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
{senyu,bmhauer,ning.shi,gkondrak}@ualberta.ca

Abstract

Constructing lexicons with explicitly identified lexical gaps is a vital part of building multilingual lexical resources. Prior work has leveraged bilingual dictionaries and linguistic typologies for semi-automatic identification of lexical gaps. Instead, we propose a generally-applicable algorithmic method to automatically generate concept lexicalizations, which is based on machine translation and hypernymy relations between concepts. The absence of a lexicalization implies a lexical gap. We apply our method to kinship terms, which make a suitable case study because of their explicit definitions and regular structure. Empirical evaluations demonstrate that our approach yields higher accuracy than BabelNet and ChatGPT. Our error analysis indicates that enhancing the quality of translations can further improve the accuracy of our method.

1 Introduction

In computational semantics, the term *lexicon* refers to the vocabulary which speakers of the language use to express concepts. A language L *lexicalizes* a concept s if it has a lexeme (or a word) that can express s; otherwise s corresponds to a *lexical gap* in L (Murphy and Koskela, 2010). For example, the Polish word *pojutrze* is a *lexicalization* of the concept "the day after tomorrow" which corresponds to a lexical gap in English. In particular, kinship terms describe familial relations such as "grandparent" and "female cousin". The clear definitions (*glosses*) and regular hierarchical structure of kinship concepts make them well-suited for investigations into lexicons and lexical gaps.

The task of identifying lexicalizations for a given concept underlies the automatic construction of *multilingual wordnets* (Martelli et al., 2023), lexical knowledge bases modeled after the Princeton WordNet (Miller et al., 1990). Wordnets are comprised of synonym sets (*synsets*), each of which

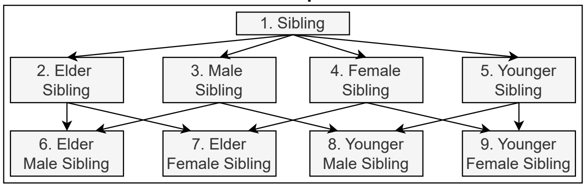
Zh: 我有一个堂哥，但是我没堂姐。	
En: I have a cousin, but I have no cousin.	

Table 1: An incorrect translation produced by Google Translate. 堂哥 means "elder *son* of father's brother" and 堂姐 means "elder *daughter* of father's brother".

corresponds to a single concept, and contains the set of words which can express that concept. Multilingual wordnets constructed by expanding the synsets of the Princeton WordNet, such as MultiWordNet (Pianta et al., 2002), Universal Wordnet (De Melo and Weikum, 2009), BabelNet (Navigli and Ponzetto, 2010), and Open Multilingual Wordnet (Bond and Foster, 2013), may misrepresent terms and concepts that correspond to English lexical gaps (Kwong, 2018). For example, the synset designated `bn:00023333n` in BabelNet 5.2 includes English *cousin*, as well as both Spanish *prima* "female cousin" and *primo* "male cousin", which are clearly *not* synonymous. Furthermore, translation models may fail to correctly handle lexical gaps, as in the example in Table 1.

Prior work on semi-automatic identification of concept lexicalizations and lexical gaps has leveraged bilingual dictionaries, wordnets and linguistic typologies. Bentivogli and Pianta (2000) apply a decision tree approach based on information from a machine-readable bilingual dictionary, but their experiments are restricted to English and Italian. Gregori and Panunzi (2017) focus on a subset of bilingual action verbs in the context of video-based translation, but establish no mapping to wordnet-type concepts. Khishigsuren et al. (2022) compile a dataset of 1911 kinship terms and a list of lexical gaps in 176 languages by combining native speaker expertise in 10 languages, lexicalization information from Wiktionary, and a set of typological patterns of Murdock (1970). We leverage their resource to develop a method which is fully

Concepts



Lexicalizations

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	∅

Figure 1: An illustration of the hierarchical structure of kinship concepts (top), with the corresponding concept-language matrix of lexicalizations (bottom).

automated, language-independent, and not specific to kinship terms.

The principal idea behind our approach is to reduce the task of lexical gap detection to the task of lexicalization generation. The latter can be viewed as populating a concept-language matrix (Figure 1) in which each cell contains a lexicalization of the concept in the corresponding language, if one exists. Given a correctly populated lexicalization matrix, empty cells correspond to lexical gaps. The task of lexicalization generation differs from the task of synset population in that the former requires at most one rather than *all* concept lexicalizations.

We propose a translation-based method for lexicalization generation, which we also apply to lexical gap detection. Our method is grounded in theoretical propositions based on the hypernym/hyponym relationships between concepts, which provide a basis for detecting incorrect lexicalizations. In particular, our model predicts that languages tend to avoid ambiguity between lexicalizations within disjunctive triples of concepts such as *parent/mother/father*. We develop a method for lexical gap detection via filtering concept lexicalizations produced by translating unambiguous "seed" words in the context of the corresponding concept glosses. We leverage existing lexical knowledge bases and machine translation systems, as well as the tree structure of kinship concepts, to decide if a given translation is literal or indicative of a lexical gap. While we focus on kinship terms in this work, our approach is also applicable to other domains.

