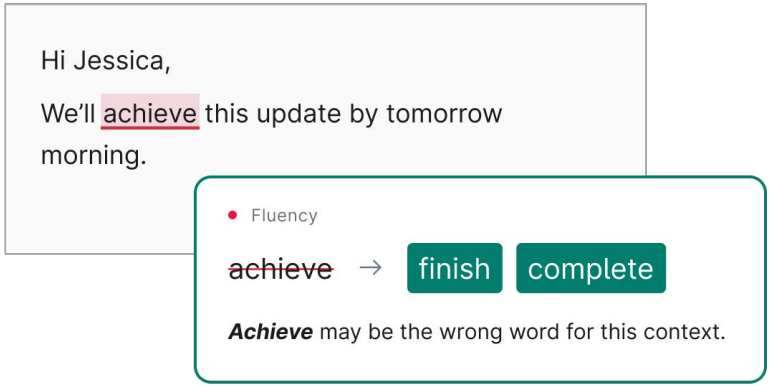
Ning Shi, Grzegorz Kondrak

Alberta Machine Intelligence Institute (Amii)
Department of Computing Science
University of Alberta, Edmonton, Canada

Word Probabilities

Many downstream tasks operate at the word level.

- Surprisal (psycholinguistic modeling of expectation)
- Perplexity (language models benchmarking)
- Scoring (fluency, reranking)
- Lexical Semantics (word sense disambiguation, lexical substitution)
- Natural Language Inference / Reasoning (I will show this later.)



Language Model / Next Token Predictor

Language Modeling

LMs are defined to assign a probability to each possible next word

Next Token Prediction

Predict the next token given the previous context

Next Word Prediction

Predict the next word given the previous context

Jurafsky, D., & Martin, J. H. (2024). Speech and language processing: An introduction to natural language processing, computational linguistics, and speech recognition (3rd ed., draft).

Example (Word -> Token)

Text String

"My name is Shining."

Token Sequence (Llama-3.2-3B)

["My", "Ġname", "Ġis", "ĠSh", "ining", "."]

Beginning of Word (BOW) Tokenizer

"Ġ" is the special bow mark representing the space.

"Shining" -> "ĠSh" + "ining"

CON2LM - Contradiction Detection

Surprisal Word

Given a text T, a word w in T is a surprisal word if after reading the text up to and including w it can be concluded that the text contains a contradiction.

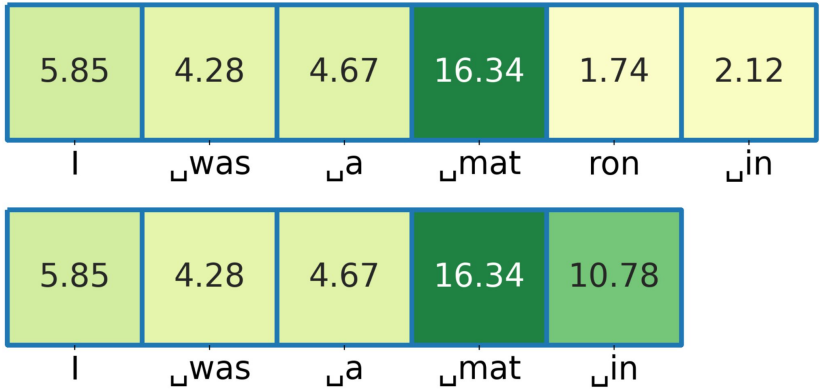
Hypothesis

A text T involves a contradiction if and only if T contains a surprisal word w.

$$\text{Surprisal}(w \mid C) = -\log \mathcal{P}(w \mid C)$$

Problems

Causal LLMs do not have an end-of-word (EOW) symbol. As the model can not tell when a word ends, so it always assumes more tokens might follow.



Surprisal values calculated with leading whitespaces.

Leading Whitespaces of Language Models' Subword Vocabulary Pose a Confound for Calculating Word Probabilities (Oh & Schuler, EMNLP 2024)

Problems

Next-token probability can include both a word and its longer extensions, so using it directly overestimates the true probability of the word.

