COMPUTATIONAL ETYMOLOGY:

WORD FORMATION AND ORIGINS

by

Winston Suen Wu

A dissertation submitted to The Johns Hopkins University in conformity with the requirements for the degree of Doctor of Philosophy.

Baltimore, Maryland

February, 2022

Abstract

While there are over seven thousand languages in the world, substantial language technologies exist only for a small percentage of these. The large majority of world languages do not have enough bilingual or even monolingual data for developing technologies like machine translation using current approaches. The computational study and modeling of word origins and word formation is a key step in developing comprehensive translation dictionaries for low-resource languages. This dissertation presents novel foundational work in computational etymology, a promising field which this work is pioneering. The dissertation also includes novel models of core vocabulary, dictionary information distillation, and of the diverse linguistic processes of word formation and concept realization between languages, including compounding, derivation, sense-extension, borrowing, and historical cognate relationships, utilizing statistical and neural models trained on the unprecedented scale of thousands of languages. Collectively these are important components in tackling the grand challenges of universal translation, endangered language documentation and revitalization, and supporting technologies for speakers of thousands of underserved languages.

ABSTRACT

Primary Reader and Advisor: David Yarowsky

Secondary Readers: Kevin Duh, Philipp Koehn

Acknowledgments

First and foremost, thank you to my advisor David Yarowsky for guidance and mentorship throughout my PhD. David let me explore my own interests while guiding me to be an inquisitive, well-rounded researcher. I appreciated David's insights from his seemingly boundless knowledge of languages and cultures. David's stories of his life experiences were entertaining and enlightening at the same time, and I never minded him going off on tangents at our meetings. Thank you to my other committee members Kevin Duh and Philipp Koehn, who provided helpful comments both at my defense and the graduate board oral exam. Kevin encouraged me to always think about the important questions "why should anyone care?" and "what's next?" and was also a supportive mentor and collaborator for my qualifying project. Thank you also to my GBO members David Yarowsky, Kevin Duh, Philipp Koehn, Colin Wilson, and Paul McNamee for their helpful feedback on an earlier iteration of my work.

I am grateful to have many excellent colleagues both in and outside CLSP. David's group has always been small and close-knit, and I am fortunate to have worked closely with Arya McCarthy, Patrick Xia, Aaron Mueller, Dylan Lewis, Jamie Scharf, George

Botev, Nidhi Vyas, Chris Kirov, John Sylak-Glassman, Garrett Nicolai, and Oliver Adams. Thank you also to my many other collaborators and co-authors in CLSP: David Yarowsky, Kevin Duh, Matt Post, Garrett Nicolai, Arya McCarthy, Huda Khayrallah, Aaron Mueller, Dylan Lewis, Kenton Murray, Jacob Bremerman, Rachel Wicks, and Shuoyang Ding. I would also like to thank my collaborators and colleagues at my internship at PNNL — Svitlana Volkova, Dustin Arendt, Maria Glenski, and Prasha Shrestha — who gave me a fun short break from multilingual NLP.

I always felt at home at CLSP, and I wish to thank all my colleagues and friends at CLSP, especially those in Hackerman 226 (the best office), for the memorable interactions, enlightening discussions, and overall fun atmosphere. Thank you to Huda Khayrallah for being a wonderful cohortmate and desk neighbor, Patrick Xia for lots of feedback and camaraderie, Rebecca Knowles for helpful advice about HEART classes and leading NACLO, Zach Wood-Doughty for more feedback and animals, Pamela Shapiro for more animals, Jonathan Jones for cooking advice, Shuoyang Ding for Chinese and music discussions, Adi Renduchintala and Gaurav Kumar for non-music discussions whenever I went to visit the MT corner, Hainan Xu (who actually didn't have a desk in 226 but always seemed to be there) for being the first member of Piano Club, and Rachel Wicks and Carlos Aguirre for livening up the office. I am also thankful for the research feedback, helpful advice, and friendly conversations with Amittai Axelrod, Tongfei Chen, Jeff Craley, Seth Ebner, Matthew Francis-Landau, Dongji Gao, Ke Li, Chu-Cheng Lin, Xutai Ma, Arya McCarthy, Hongyuan Mei, Sabrina Mielke, Nanyun Peng, Adam Poliak, Matt

Post, Rachel Rudinger, Keisuke Sakaguchi, Liz Salesky, Suzanna Sia, Ben Van Durme, Tim Vieira, Dingquan Wang, Yiming Wang, Matthew Wiesner, Jinyi Yang, Sheng Zhang, Xiaohui Zhang, and certainly others.