Our evaluation on kinship terms across 10 languages demonstrates substantial improvements over BabelNet and ChatGPT. BabelNet fails to represent many of the concepts that are not lexicalized in English, while ChatGPT exhibits a tendency to generate overly specific or irrelevant responses. We identify three main causes of errors made by our algorithm: inaccurate translations, non-standard terms, as well as errors and omissions in the benchmark dataset itself. We release our code and BabelNet concept mapping on GitHub.

2 Theoretical Framework

We start this section by discussing the linguistic background related to the issue of lexical gaps. We then formally define the tasks addressed in this paper, as well as related theoretical concepts, such as literal translations, seed words, and disjunctive concept triples. This is followed by propositions and proofs that form the theoretical basis of our method.

2.1 Linguistic Background

Chomsky and Halle (1965) introduce the distinction between *accidental gaps* (words that could theoretically exist) and *systematic gaps* (words that would contravene phonological constraints). Lehrer (1974) discusses several types of gaps: phonological, morphological, syntactic, paradigmatical, derivational, functional, and semantic. Ivir (1977) questions the utility of systemic gaps, and focuses instead on *lexical gaps*, which correspond to culture-specific concepts, and *conceptual gaps*, which correspond to "universal" concepts. The latter type, which includes kinship terms, is considered more important, being an inter-language rather than intra-language phenomenon.

In the context of translation, Cvilikaitė (2006) defines lexical gaps as instances of lack of lexicalization for a given concept, and emphasizes the importance of identifying them prior to translation. Janssen (2004) observes that lexical gaps correspond to words for which there is no single-word translation in a target language. For example, the concept expressed by the Russian word *goluboj* "light blue" is considered a lexical gap in English, even though it can be approximately translated with a single word *blue*. According to Gouws (2002), a translation dictionary entry for a culture-specific lexical gap needs to include a "brief paraphrase of meaning" (i.e., *gloss*) and/or a "loan word" (source language term); e.g., "*bobotie, South African curried mincemeat*".

2.2 Definitions

A *wordnet* is a semantic knowledge base composed of synonym sets, or *synsets*. Each synset corresponds to a unique concept, and to a different sense of each word that it contains. Each synset is associated with a part of speech, and a gloss that describes the meaning of the concept. Each word in a synset can express (i.e., *lexicalizes*) the corresponding concept.

Hauer and Kondrak (2023b) define a theoretical binary problem Sense(w, s) for deciding whether the word w can express the concept s. A word lexicalizes a concept if it can express the meaning conveyed by the concept's gloss. For example, unlike the English compositional phrase *female cousin* or the Spanish word *prima*, the English word *cousin* on its own cannot express the concept of "female cousin", which is defined as "the daughter of your aunt or uncle". A method that solves the Sense problem could theoretically be used to populate any wordnet synset, by testing each word in the lexicon on whether it can express the concept corresponding to that synset.

We define the task of *lexicalization generation* (LexGen) as follows: given a language L and a concept s, a method must return either a word in L which lexicalizes s, or a special GAP token indicating that no such word exists. For example, the word *prima* is a possible return value of LexGen(SPA, "female cousin"). The LexGen task is reducible to the Sense problem by returning any word in L for which Sense(w, s) is TRUE, or GAP if no such word exists.

We define the binary task of *lexical gap detection* (LexGap) as follows: given a language L and a concept s, LexGap(L, s) returns TRUE if L has no word that lexicalizes s, or FALSE otherwise. For example, LexGap(ENG, "female cousin") returns TRUE, as there is no word in the English lexicon to express the concept. LexGap is reducible to LexGen in a straightforward manner by returning TRUE if and only if LexGen returns GAP. LexGap can also be reduced directly to Sense:

$$LexGap(L, s) \Leftrightarrow \forall w \in L : \neg\text{Sense}(w, s)$$

A *literal translation* is an expression in the target language that preserves the meaning of the expression in the source language in a given context. In the case of a *literal lexical translation*, the target word expresses the same concept as the source word. Following Hauer and Kondrak (2023a) we assume that a translator, which can be either a human or a machine, is guided by the following priorities: (1) fidelity (meaning preservation), (2) brevity (conciseness), and (3) fluency. Therefore, a translator prefers literal to non-literal translations, as well as single-word translations to multi-word phrases. In the case of a lexical gap, a literal lexical translation is not an option. Both non-literal and phrase translations can therefore be considered indications of lexical gaps in the target language. For example, Spanish *prima* can be translated into *female cousin* (phrasal translation) or just *cousin* (non-literal translation). A heuristic for detecting non-literal lexical translations is the *back-translation test*: a source word w in context C is first translated into a target word w', which is then translated back in the same context into a source language word w''; the test succeeds if and only if $w'' = w$. For example, *cousin* as a translation of *prima* may fail the back-translation test.[1]

We introduce a notion of *seed words*, defined as words that lexicalize exactly one concept within a set of concepts. For example, the Spanish word *prima* is considered a seed word for the concept of "female cousin" within the set of kinship terms. We use seed words in Section 4 as unambiguous source words to generate target concept lexicalizations via translation.