Input: “My name is”

Step 0: “Ġ John” (0.0073)

Step 1:

“Ġand” (0.1085) -> “My name is John and”

“athan” (0.0183) -> “My name is Johnathan”

Token-to-Word Decoding

EOW Injection

We simulate an End-of-Word (EOW) event by:

Identifying BOW tokens (tokens that can start a new word)

Reallocating their probability mass to a virtual <*eow*> token

This tells the model: “you may stop generating the current word now”

$$\mathcal{P}(\text{“<}eow\text{>”} \mid \mathbf{s}^C, \mathbf{s}^w) = \sum_{s \in \mathcal{S}_{\text{bow}}} \mathcal{P}(s \mid \mathbf{s}^C, \mathbf{s}^w)$$

Token-to-Word Decoding

Dynamic Normalization

At depth 1 (start of word): restrict to BOW tokens

At depth > 1 (inside word): restrict to mid-word tokens only (no BOW)

$$\mathcal{S}' = \begin{cases} \mathcal{S}_{\text{bow}} & \text{if } d = 1 \\ \mathcal{S}_{\text{mid}} & \text{if } d > 1 \end{cases}$$

In practice, it depends on the tokenizer and the language. For example, some words include non-alphabetical characters such as “N’Djamena”.

Token-to-Word Decoding

Beam Search

We do not just want to estimate the probability of one known word.

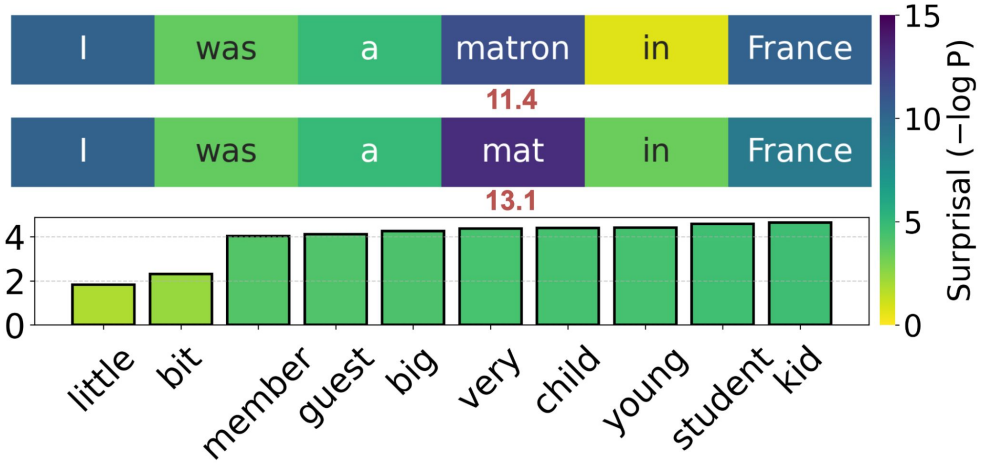
We want to explore top-k most probable word completions.

Exhaustive search over all token paths is intractable: $O(|\mathcal{S}|^D)$

We keep the top beam width candidates, so

$$O(D \cdot |\mathcal{S}'| \cdot W), \text{ where } \mathcal{S}' \ll \mathcal{S}$$

Unit Test



Surprisal values for the example proposed by Oh and Schuler (2024).

Leading Whitespaces of Language Models' Subword Vocabulary Pose a Confound for Calculating Word Probabilities (Oh & Schuler, EMNLP 2024)

