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

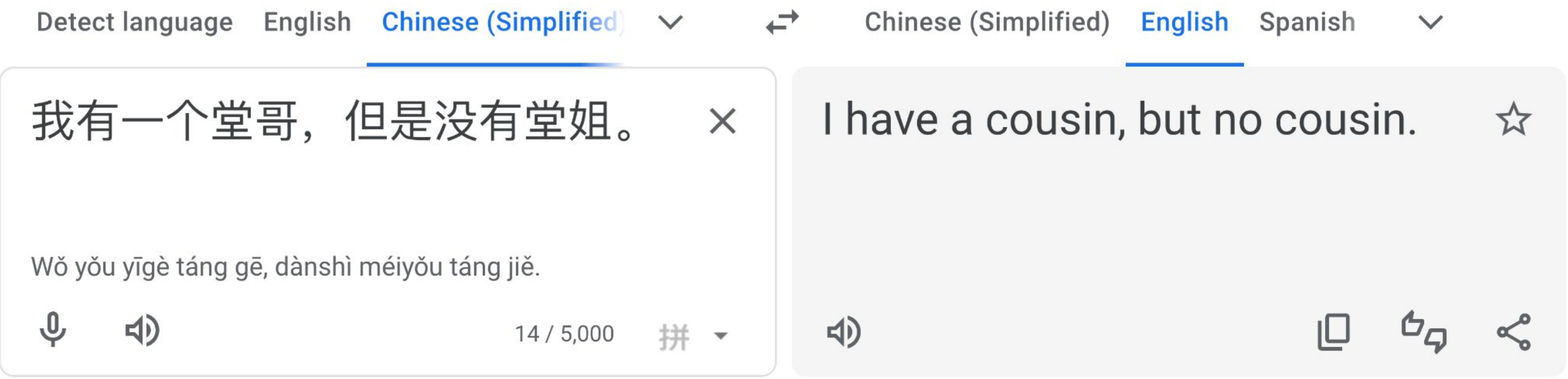
Senyu Li, Bradley Hauer, Ning Shi, Grzegorz Kondrak

Alberta Machine Intelligence Institute, Dept of Computing Science
University of Alberta, Edmonton, Canada

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.)



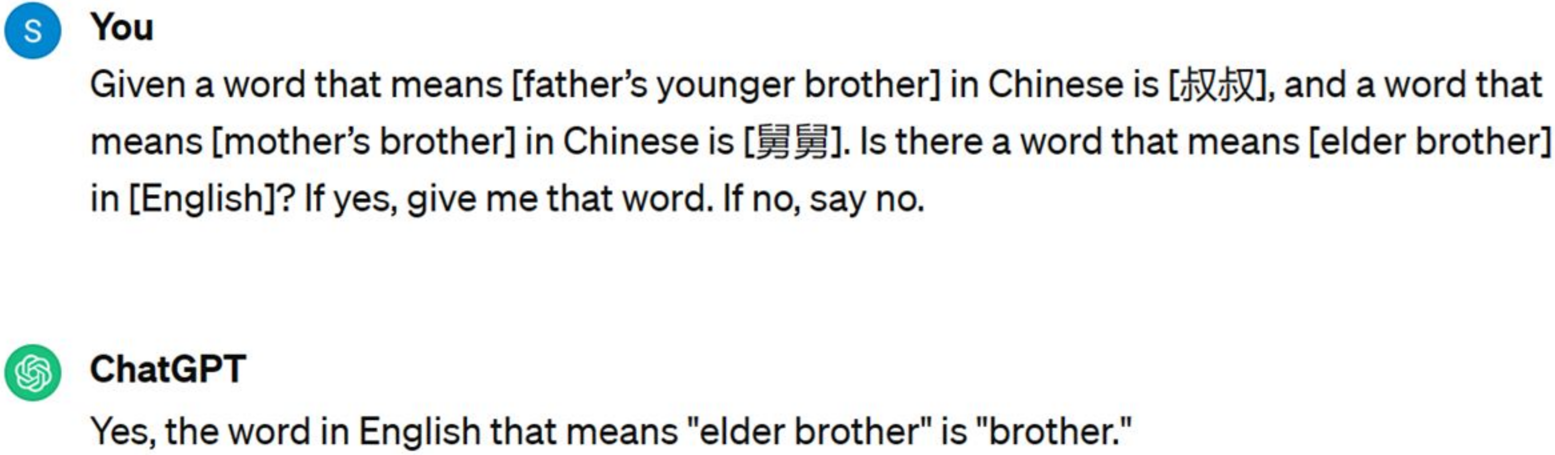
我有一个堂哥，但是没有堂姐。

Wǒ yǒu yīgè táng gē, dànshì méiyǒu táng jiě.