2.3 Disjunctive Triples

Simple natural language statements can often be mapped to symbolic logic, and vice versa, with the logical operators represented by conjunctions such as *and*, *or*, and *not*. In particular, an apparent colloquial or textual contradiction can often be expressed as a logical proposition that is false for all values of its variables. For example, the statement *"Robin is brave and not brave"* intuitively corresponds to $brave(x) \wedge \neg brave(x)$, where the variable x represents *Robin*. We refer to such natural language expressions as *colloquial contradictions*.

Since the typological phenomena that underlie the hypernymy graph of kinship terms are binary, kinship concepts can often be arranged into triples, wherein a concept s_0 is an *exclusive disjunction* of its hyponym concepts s_1 and s_2. Among the kinship terms, the principal type of exclusive disjunction is *gender*; for example, a *sibling* is either a *sister* or *brother*. The gender distinction can be

[1] Google translates *"amo a mi* prima*"* into *"I love my* cousin*"*, and then back into *"amo a mi* primo*"* (accessed April 18, 2024).

indirect; for example, an *uncle* can be referred to as either *maternal* or *paternal*. Another type of disjunction is *relative age*; for example, a *cousin* can be either younger or older. Other distinctions are also possible, such as the speaker's gender, or consanguinity vs. affinity.

Because hyponymy is the IS-A relation, any instance of s_0 must be either an instance of s_1 or s_2 (but not both). If a single word w could express both s_1 and s_2, then w would also necessarily express the hypernym s_0. To avoid confusion, if a speaker specifically wishes to refer to concept s_1, as opposed to its hypernym s_0, it is logical to choose a word (or phrase) which excludes s_2. For example, since the Spanish word *padre* can lexicalize both concepts of "father" and "parent" (especially in its plural form), speakers may instead use the word *progenitor* to express the latter concept.

In symbolic logic, an exclusive disjunction is expressed by the XOR (exclusive OR) operator: $\oplus$. In plain English, an exclusive disjunction can be expressed as *"either _ or _"*. if a concept s_0 is an exclusive disjunction of its hyponyms s_1 and s_2, the phrase that combines the glosses of the hyponyms as"either C_1 or C_2" is a possible gloss for s_0. For example, since "parent" is an exclusive disjunction of its hyponyms "father" and "mother", it can be defined as "father or mother".

2.4 Propositions

In the remainder of this section, we present two propositions formulated on exclusive disjunctive triples, which form the basis of our methods in Section 4 for removing spurious lexicalizations.

Proposition 1. *If a concept s_0 is an exclusive disjunction of its hyponym concepts s_1 and s_2, then expressing both s_0 and s_1 with the same word can result in a colloquial contradiction.*

Proof. Suppose that there exists a word w that lexicalizes both concept s_0 and its hyponym s_1. Since s_1 and s_2 are disjunctive hyponyms of s_0, the meaning of s_2 could be expressed by a phrase "w but not w", in which w is used in two different senses of s_0 and s_1. This phrase intuitively corresponds to a logical contradiction: $w(x) \wedge \neg w(x)$. □

Intuitively, the use of the same word to lexicalize both members of a hypernym/hyponym pair can lead to highly ambiguous expressions, which is undesirable in any natural language. For example, since Spanish *padre* can mean both "parent" and "father", Google translates the English sentence *"Robin is my parent but not my father"* into *"Robin es mi padre pero no mi padre"*. Contextual disambiguation of such apparently contradictory statements is particularly difficult if the two concepts are closely related by hyponymy.

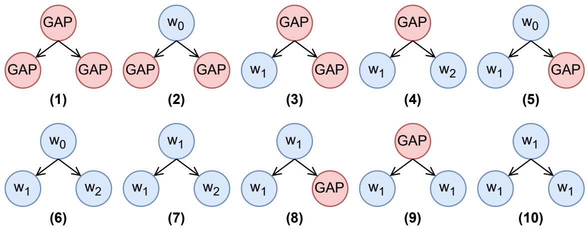


Figure 2: Types of concept triples. Distinct lexicalizations are represented by different variables.

Proposition 2. *If a concept s_0 is an exclusive disjunction of its hyponym concepts s_1 and s_2, then expressing both s_1 and s_2 with the same word can result in a colloquial contradiction.*

Proof. Suppose that there exists a word w that lexicalizes both s_1 and s_2. Since s_1 and s_2 are disjunctive hyponyms of s_0, the meaning of s_0 could be expressed by a phrase "either w or w", in which w is used in two different senses of s_1 and s_2. This phrase intuitively corresponds to a logical contradiction: $w(x) \oplus w(x)$. □

For example, the concepts of "female cousin" and "male cousin" which are lexicalized in Spanish by *prima* and *primo*, respectively, correspond to lexical gaps in English. Given the Spanish sentence *"Tengo una prima pero no tengo ningún primo"*., Google[2] produces a translation which is at best ambiguous, at worst nonsensical: *"I have a cousin but I have no cousin"*.