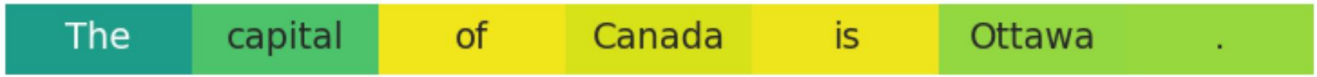
Results - Wiki Capital

Wiki Capital

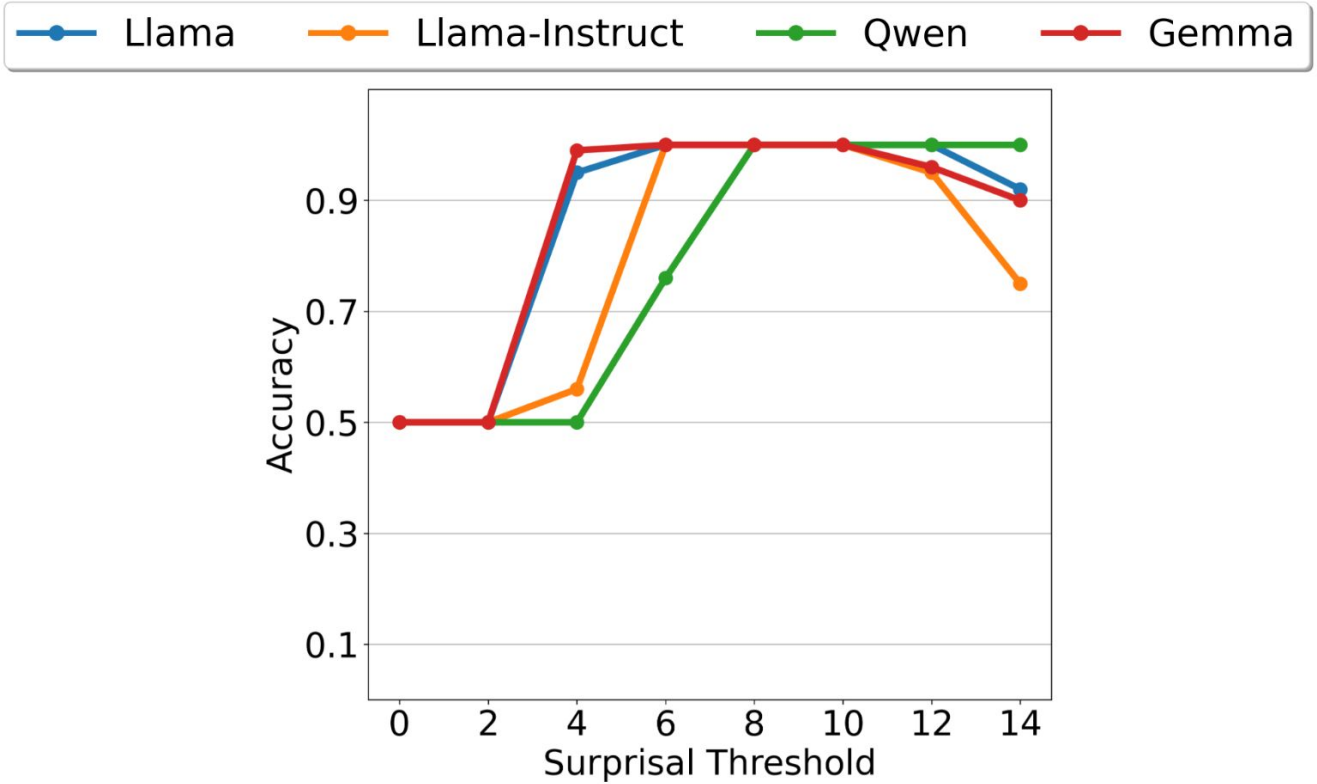
Context: None

Context: The capital of Canada is Edmonton.

Context: Since the capital of Canada is Edmonton, therefore



Results – Wiki Capital



Error Cases - bAbI

Open Challenges

Invalid Token Paths

Not all token continuations lead to valid words. And due to the nature of the softmax function, LLM will never assign zero probability to a token.

> “My name is Shining.”
>
> [“My”, “Ġname”, “Ġis”, **“ĠSh”**, **“ining”**, “.”]
>
> [“My”, “Ġname”, “Ġis”, **“ĠSh”**, **[Any Token]**, “.”]

Open Challenges

Word Probability Underestimation

While only the top (greedy) path is often used in practice, BPE tokenization allows multiple ways to form the same word.

Input: "The capital of Canada is"

Greedy path: [“ĠOttawa”] (0.2309)

A possible path: [“ĠO” (0.00015), “tt” (0.0156), “awa” (0.0047)]

Takeaways

Goal: Bridge sentence-level and word-level semantics via language modeling

Main Reductions:

- PI2NLI: Paraphrase = Mutual Entailment
- PromptSub: Lexical substitution = Causal LM
- Con2LM: Contradiction = Language Modeling