Thank you to Ruth Scally, Yamese Diggs, Carl Pupa, and Steve Hart, who worked tirelessly to make CLSP run smoothly behind the scenes.

During my PhD, I had multiple opportunities to teach, which affirmed my love for teaching. Thank you to the HEART program for giving me the opportunity to design from scratch and teach a class of my own. Thank you to all my HEART students, NACLO students, Information Retrieval students, piano students, research mentees, and anyone else that I could share my knowledge with.

Thank you to Katrin Erk, my mentor and undergraduate thesis advisor at UT Austin who first introduced me to research and encouraged me to apply to grad school. Her enthusiasm about language has always stuck with me.

Outside of research, I am grateful for all the friends I made through my other passion, music. Specifically, a small group of musicians, cooking buddies, and chaotic conversationalists provided regular reprieve from papers and coding. Buheng Wang was the most dedicated member of Piano Club and could always lighten the mood. Audrey Chang provided great company on Great Wall trips. Alex Yu introduced me to arts and crafts. Jason Lenh taught me that high-effort dishes are not always worth it. Mathis Leblanc was always enthusiastic about Chinese and opened my eyes to some “creative” dishes. Cicero Lu somehow filled all the roles above once the old crew graduated. In addition, I was fortunate to have Qi Wang as my housemate. Thank you to Qi for the music, walks, and philosophical conversations (sometimes at the same time). Qi, who obtained his PhD a year before me, also provided a valuable perspective on the process from PhD to postdoc to professorship. I am also very thankful to Qi for encouraging me to restart teaching piano.

At Hopkins, I was fortunate to be a part of several music organizations. Thank you to Piano Club, which I led for the past 6 years. I always looked forward to the weekly meetings and was happy to be able chat and share music with such a variety of pianists. Thank you to the Hopkins Oriental Music Ensemble for allowing me to grow as an erhuist, composer, arranger, and conductor. I am especially grateful for Jason Lenh, who dumped loads of resources about traditional Chinese music onto me; Cicero Lu, who was a dedicated and enthusiastic music partner; Elvin Meng, for the free books and instruments, and Charles Agyemang, whom I had the pleasure of teaching zhonghu and daruan. Thank you to Homewood Chamber Music Seminar for all the wonderful moments making music. As one of the longest-standing members, I had the chance to perform with many excellent musicians in a variety of small chamber groups: Julia Costacurta, Lucinda Chiu, Brian Chu, Sara Earnest, Emma Elias, Jordan Elum, Susan Kim, Teagan Kim, Kevin Li, Ben Lee, Angelique Ruml, Andrew Shiau, Celina Shih, Sydnee Wong, and Jinghang Zhang. Thank you especially to Emma, Sydney, and Brian, who were willing to learn and premiere my own composition. Thank you also to my chamber coaches Audrey Wright, Jacob Shack, and Lior Willinger for their insightful comments.

ACKNOWLEDGMENTS

Thank you to my piano teachers Chad Bowles and Marian Hahn at the Peabody Conservatory. In a way, piano teachers are like PhD advisors: experts in their field and long-term mentors, just without any expectations to publish. Chad and Marian encouraged me to always improve every week, not just as a musician, but also as a person.

Besides attending conferences, I did not do much travelling during my PhD, but when I did, I was fortunate to have friends along the way. Thank you to Buheng Wang who traveled with me across China, Cicero Lu who survived a roadtrip with me, and my old friends Jerry So and Jason Cheng who hosted me in Chicago and Seattle, respectively. Thank you also to my old piano teacher in Austin, Sophia Gilmson, for the lesson and wonderful afternoon of conversations.

Finally, I am grateful for my girlfriend Cicero Lu, who is such a thoughtful and supportive adventure partner. I am also very thankful to my father Honglu, my mother Lisa, my sister Natalie, and my extended family for all their support throughout my life. Thank you especially to Aunt Vivian and Uncle Pete, who often invited me over for dinner when I did not go home for the holidays. Finally, thank you to Tigger and Bean for sparking joy whenever I see them.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

The world has over 7,000 languages, and the top 20 languages are spoken by 50% of the world's population.[1] These top 20 languages are shown in Table 1.1 and include those which are typically called *high-resource languages*, i.e. languages that have existing language technologies and sufficient data for training them.