Taken together, Propositions 1 and 2 yield the following corollary, which applies to disjunctive triples of concepts, based on the intuition that colloquial contradictions should be rare.

Corollary 1. *If a concept s_0 is an exclusive disjunction of its hyponyms s_1 and s_2, then all their lexicalizations should be different.*

Figure 2 shows 10 possible types of exclusive disjunctive triples, of which 4 types (numbered 7-10) are excluded by Corollary 1 because two or more concepts are lexicalized by the same word. Types 7, 8, and 10 fall under Proposition 1, while types 9 and 10 fall under Proposition 2.

[2] translate.google.com, accessed February 15, 2024.

3 Kinship Taxonomies

Unlike prior work, we generate our kinship taxonomy algorithmically. The generated taxonomy is a superset of the kinship terms in the *Database of Lexical Diversity in Kinship Domain* of Khishigsuren et al. (2022), henceforth referred to as the Kinship Database.

3.1 Kinship Database

The Kinship Database compiles lexical data from 699 languages pertaining to 198 kinship concepts, divided into six subdomains: *siblings*, *grandchildren*, *grandparents*, *uncles/aunts* (*auncles*) *nephews/nieces* (*niblings*), and *cousins* (c.f., Table 2). It explicitly lists over 37 thousand lexical gaps, based on various resources and inference methods, as well as 1911 lexicalizations, from 168 of the 699 languages.

For each of the six concept categories, more specific concepts are derived by the application of mutually exclusive distinctions. Each distinction induces two hyponyms of a given concept, which together form an exclusive disjunctive triple (Section 2.3). For example, the application of the *relative age* distinction to the concept “sibling” yields the concepts “elder sibling” and “younger sibling”. This property is crucial, as it admits the application of Propositions 1 and 2 in our method.

The 198 concepts in the Kinship Database do not include all possible concepts that could be derived by the application of the *gender* and *age* distinctions, because the creators of the Kinship Database excluded concepts that were not attested in their sources. Furthermore, 74 terms are distinguished only by the gender of the speaker; we do not consider such terms to denote separate concepts.

3.2 Concept Generation Algorithm

Our kinship taxonomy is composed of six directed acyclic graphs (DAGs): nodes represent concepts, while edges represent the IS-A relationship between hypernyms and hyponyms. Each concept corresponds to an ordered list of atomic kinship relations: *child*, *parent*, and *sibling*. The list of relations is specific to each DAG, as shown in Table 2. For example, the list of relations for the root concept “cousin” is [*child, sibling, parent*], which corresponds to a synthetic gloss “child of sibling of parent”.

Each relation in the ordered list can have a value. The three atomic relations admit the gender distinction (i.e., *male* vs. *female*). The age distinction (i.e., *younger* vs. *elder*) is relative either to the speaker (when referring to siblings or cousins) or to “sibling of parent”. For example, the concept “younger, male child of sibling of female parent” (that is, “son of mother’s sibling, younger than the speaker”) is represented by [*child* = *male, sibling* = *undefined, parent* = *female, age* = *younger*]. Concepts that have the same representation are considered identical, so that there is at most one node in any graph with a given representation.

Root Concept	Synthetic Gloss	Nodes	KD
sibling	“sibling”	9	9
grandchild	“child of child”	9	9
grandparent	“parent of parent”	9	9
auncle	“sibling of parent”	27	23
nibling	“child of sibling”	27	21
cousin	“child of sibling of parent”	81	53
Total		162	124

Table 2: The root concepts, their generated glosses, and the size of the generated DAGs vs. the number of concepts in the Kinship Database (KD).

Figure 3 shows our algorithm which generates a complete directed acyclic graph (DAG) given one of the root concepts from Table 2. Graph G is initialized with its root concept, in which every relation is set to *undefined*. The algorithm maintains a queue Q which contains the nodes to be expanded by setting each of the available relations to either of its two possible values. Each iteration of the innermost *foreach* loop results in the creation of a directed edge between the current node s_0 and one of its hyponyms s_1. If the hyponym node s_1 has not yet been created, it is added to the graph and the queue.

When applied to the six root concepts in Table 2, our algorithm in Figure 3 generates DAGs which include all 124 distinct concepts in the Kinship Database, as well as 38 additional concepts; the full list of concepts is listed in our mapping resource. An example of a generated concept which is not included in the Kinship Database is “child of younger sibling”. Furthermore, a gender-neutral concept “sibling of parent” (*auncle*, “aunt or uncle”) is lexicalized in the Kinship Database only in the constructed languages Esperanto, Ido, and Volapük. It is an open question whether such concepts are lexicalized in any natural language, but more gender-independent concepts are expected to be introduced in the future.