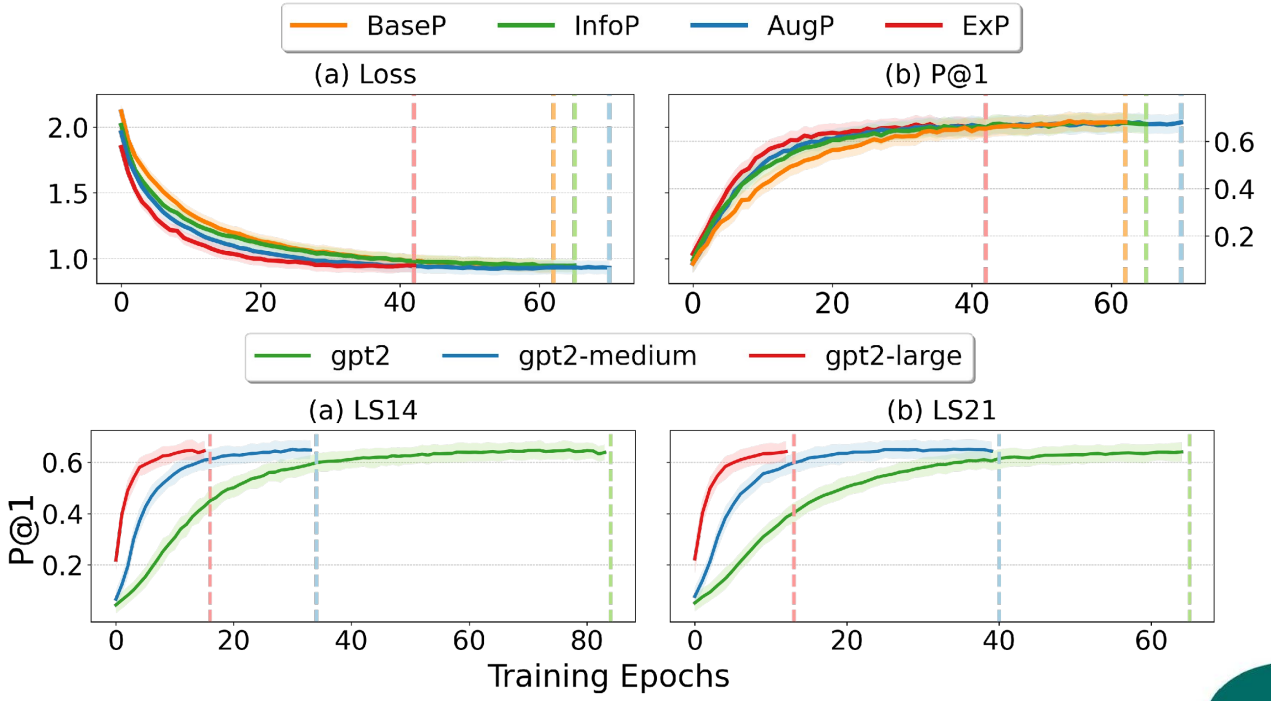
Key Insight: Language Modeling or Language Models

Impact: Performance, interpretability, and cross-task generalization

Thanks!
Q & A

PromptSub - Scalability

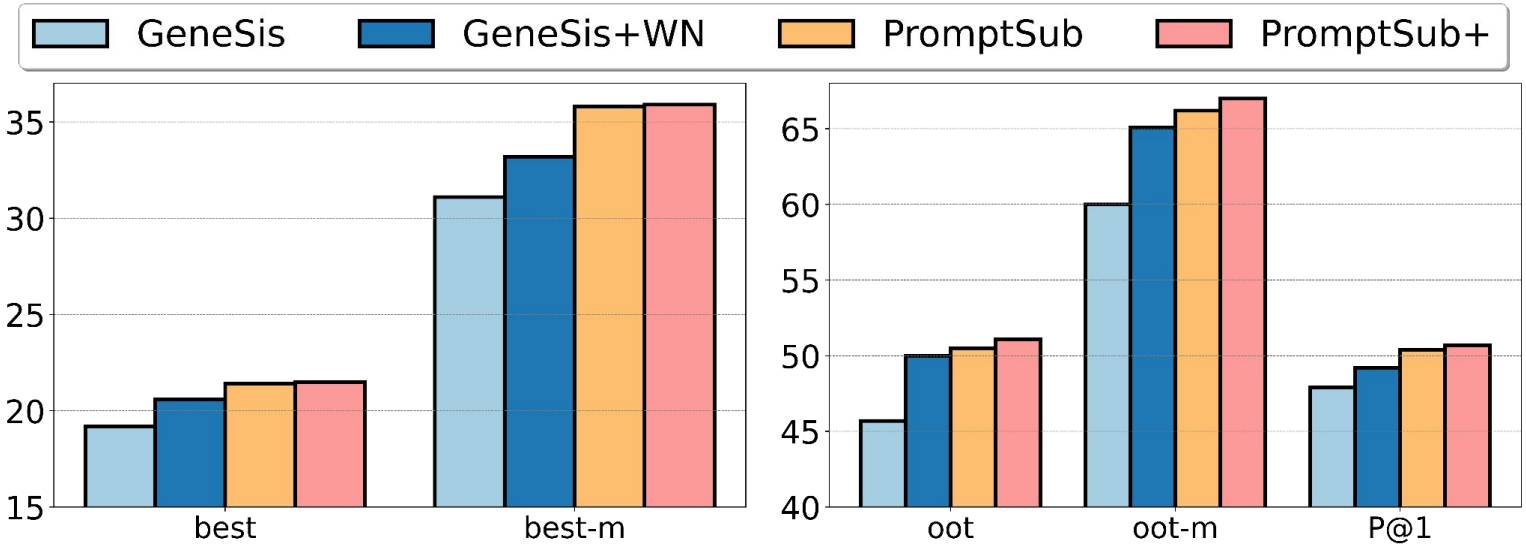
ExPrompt retrieves WordNet synsets for RAG, resulting in **lower** loss, **improved** P@1, and **earlier** convergence.