I have a cousin, but no cousin.

*Google Translate, February 15, 2024.

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, February 15, 2024.

Outline

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

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

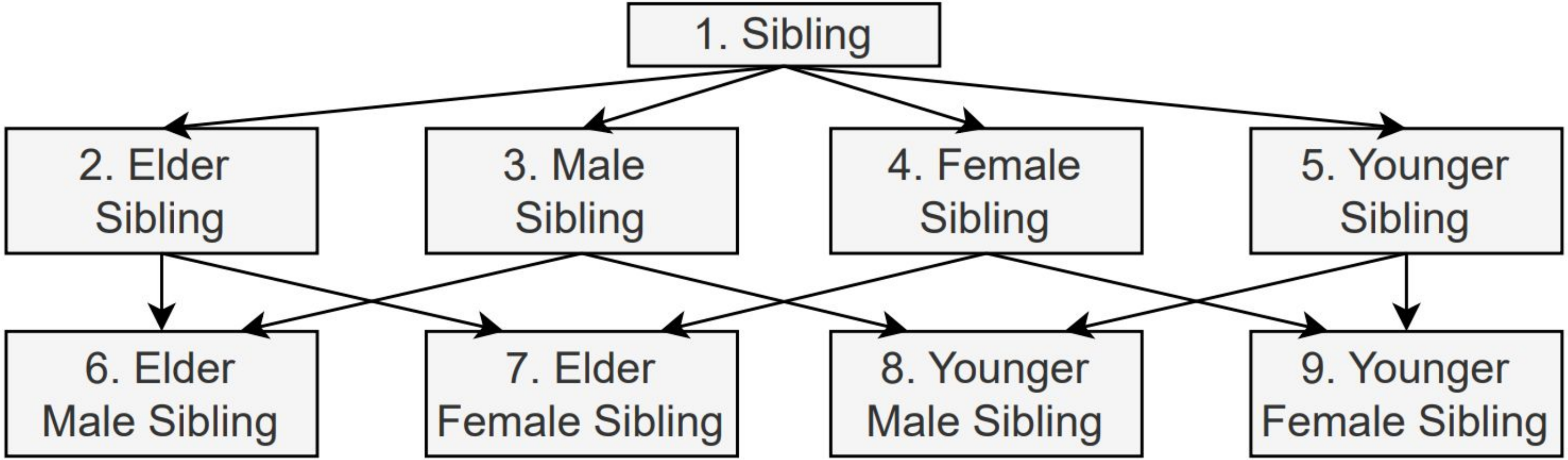
- **Method:** Generate a candidate lexicalization for each concept by translating an unambiguous lexicalization into the target language in the context of the concept gloss. Then filter out incorrect translations using the theory. .