4 Methodology

In this section, we describe our approach to lexicalization generation. The essence of the method is to generate a candidate lexicalization for each concept by translating a seed word into the target language in the context of the concept gloss, and then apply a series of filters to remove incorrect candidates. For each concept, we output the corresponding lexicalization if it has not been filtered out, or GAP otherwise.

4.1 Candidate Generation

Given the seed word for a concept, we translate the seed word in the context of the concept gloss using the template "*[seed word]: [concept gloss]*" which we refer to as a *gloss context*. The gloss provides the translation system with additional context for the seed word, which yields more accurate translations. Ideally, the translation of the seed word should be a proper lexicalization of the input concept.

After translating the gloss context into the target language, we extract the candidate lexicalization by retrieving the part of the translation before the colon. For example, to identify a French lexicalization for the concept of "aunt", we translate the gloss context *"aunt: a parent's sister"* into *"tante: la sœur d'un parent"*. We then extract the lexicalization candidate *tante*.

4.2 Candidate Filtering

Translation errors and lexical gaps may lead to inaccurate, non-literal, or non-lexical translations, which are not appropriate as lexicalizations. We therefore apply a sequence of filters to remove incorrect candidate lexicalizations. The pseudo-code of the algorithm is shown in Figure 4.

Multi-Word Filter (#1) The multi-word filter rejects any candidates which are composed of multiple word tokens. This effectively enforces a strict definition of a lexicalization as a single orthographic word.[3] We found that multi-word expressions, such as *female cousin* are usually compositional, and therefore not suitable as lexicalizations. Since the Chinese language does not separate words orthographically, we detect multi-word expressions by identifying characters which are indicative of word boundaries: 的 and 或.

[3]Some linguists adopt an even stricter definition which stipulates that a lexicalization must be a mono-morphemic word (Khishigsuren et al., 2022).

G.create(); Q.create()
$s_r \leftarrow$ concept($root$)
G.addNode(s_r); Q.enqueue(s_r)
while not Q.isEmpty() **do**
  $s_0 = Q$.dequeue()
  for each undefined *relation* in s_0 **do**
    for each possible *value* of *relation* **do**
      $s_1 \leftarrow$ concept(s_0)
      s_1.relation $\leftarrow$ $value$
      if $s_1 \notin G$ **then**
        G.addNode(s_1); Q.enqueue(s_1)
      G.addEdge(s_0, s_1)

Figure 3: The algorithm for generating a concept graph.

Horizontal Filter (#2) The horizontal filter is based on Proposition 2 from Section 2.4, which implies that both hyponym concepts within a disjunctive triple are unlikely to share the same lexicalization. This filter considers pairs of hyponyms, rather than individual concepts, in order to detect non-literal translations. If both hyponyms in a disjunctive triple are found to have the same candidate lexicalization, it is deemed to reflect a non-literal, hypernym translation, implying the existence of two lexical gaps. For example, if the Spanish terms *primo* "male cousin" and *prima* "female cousin" are both translated into English as *cousin*, the horizontal filter replaces both instances of *cousin* with GAP indicators.

Back-Translation Filter (#3) If the candidate lexicalization can indeed express the same concept as the seed word in the context of its gloss, it should be possible to recover the seed word by *back-translating* the candidate in the context of the translated gloss. The back-translation filter is designed to detect and remove non-literal translations that fail this test. If the original seed word is not recovered, the candidate is discarded, and the output for that concept will be a lexical gap. For example, if the Chinese seed word 弟弟 "younger brother" is translated into English as *brother*, and then back-translated into Chinese as 兄弟 "brother" then the filter discards the English translation, which effectively labels this concept as a lexical gap in English.

Vertical Filter (#4) Our final filter is based on Proposition 1 from Section 2.4, which implies that a concept and its hyponym within a disjunctive triple are unlikely to share the same lexicalization. If such a case is detected, we need to decide which

```
for each concept s do
  L0(s) ← Translate(seed(s), gloss(s))
for each concept s do                                  ▷ Filter #1
  L1(s) ← GAP if L0(s) is not a word
for each triple (s0, s1, s2) do                        ▷ Filter #2
  L2(s1) ← GAP; L2(s2) ← GAP if L1(s1) = L1(s2)
for each concept s do                                  ▷ Filter #3
  L3(s) ← GAP if BackTrans(L2(s), gloss(s)) ≠ seed(s)
for each triple (s0, s1, s2) do                        ▷ Filter #4
  if L3(s0) = L3(s1) then
    if L3(s2) = GAP then L4(s1) ← GAP else L4(s0) ← GAP
```

Figure 4: Pseudo-code of the algorithm for lexicalization generation and lexical gap detection. All lexicalizations are copied from L_{i-1} to L_i before Filter #i is applied. L_4 contains the final lexicalization predictions.

of the two instances of the lexicalization should be filtered out. Intuitively, we expect languages to be consistent in their lexicalization patterns; for example, if a language has no word for “elder brother” it is less likely to lexicalize “elder sister”. Therefore, we remove the translation of the hyponym if the translation of its co-hyponym has already been removed by an earlier filter; otherwise, we remove the translation of the hypernym instead (Figure 4).