PromptSub – LS07

PromptSub+ augments the training set by incorporating the dev set.

GeneSis+WN relies on external resources from WordNet.



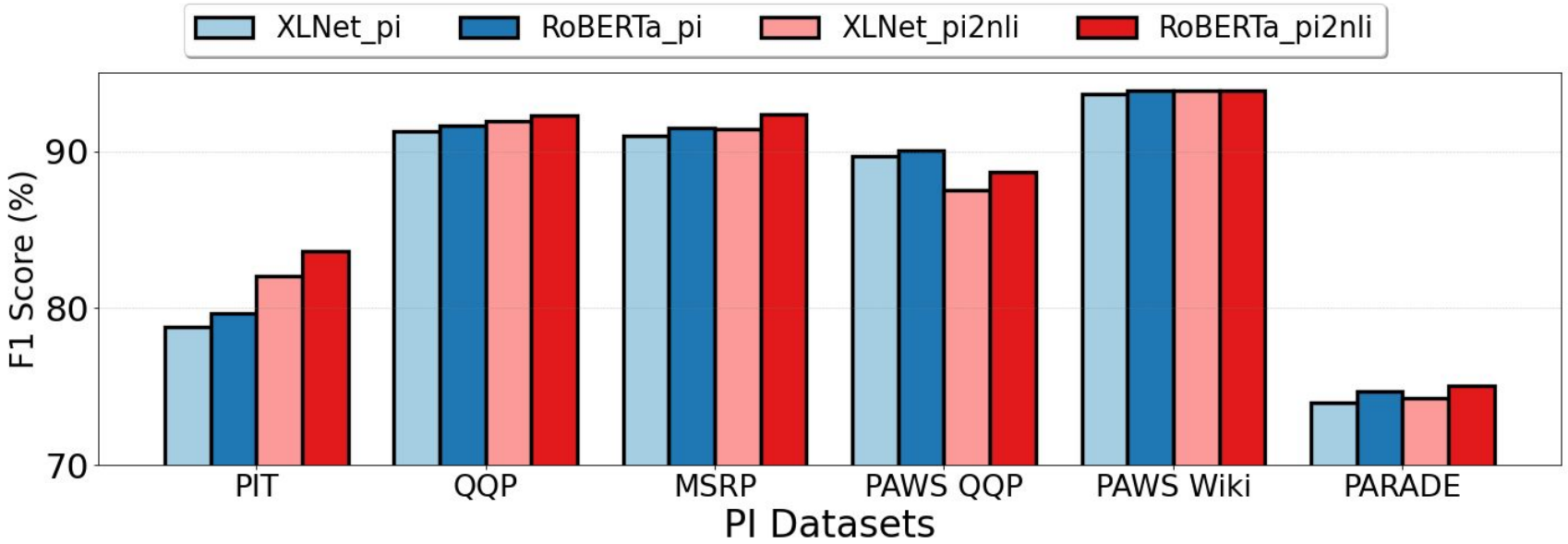
LS07 (Lacerra et al., 2021)