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

Concepts

- **Concept: discrete word meaning**

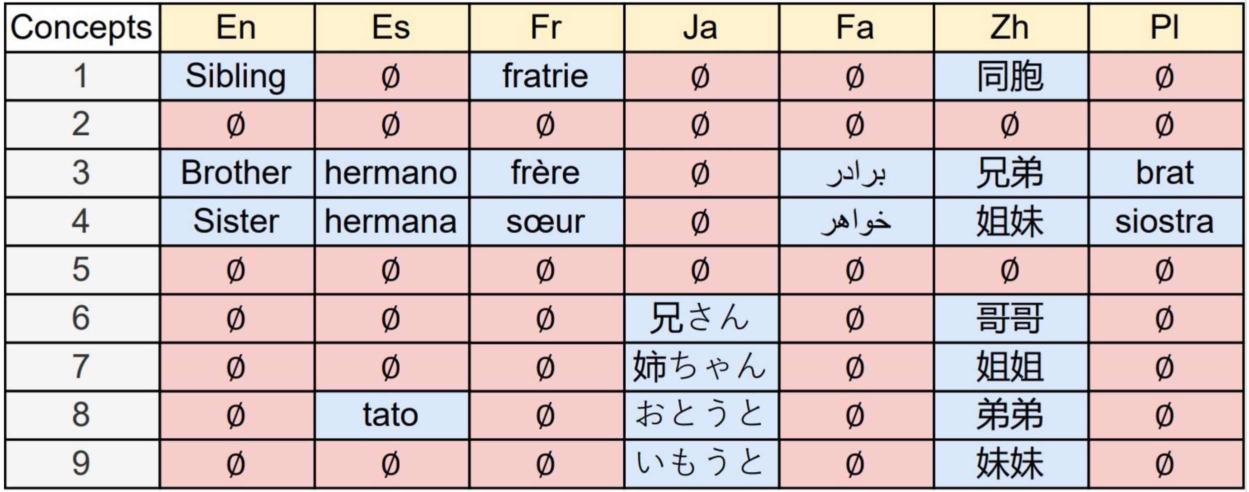
- Kinship concepts have clear definitions and hierarchical structure



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	∅	fratrie	∅	∅	同胞	∅
2	∅	∅	∅	∅	∅	∅	∅
3	Brother	hermano	frère	∅	برادر	兄弟	brat
4	Sister	hermana	sœur	∅	خواهر	姐妹	siostra
5	∅	∅	∅	∅	∅	∅	∅
6	∅	∅	∅	兄さん	∅	哥哥	∅
7	∅	∅	∅	姉ちゃん	∅	姐姐	∅
8	∅	tato	∅	おとうと	∅	弟弟	∅
9	∅	∅	∅	いもうと	∅	妹妹	∅

*Using Linguistic Typology to Enrich Multilingual Lexicons: the Case of Lexical Gaps in Kinship (khishigsuren, 2022)

Task definition: LexGen and LexGap

- **LexGen:** Lexicalization Generation
    - Given a language L and a concept s
    - LexGen(L, s) returns a word in L which lexicalizes s
    - or a special GAP token indicating that no such word exists

- **LexGap:** Lexical Gap Detection
    - Given a language L and a concept s
    - LexGap(L, s) returns TRUE if L has no word that lexicalizes s
    - FALSE otherwise.

- LexGap returns TRUE if and only if LexGen returns a GAP.

Proposition 1

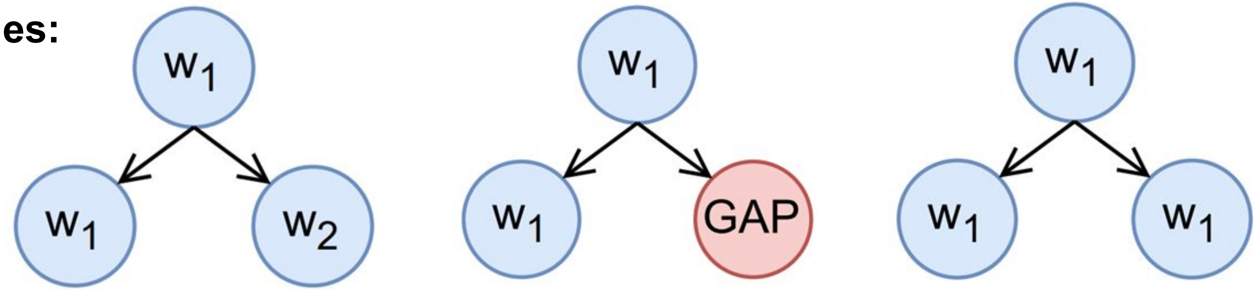
If a concept P is an exclusive disjunction of its hyponym concepts C1 and C2, expressing P and C1 with the same word w can result in a colloquial contradiction.

Proof: C2 could be expressed by a phrase “w but not w”, This phrase intuitively corresponds to a logical contradiction: $w(x) \wedge \neg w(x)$.