One such technology is machine translation (MT). Originating in the 1940s, the notion of using computers to perform translation has had far-reaching impact, enabling communication between speakers of different languages and helping to build a more interconnected world. In the present day, commercial machine translation tools are available for many languages and easily accessible at the click of a button. As of December 2021, Google Translate[2] exists for 180 languages, Microsoft Translator[3] supports 103 languages,

[1] https://en.wikipedia.org/wiki/List_of_languages_by_number_of_native_speakers

[2] https://translate.google.com

[3] https://www.bing.com/translator

Rank	Language	Speakers (millions)	% of World Population
1	Mandarin Chinese	918	11.922%
2	Spanish	480	5.994%
3	English	379	4.922%
4	Hindi	341	4.429%
5	Bengali	300	4.000%
6	Portuguese	221	2.870%
7	Russian	154	2.000%
8	Japanese	128	1.662%
9	Western Punjabi	92.7	1.204%
10	Marathi	83.1	1.079%
11	Telugu	82.0	1.065%
12	Wu Chinese	81.4	1.057%
13	Turkish	79.4	1.031%
14	Korean	77.3	1.004%
15	French	77.2	1.003%
16	German	76.1	0.988%
17	Vietnamese	76.0	0.987%
18	Tamil	75.0	0.974%
19	Yue Chinese	73.1	0.949%
20	Urdu	68.6	0.891%

Table 1.1: The top 20 languages by number of native speakers. Reproduced from Wikipedia.

and DeepL[4] supports 28 languages.

Other major language technologies also exist at this (limited) scale of language coverage. Universal Dependencies (Nivre, Marneffe, Ginter, Y. Goldberg, et al., 2016; Nivre, Marneffe, Ginter, Hajič, et al., 2020), used for developing parsers, is available for 122 languages. Automatic speech recognition is available from Google for 137 languages.[5] While these technologies are available for many of the major languages in the world, *they fail to account for the other roughly 6,900 languages spoken by the other half of the world's population.*

Suppose that a disaster occurs somewhere in the world. Perhaps this is an earthquake, a disease outbreak, or some other phenomenon. The inhabitants of the affected area do not use a major language for which we have translation capabilities. Thus, any communication, including news, TV, radio, and social media, is unintelligible. The global community is trying to figure out what is happening. Where exactly is it? Who is affected? Who needs help? How urgent is the situation?

This is the scenario envisioned by the grant program that funded much of my PhD work. The mission of the DARPA Low Resource Languages for Emergent Incidents (LORELEI) program was to develop technology to help disaster responders quickly achieve understanding of a local language. The problem is that these low-resource languages have poor-quality or no existing machine translation systems, and little to no readily available data for training said systems. The program participants were tasked to develop effective

[4] https://www.deepl.com

[5] https://cloud.google.com/speech-to-text/docs/languages

machine translation technology (among others) in the face of such data scarcity.

Machine translation systems are typically trained on sentence pairs (bitext) where one sentence is a translation of the other. Large collections of bitext are called parallel corpora. These corpora are likely to exist for high-resource languages, but not for low-resource languages. Since high-resource and low-resource are not precise terms, I loosely group languages into several classes to clarify what is meant when talking about the quantity of available resources.

Class 1 languages are the top 30 or so languages in the world in terms of available resources. These languages have extensive existing corpora on which to train MT systems. One source of parallel sentences is the European Parliament proceedings, which is translated into 24 languages. These are typically called high-resource languages.

Class 2 languages are languages ranked around 30–200, which may have existing parallel corpora (which might be mined from the web using Bañón et al. (e.g. 2020), which supports under 50 languages), existing monolingual corpora (which might be mined from the web using Common Crawl J. R. Smith et al. (2013, e.g.), which supports 160 languages) and decent sized dictionaries. At this resource level, one can apply unsupervised machine translation techniques such as cross-lingual embeddings (e.g. Ravi and Knight, 2011; Artetxe, Labaka, and Agirre, 2019; Marchisio, Duh, and Koehn, 2020; Marchisio, Koehn, and Xiong, 2021) or other methods (e.g. Schafer and Yarowsky, 2002) to obtain lexical translations without parallel corpora. These languages are typically considered medium- to low-resource.