Train: CoInCo and TWSI; Test: Semeval-2007 Task 10

PromptSub - Examples

Instance:	let me begin again.
BaseP:	the "begin" in the sentence "let me begin again." can be substituted with "start".
InfoP:	at position 3 in the sentence, "let me begin again.", the verb "begin", derived from the lemma "begin", can be substituted with "start".
AugP (Train):	at position 3 in the sentence, "let me begin again.", the verb "begin", derived from the lemma "begin", can be substituted with "start", "commence", "open", "bring about", "carry on", "initiate", "introduce", "originate", "restart", "try".
AugP (Test):	at position 3 in the sentence, "let me begin again.", the verb "begin", derived from the lemma "begin", can be best substituted with "start".
ExP (Train):	at position 3 in the sentence, "let me begin again.", the verb "begin", derived from the lemma "begin" with synonyms "commence", "get", "get down", "lead off", "set about", "set out", "start", "start out", can be substituted with "start", "commence", "open", "bring about", "carry on", "initiate", "introduce", "originate", "restart", "try".
ExP (Test):	at position 3 in the sentence, "let me begin again.", the verb "begin", derived from the lemma "begin" with synonyms "commence", "get", "get down", "lead off", "set about", "set out", "start", "start out", can be best substituted with "start".

PI2NLI - Ablation



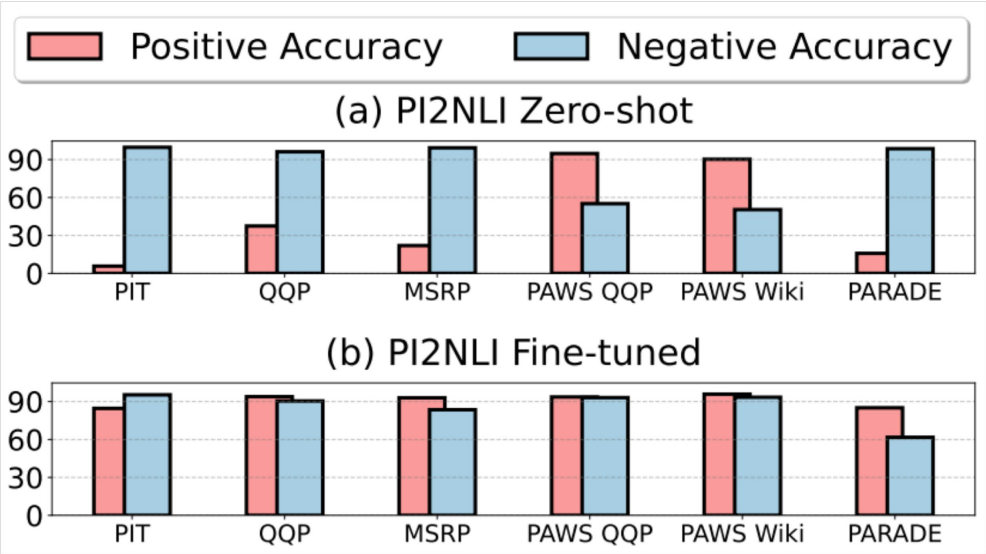
XLNet_pi, RoBERTa_pi (Nie et al., 2020)

The **same** language models with classification heads initialized from **scratch**.

PI2NLI - Analysis

A paraphrase identified in one dataset might **NOT** necessarily be considered a valid paraphrase in the other.

We view this **adjustment** as the process of how models learn the **context** inherent in each PI dataset.



Token-to-Word Decoding

Algorithm 1 Token-to-Word Decoding

Input: tokenized context $\mathbf{S}^C$, beam width W beam depth D, language model $\mathrm{F}(\cdot)$

$B_0 \leftarrow \{\langle 0, \mathbf{S}^C \rangle\}$
for $d \in \{1, \ldots, D\}$:
  $B \leftarrow \varnothing$
  for $\langle p, \mathbf{S} \rangle \in B_{d-1}$:
    if $\mathbf{S}$.last() = "<*eow*>" :
      B.add($\langle p, \mathbf{S} \rangle$); **continue**
    $\mathcal{P} \leftarrow \mathrm{F}(\mathbf{S})$
    if $d = 1$:
      $\mathcal{P} \leftarrow \mathrm{Normalize}(P, \mathcal{S}_{\text{bow}})$
    else:
      $\mathcal{P} \leftarrow \mathrm{InjectEOW}(\mathcal{P}, \mathcal{S}_{\text{bow}})$
      $\mathcal{P} \leftarrow \mathrm{Normalize}(\mathcal{P}, \mathcal{S}_{\text{mid}})$
    for $(p', s) \in \mathrm{Top}(\mathcal{P}, \mathrm{W})$:
      B.add($\langle p \cdot p', \mathbf{S} \circ s \rangle$)
  $B_d \leftarrow B$.top(W)
return B_D.sort()