5 Experiments

This section is devoted to the empirical evaluation of our method, and includes information on our datasets, resources, metrics, and baselines. Our principal dataset is the Kinship Database described in Section 3. We develop our method on 3 languages: English (eng), Chinese (zho), and Persian (pes). We test it on 10 languages: Spanish (spa), Russian (rus), French (fra), German (deu), Polish (pol), Arabic (ara), Italian (ita), Mongolian (mon), Hungarian (hun), and Hindi (hin). For each language, we test on the set of 71 concepts which are relatively well represented in the Kinship Database.

5.1 Seed Words and Glosses

In order to generate a candidate lexicalization for a given concept, we construct a gloss context (Section 4.1) by concatenating a seed word with a concept gloss in the same language. We then extract the candidate lexicalization from the translation of the gloss context obtained with the *Googletrans* Python library.

We select the seed words from the set of words that lexicalize exactly one concept in the Kinship Database. If there is more than one such word, we prefer highly-resourced languages, which are likely to yield more accurate translations.[4] 53 out of 71 seed words are from one of our three development languages.

For most concepts, we use the glosses provided in the Kinship Database, such as “elder daughter of mother’s sibling”. The exceptions are the six root concepts in Table 2, for which we instead retrieve glosses from BabelNet (Navigli and Ponzetto, 2012). For concepts with non-English seed words, we use ChatGPT to translate the glosses into the language of the seed word, following the template in Table 4. We manually verified that the Chinese gloss translations are correct.

5.2 Evaluation and Comparison Methods

We test our method against the Kinship Database on both lexicalization generation (LexGen) and lexical gap detection (LexGap). For LexGen, we compute accuracy as the proportion of instances for which the predicted lexicalization (or a lack of it) matches the information in the Kinship Database. For LexGap, we evaluate the results with the standard F-score measure, the harmonic mean of precision and recall. The absence of a lexicalization is considered to be an indication of a lexical gap.

We compare our method with three approaches: (1) BabelNet lookup, (2) ChatGPT, and (3) a naive majority-class baseline (All-Gaps), which simply predicts that all concepts are lexical gaps in any language. We perform BabelNet lookup by retrieving lexicalizations from BabelNet. We manually identified 28 BabelNet synsets which correspond to concepts in the Kinship Database.[5] From each such synset, we take the first single word in the target language as the lexicalization for that concept. If the synset contains no single word in the target language, or there is no synset associated by our mapping, a lexical gap indicator is returned instead.

Finally, ChatGPT involves directly querying a large language model for either a lexicalization or an explicit confirmation that the concept is a lexical gap. To this end, we use *in-context learning* (Brown et al., 2020), a technique allowing large language models to execute tasks based on examples included in their input instructions, without the need for external updates or specific model training. We prompt ChatGPT with the template specified in Table 4.

[4] We follow the order of language coverage in BabelNet v5.3, https://babelnet.org/statistics.

[5] We include the synset mapping in our resource.

LexGap (F1)														
Method	**eng**	**zho**	**pes**	**spa**	**rus**	**fra**	**deu**	**pol**	**ara**	**ita**	**mon**	**hun**	**hin**	**Test Avg.**
All-Gaps	81.6	62.4	79.5	81.7	83.8	75.0	82.2	81.9	84.4	71.4	**92.5**	61.3	**70.5**	78.5
BabelNet	98.4	59.5	76.7	85.7	91.2	87.7	93.1	80.6	85.7	**83.0**	89.7	**80.9**	60.5	83.8
ChatGPT	65.2	6.1	39.1	57.1	43.9	40.0	80.8	46.8	56.4	42.1	27.9	50.0	11.4	45.6
Ours	**100.0**	**96.6**	**88.6**	**98.3**	**96.9**	**93.1**	**98.4**	**85.7**	**85.7**	82.0	90.4	74.5	59.7	**86.5**
LexGen (Acc.)														
Method	**eng**	**zho**	**pes**	**spa**	**rus**	**fra**	**deu**	**pol**	**ara**	**ita**	**mon**	**hun**	**hin**	**Test Avg.**
All-Gaps	80.3	50.7	76.1	81.7	83.1	74.6	81.7	78.9	**85.9**	71.8	**91.5**	66.2	**63.4**	77.9
BabelNet	98.6	39.4	69.0	85.9	88.7	88.7	84.5	77.5	77.5	80.3	88.7	**77.5**	53.5	80.3
ChatGPT	43.7	28.2	32.4	36.6	14.1	38.0	40.8	28.2	36.6	29.6	15.5	32.4	23.9	29.6
Ours	**100.0**	**93.0**	**83.1**	**98.6**	**97.2**	**93.0**	**97.2**	**85.9**	81.7	**84.5**	90.1	69.0	53.5	**85.1**

Table 3: Results for lexical gap detection and lexicalization generation (in %).