Example:

Robin is my parent but not my father => Robin es mi padre pero no mi padre

Excluded triples:



*This Example was obtained from Google Translate accessed on February 15, 2024

Proposition 2

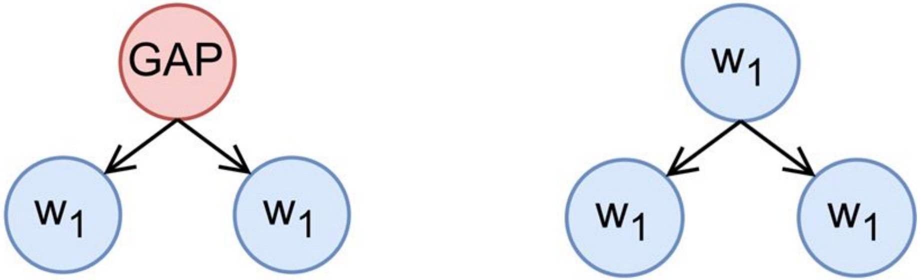
If a concept P is an exclusive disjunction of its hyponym concepts C1 and C2, expressing C1 and C2 with the same word w can result in a colloquial contradiction.

Proof: P could be expressed by a phrase “either w or w”, this phrase intuitively corresponds to a logical contradiction: $w(x) \oplus w(x)$.

Example:

Tengo una prima pero no tengo ningún primo => I have a cousin but I have no cousin

Excluded triples:

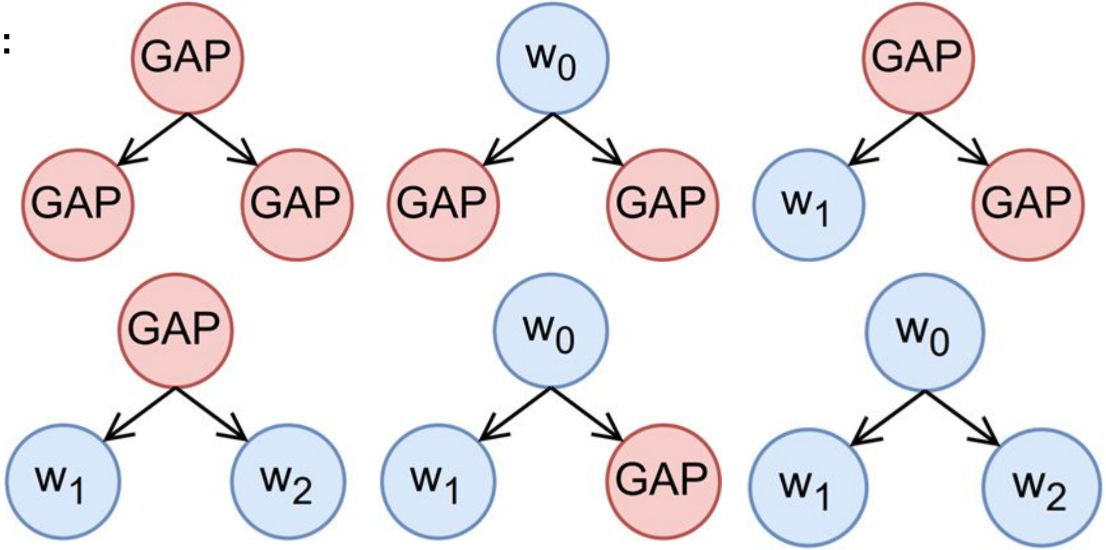


*This Example was obtained from Google Translate accessed on February 15, 2024

Corollary

If a concept P is an exclusive disjunction of its hyponyms C1 and C2 then all their lexicalizations should be different.

Remaining triples:



Our Method

Generate a candidate lexicalization for each concept by translating a seed word into the target language in the context of the concept gloss. Then Apply 4 filters sequentially to the obtained translations.