Class 3 languages are language ranked around 200–1600. These languages do not have any significant bilingual corpus except for the Bible (McCarthy, Wicks, et al., 2020), the most translated document in the world. Another widely translated text, though substantially smaller than the Bible, is the Universal Declaration of Human Rights, available in 530 languages.[6] These corpora also act as monolingual text in that language. At this level of resourceness, languages are unlikely to have much of a web presence, and even if text is available, there do not exist adequate tools for identifying these languages. One can apply cross-lingual embedding methods on the Bible, but as the Bible is a text in a specialized domain, these methods miss large chunks of the world's concepts and thus are not applicable for general vocabulary. However, the methods I describe in this dissertation can successfully predict missing translations for out-of-Bible vocabulary. These languages are low-resource languages.

Class 4 languages are languages ranked 1600+. There are simply no monolingual corpora available. At best, these languages may have a dictionary on the order of 100–1000 words, which might be manually constructed by a field linguist or a native informant at the first contact with this language. These languages are very-low resource, or may not have any resources at all. At the higher end of this range, the methods in this dissertation are still applicable. At the lower end of this range, any method for dictionary induction is essentially guessing.

The work presented in this dissertation aims at class 3 (and to some extent, class 4)

[6] https://www.ohchr.org/en/udhr/pages/introduction.aspx

languages above in tackling the task of *massively multilingual dictionary induction*: fill in missing entries in a translation dictionary. Leveraging signal from related languages as well as from all the languages in the world for which there is an available dictionary, I develop computational models of multiple linguistic processes of word formation on an unprecedented scale in order to induce missing entries in a low-resource language's dictionary. Below is an example of these linguistic processes.

To illustrate the motivation for tackling dictionary induction from the angle of word formation, consider the concept WATERMELON.[7] The English word *watermelon* originated as a compound of the English words *water* and *melon*. Below are several languages' word for WATERMELON, which can be roughly grouped into categories, as presented in Figure 1.1. In the remainder of this dissertation, I use the three-letter ISO 639-3 language codes to indicate a word's language.

As seen in Figure 1.1, realizations of WATERMELON follow several linguistic processes. The first group contains compound word that are literal translations of WATER+MELON in their respective languages and thus are calques (loan translations) from English (e.g. the Danish *vandmelon* 'water' + 'melon'), the language in which the composition of the concepts of WATER+MELON was first observed. The second group contains translations that are combinations of WATER+MELON, but are also cognate[8] with English, because these are Germanic languages that are related to English. A third category of translations contains compound words that are not composed of WATER+MELON (e.g. 'west melon' in

[7] I denote a semantic concept in SMALL CAPS, which is distinct from the realization of the concept in a specific language, which may be in regular type or italic.

[8] Cognates are words that have a shared etymological origin

Compounds of WATER+MELON

Lang	Word
cze	vodní meloun
dan	vandmelon
epo	akvomelono
fin	vesimeloni
ido	aquomeloniero

Compounds of WATER+MELON, also cognate with English *watermelon*

Lang	Word
afr	waatlemoen
deu	Wassermelone
ltz	Waassermeloun
nld	watermeloen
srn	watramun
swe	vattenmelon

Compounds that are not WATER+MELON

Lang	Word	Literal translation
zho	西瓜	west melon
hun	görögdinnye	Greek melon
ron	pepene verde	green melon

Other realizations

Lang	Word	Literal translation
spa	sandía	Sindhi (origin location)
glg	sandía	Sindhi (origin location)
sdn	síndriadan	Sindhi (origin location)
mkd	бóстан (bostan)	garden (Persian borrowing)
alb	bostan	garden (Persian borrowing)
kaz	қарбыз (karbiz)	honeydew (co-hyponym)
ita	cocomero	cucumber (remote co-hyponym)
ron	pepene	melon (hypernym)
rup	peapini	melon (hypernym)

Figure 1.1: Translations of the concept of WATERMELON in various languages, following various linguistic processes.