5.3 Results

Table 3 shows the results for lexical gap detection and lexicalization generation. On average, our system is more accurate than other methods, and achieves the best results on the majority of the test languages. In particular, it performs extremely well on high-resource European languages, such as Spanish and German. Contrariwise, lower-resource languages such as Hungarian and Hindi prove more difficult. We speculate that these trends are due to varying quality of automatic translations and data.

In terms of the comparison approaches, the results of the All-Gaps baseline are surprisingly strong. However, it must be remembered that the All-Gaps baseline would be useless for any practical application, as it cannot produce any lexicalizations. This result therefore reflects the imbalanced nature of the data, in which most instances are lexical gaps. For example, the concept “younger son of mother’s sibling” corresponds to a lexical gap in every tested language except Chinese. Similarly, we found that the BabelNet baseline performs well because most concepts in the Kinship Database are not mapped to any BabelNet synset, resulting in a large number of gap predictions.

The accuracy of ChatGPT is poor. We found that it often provides overly specific terms. For example, for “male cousin”, it returns the Chinese lexicalization 堂兄, which specifically refers to “elder son of father’s brother”, a hyponym of “male cousin”. We speculate that this phenomenon is related to the well-known problem of *hallucination*, in which large language models favor the production of incorrect answers, rather than indicating a lack of knowledge, or that no good answer exists.

Overall, the results demonstrate that our method yields highly competitive performance on both tasks across a diverse set of languages. Our approach of generating and filtering lexical translations is able to accurately identify lexical gaps where they exist, and produce lexicalizations where they do not, even on low-resource languages, outperforming methods based on existing multilingual knowledge bases or large language models. We interpret these results as strong evidence of the utility of our method, as well as for the soundness of our theoretical model.

5.4 Error Analysis

Inspecting the output of our method, we found three main types of errors. The primary factor is imperfect translations. For instance, the Chinese translation generated for the concept “grandchild” was 孙子或孙女, a compositional phrase meaning “the son’s son or the son’s daughter” instead of 孙辈, which is a single word that precisely lexicalizes the concept.

Another factor is the existence of rare words or senses. For instance, the Kinship Database contains the Spanish word *tato*, defined as “elder brother”. However, this translation is not produced by our translation system, nor is it found in the Oxford Spanish Desk Dictionary which contains over 130,000 translations.

Finally, the Kinship Database itself unavoidably contains errors and omissions. For instance, it has no lexical entry for the concept “sibling” in Polish, for which our method correctly generates the word *rodzeństwo*. This demonstrates that our method has the capability to uncover and address the gaps in existing datasets.

Method	Template
Google Translate	[*seed word*]: [*gloss*]
ChatGPT (Seed Word)	Translate a/an [*seed word language*] sentence into [*target language*] literally focusing on the topic of kinship. Retain the ":" symbol. Provide only the translation. Each word in the final translation must be in [*target language*]. The first word before the ":" sign must be translated into the singular form. [*seed word*]: [*gloss*]
ChatGPT (Gloss)	Translate a/an [seed word language] sentence into [target language] literally focusing on the topic of kinship. Provide only the translation. Each word in the final translation must be in [target language].
ChatGPT (Baseline)	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 [concept] in [target language]? If yes, give me that word. If no, say no.

Table 4: Gloss-context templates used to obtain candidate lexicalizations.

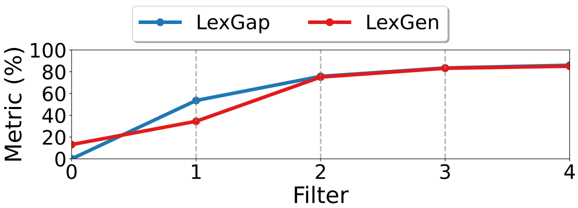


Figure 5: Average ablation results with an increasing number of filters. Metrics are the same as in Table 3.

5.5 Ablation study

We conducted an ablation study to assess the individual contributions of each filter within our method. As described in Section 4, our method starts from lexical translations, and applies four filters in sequence: 1) multi-word, 2) horizontal, 3) back-translation, and 4) vertical. We report average results across all 10 test languages. The evaluation metrics are the same as in Section 5.

Figure 5 shows a clear trend of improvement for both LexGap and LexGen following the application of each filter. Specifically, the largest boosts are provided by the multi-word filter for LexGap, and the horizontal filter for LexGen. This confirms the appropriateness of our theoretical propositions and constraints in Section 2.

6 Conclusion

We proposed a novel computational method that generates concept lexicalizations and detects lexical gaps. The method is grounded in formal definitions and propositions, and leverages translation and hypernym/hyponym taxonomy relations. We also presented an algorithm that generates both kinship concepts and the relations between them. Experimental results on diverse languages confirm the effectiveness of our approach. Our approach is general and applicable to other domains.

Limitations

While our work has made significant strides in lexicalization generation and lexical gap detection, it is not without its limitations. We expect that the increasing availability of multilingual lexical resources, and ongoing improvements in translation involving low resource language, will ameliorate these limitations over time.