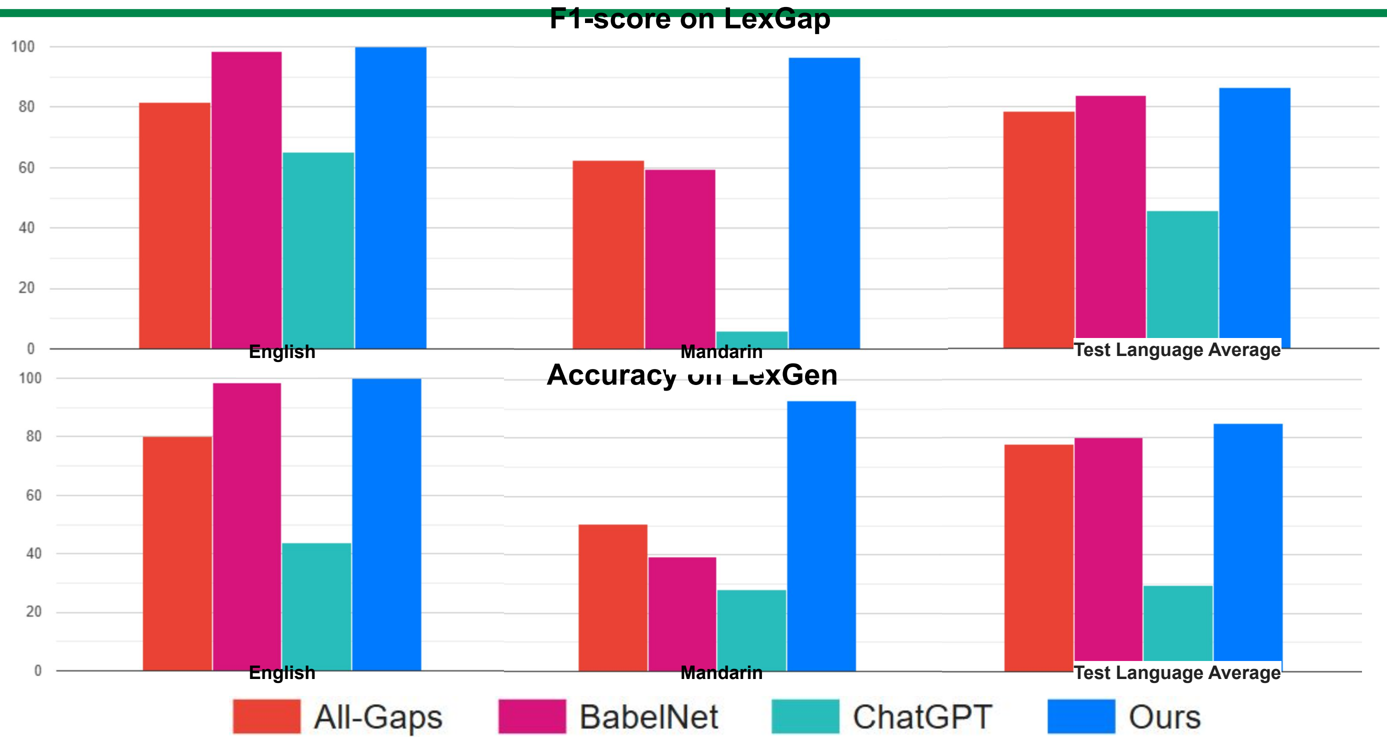
- **Multi-word filter**
  - **for** each concept s **do** $L_1(s) \leftarrow$ GAP **if** $L_0(s)$ is not a word
- **Horizontal filter (backboned by proposition 2)**
  - **for** each triple (s0, s1, s2) **do** $L_2(s1) \leftarrow$ GAP; $L_2(s2) \leftarrow$ GAP **if** $L_1(s1) = L_1(s2)$
- **Back-translation filter**
  - **for** each concept s **do** $L_3(s) \leftarrow$ GAP **if** BackTrans($L_2(s)$, gloss(s)) $\neq$ seed(s)
- **Vertical filter (backboned by proposition 1)**
  - **for** each triple (s0, s1, s2) **if** $L_3(s0) = L_3(s1)$ **then**
    if $L_3(s2) =$ GAP **then** $L_4(s1) \leftarrow$ GAP **else** $L_4(s0) \leftarrow$ GAP

Experimental Setup

- **Data:** *Database of Lexical Diversity in Kinship* by Khishigsuren et al. (2022)
- **Translator:** Google Translate
- **Metrics:** Accuracy for LexGen and F1 score for LexGap
- **Comparison:** All-Gaps, BabelNet (v. 5.1), and ChatGPT (GPT-3.5 Turbo).
- **Languages**
  - Development languages: English, Mandarin, and Persian.
  - Test languages: Spanish, Russian, French, German, Polish, Arabic, Italian, Mongolian, Hungarian, and Hindi.