Chinese, or 'Greek melon' in Hungarian), in these cases referring to the watermelon's ascribed origin. Finally, a fourth category of translations contains words that can roughly be translated in their respective language as words related to WATERMELON, such as its semantic hypernym ('melon'), sibling co-hyponym ('honeydew'), more distantly related co-hyponym ('cucumber') or its ascribed origin (e.g. 'garden' or 'Sindhi', a region in Pakistan where presumably Watermelons were sourced).

We see that across languages, there are many ways to express the concept WATERMELON, but they follow regular processes that can be computationally modeled. I discuss compositional word formation in Chapter 4, cognate relationships in Chapter 5, and related words in Section 4.2. These chapters make up the bulk of this dissertation on computational word formation.

Word formation falls under the larger umbrella of etymology, the study of the origin of words. My work is one of the first to thoroughly study word etymology using computational means. Thus, I call this field of study *computational etymology*. The first usage of this term seems to be in Yang (2004), but he restricts his study to cognates. I define computational etymology more broadly: computational etymology is the computational study of the etymology of words, which includes word formation, the origins of words, and how words and their meanings change. In this dissertation, I seek to answer questions such as:

- What language did this word come from?
- How did it enter its current language?

- When did it enter its current language?
- What might this word look like in another language?

The study of etymology has historical interest. Since antiquity, philologists have been interested in the origins of and relationships between languages, and their studies have given rise to the modern fields of comparative and historical linguistics. Lexicographers and linguists with specialized knowledge of multiple languages have painstakingly compiled dictionaries containing (some of) the answers to these questions. In modern times, large crowdsourcing efforts have allowed the general public to contribute to multilingual dictionaries such as Wiktionary,[9] which also acts as a central repository for storing and disseminating the information resulting from numerous linguists' efforts at documenting languages around the world.

Yet, dictionaries like Wiktionary follow the classic Zipf's law in terms of coverage across languages (see Table 1.2). As of December 2021, Wiktionary contains entries in 4,278 languages,[10] but only 208 of these languages have over 1,000 definitions. Only 55 of these languages contain over 10,000 definitions; these are high-resource languages. Crucially, almost 3,000 very-low-resource languages have fewer than 100 definitions, indicating that there is still much work needed to develop a comprehensive multilingual dictionary.

[9] wiktionary.org

[10] Recall that there are around 7,000 languages in the world. Wiktionary recognizes 8,155 language codes, but some of these languages are extinct. Source: https://en.wiktionary.org/wiki/Wiktionary:List_of_languages

55	languages with 10000 or more definitions
153	languages with 1000 to 9999 definitions
439	languages with 100 to 999 definitions
795	languages with 10 to 99 definitions
1364	languages with 2 to 9 definitions
1470	languages with a single definition

Table 1.2: Statistics of language coverage in Wiktionary. Reproduced from Wiktionary.

Modeling the etymology of words computationally has many benefits. For lexicography, philology, and historical linguistics, the results of computational models of etymology can help researchers in this field construct new etymologies and verify existing ones. Practically, successfully answering questions in computational etymology enables the construction of a fully comprehensive multilingual dictionary. This comprehensive dictionary will enable users from around the world to communicate across language boundaries, which is important for business and social interactions. Comprehensive dictionaries are important components in machine translation systems when existing bitext is not available for low-resource languages. Even if bitext is available, machine translation systems frequently encounter out-of-vocabulary (OOV) words that are not seen during training. The methods I describe in the following chapters on computational word formation can propose candidate translations for unknown words, which can be used to augment existing machine translation systems. My methods are massively multilingual, leveraging the combined resources of many other (potentially higher-resource) languages. And they are also automatic, alleviating the need for native speakers or linguists with specialized knowledge.

Besides applications to machine translation, a comprehensive multilingual dictionary provides a platform for language documentation and revitalization, which will help underserved language communities better participate in the global economy. Such a dictionary will enable broader universal access to knowledge that is locked within a single language. It will also be a valuable resource for language learning, serving as the base for language learning software for thousands of languages. With contributions from both computational models and humans, these dictionaries may also reveal unknown connections between languages, allowing researchers to create more accurate linguistic phylogenies and better understand how languages interacted across time.

Below, I briefly introduce the major sections of this dissertation and how they fit into the overall goals of computational etymology.