Our approach relies on the accuracy and reliability of machine translation systems like Google Translate and ChatGPT. While machine translation has made significant advancements, it still presents challenges, particularly for mid and low-resource languages. The translation quality directly affects the performance of our method. Thus, our results may be less reliable for languages where high-quality machine translation is not available.

The validity of our experimental findings depends on the completeness and accuracy of the existing datasets. The datasets contain rare words or senses, which can lead to mistranslations and inaccuracies in our output. We excluded 53 concepts from our experiments because they have very few lexicalizations in the Kinship Database, and even those are in low-resource languages, such as Malayalam and Kannada, for which satisfactory automatic translation is not available. Our work underscores the utility of such datasets in diverse languages.

Acknowledgements

We thank Hadi Sheikhi for help with Persian lexicalizations, and A. Seza Doğruöz for comments on an earlier version of this paper.

This research was supported by the Natural Sciences and Engineering Research Council of Canada (NSERC), and the Alberta Machine Intelligence Institute (Amii).

References

Luisa Bentivogli and Emanuele Pianta. 2000. Looking for lexical gaps. In *Proceedings of the ninth EURALEX International Congress*, pages 8–12. Stuttgart: Universität Stuttgart.

Francis Bond and Ryan Foster. 2013. Linking and extending an Open Multilingual Wordnet. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1352–1362, Sofia, Bulgaria.

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, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Noam Chomsky and Morris Halle. 1965. Some controversial questions in phonological theory. *Journal of linguistics*, 1(2):97–138.

Jurgita Cvilikaitė. 2006. Lexical gaps: resolution by functionally complete units of translation. *Darbai ir Dienos*, 45:127–142.

Gerard De Melo and Gerhard Weikum. 2009. Towards a universal wordnet by learning from combined evidence. In *Proceedings of the 18th ACM conference on Information and knowledge management*, pages 513–522.

Rufus Hjalmar Gouws. 2002. Equivalent relations, context and cotext in bilingual dictionaries. *Hermes*, 28(1):195–209.

Lorenzo Gregori and Alessandro Panunzi. 2017. Measuring the Italian-English lexical gap for action verbs and its impact on translation. In *Proceedings of the 1st Workshop on Sense, Concept and Entity Representations and their Applications*, pages 102–109, Valencia, Spain. Association for Computational Linguistics.

Bradley Hauer and Grzegorz Kondrak. 2023a. Synonymy = Translational Equivalence. *arXiv preprint arXiv:2004.13886*.

Bradley Hauer and Grzegorz Kondrak. 2023b. Taxonomy of problems in lexical semantics. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 9833–9844, Toronto, Canada. Association for Computational Linguistics.

Vladimir Ivir. 1977. Lexical gaps: A contrastive view. *Studia Romanica et Anglica Zagrabiensia*, 43:167–176.

Maarten Janssen. 2004. Multilingual lexical databases, lexical gaps, and simullda. *International Journal of Lexicography*, 17(2):137–154.

Temuulen Khishigsuren, Gábor Bella, Khuyagbaatar Batsuren, Abed Alhakim Freihat, Nandu Chandran Nair, Amarsanaa Ganbold, Hadi Khalilia, Yamini Chandrashekar, and Fausto Giunchiglia. 2022. Using linguistic typology to enrich multilingual lexicons: the case of lexical gaps in kinship. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 2798–2807, Marseille, France. European Language Resources Association.

Oi Yee Kwong. 2018. Translation equivalence and synonymy: Preserving the synsets in cross-lingual wordnets. In *Proceedings of the 9th Global WordNet Conference (GWC 2018)*, page 201.

Adrienne Lehrer. 1974. *Semantic Fields and Lexical Structure*. American Elsevier, New York.

Federico Martelli, Luigi Procopio, Edoardo Barba, and Roberto Navigli. 2023. LexicoMatic: Automatic creation of multilingual lexical-semantic dictionaries. In *Proceedings of the 13th International Joint Conference on Natural Language Processing and the 3rd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 820–833, Nusa Dua, Bali. Association for Computational Linguistics.

George A Miller, Richard Beckwith, Christiane Fellbaum, Derek Gross, and Katherine J Miller. 1990. Introduction to WordNet: An on-line lexical database. *International journal of lexicography*, 3(4):235–244.

George Peter Murdock. 1970. Kin term patterns and their distribution. *Ethnology*, 9(2):165–208.

M. Lynne Murphy and Anu Koskela. 2010. *Key terms in semantics*. London: Continuum.

Roberto Navigli and Simone Paolo Ponzetto. 2010. BabelNet: Building a very large multilingual semantic network. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 216–225.

Roberto Navigli and Simone Paolo Ponzetto. 2012. BabelNet: The automatic construction, evaluation and application of a wide-coverage multilingual semantic network. *Artificial Intelligence*, 193:217–250.

Emanuele Pianta, Luisa Bentivogli, and Christian Girardi. 2002. Multiwordnet: developing an aligned multilingual database. In *First international conference on global WordNet*, pages 293–302.