*GPT-3.5 Turbo and Google Translate were accessed on February 15, 2024

Results


F1-score on LexGap
100
80
60
40
20
0
English
Mandarin
Test Language Average
Accuracy on LexGen
100
80
60
40
20
0
English
Mandarin
Test Language Average
All-Gaps
BabelNet
ChatGPT
Ours

Conclusion

- A novel translation-based method that generates concept lexicalizations and detects lexical gaps.

- Our method is grounded in formal definitions and propositions, and leverages translation and hypernym/hyponym taxonomy relations.

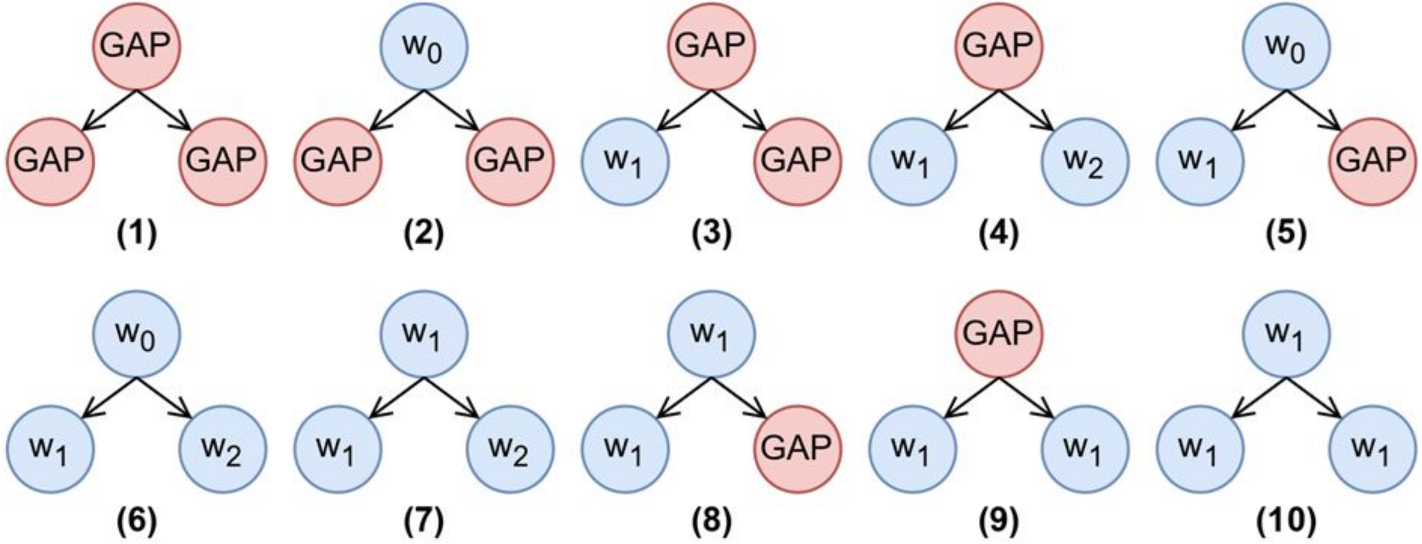
- Future work:
  - Apply our method to other domains
  - Employ large language models

github.com/UAlberta-NLP/KinshipAutoLex

Disjunctive Triples

Kinship concepts can often be arranged into triples

Concept sp is an exclusive disjunction of hyponym concepts s1 and s2.


GAP
GAP
GAP
(1)
w_0
GAP
GAP
(2)
GAP
w_1
GAP
(3)
GAP
w_1
w_2
(4)
w_0
w_1
GAP
(5)
w_0
w_1
w_2
(6)
w_1
w_1
w_2
(7)
w_1
w_1
GAP
(8)
GAP
w_1
w_1
(9)
w_1
w_1
w_1
(10)