Chapter 3: Comprehensive Dictionary Construction

In our current age, we are fortunate to have online lexical resources readily available at our fingertips. However, these resources vary greatly in types of information contained within, as well as in their coverage of the world's languages. In this chapter, I utilize PanLex (Baldwin, Pool, and Colowick, 2010; Kamholz, Pool, and Colowick, 2014) and Wiktionary (`wiktionary.org`), two of the largest multilingual dictionaries available online. PanLex's goal is to be the world's largest database of lexical translations. It is notable for having high coverage (5,700+ languages), but only contains lemma translations. On the other hand, Wiktionary is a large (4,200+ languages), multilingual dictionary freely editable by the community. In addition to information contained in a traditional paper dictionary (lemma, pronunciation, part of speech), Wiktionary contains a wealth of other information, including a word's etymology, translations, morphology, semantic relations, even anagrams. However, the data in Wiktionary is in a semi-structured Markdown-like form that is not easily usable by computer systems.

I also present Yawipa[11], a new framework for developing Wiktionary parsers. Using Yawipa, I developed comprehensive Wiktionary parsers that extract and normalize the data contained in Wiktionary into a form that can be easily processed by downstream applications. These parsers improve over several existing parsers in terms of scope and types of information extracted and facilitate the research in computational etymology contained in this dissertation.

Chapter 3: Core Vocabulary

Though Wiktionary and PanLex are the most comprehensive currently existing multilingual dictionaries, they suffer from a severe lack of coverage for low-resource languages. When documenting languages, field linguist are limited by time and must consider which words to obtain elicitations for. Similarly, for dictionary induction, I would like to prioritize words with high impact for the community to quickly allow communication with major languages. To this end, I propose a new functional definition and construction method for core vocabulary sets based on the relative coverage of a target concept in thousands of bilingual dictionaries. My newly developed core concept vocabulary list derived from these dictionary consensus methods achieves high overlap with existing widely utilized

[11] github.com/wswu/yawipa

core vocabulary lists targeted at applications such as first and second language learning or field linguistics. My in-depth analysis illustrates multiple desirable properties of my newly proposed core vocabulary set, including their non-compositionality. I argue that this core vocabulary should be prioritized for elicitation when creating new dictionaries for low-resource languages for multiple downstream tasks including machine translation and language learning. Thus, I use this core vocabulary set as the basis for evaluating my models of word formation.

Chapter 4: Compositional Word Formation

The bulk of this dissertation deals with word formation, i.e. how words are created. Since the word *word* is polysemous, in this chapter I will use *word* to refer to a lexeme. Thus, I am specifically interested in *lexeme formation* within a language, i.e. the formation of a unit of lexical meaning from existing linguistic units in that language. Complex words are formed compositionally through various linguistic processes. For example,

- **Compound words**, such as *lighthouse* and *dental*, are made up of the combination of multiple morphemes, which could be free (*light* + *house*) or bound (*cran-* + *-berry*).

- Words formed via **derivational morphology**, such as *drinkable* or *runner*, contain a morpheme whose inclusion typically modifies the original word's part of speech but may indicate a regular semantic extension within the same part of speech (e.g. *unhappy*).

- **Multiword expressions** such as *fire truck* or (in French) *pomme de terre* 'potato'

> are similar to monolexemic compound words, but composed of multiple separate words, although often with constrained syntactic behavior.

Compounding is sometimes considered a language universal (Fromkin, Rodman, and Hyams, 2018), and there are many documented mechanisms for forming compound words across the world's languages. The simplest is directly concatenating two words. Many languages have a linking element that connects the two parts, e.g. German *Liebesleid* = *Liebe* 'love' + *s* + *leid* 'song'. This linking element, also called a filler (Koehn and Knight, 2003) or glue (Garera and Yarowsky, 2008) in the compositional literature, may be inserted to ensure the compound conforms to the phonotactics of the language. It may also be an inflection marker on the first word (e.g. *Jahreszeit*, literally 'year'-'time' = 'season', with the genitive case *Jahres* of *Jahr*='year'), or a separate particle, e.g. French *pomme de terre* = *pomme* 'apple' + *de* 'of' + *terre* 'earth'. The component parts of the compound may take a variety of forms, including being a stem (German *Trinkwasser*), an infinitive (Danish *Drikkevand*), or a participle (English *drinking water*).

In this chapter, I develop a universal model of word compounding that can successfully translate compound words from a foreign language into English, as well as generate translation candidates from English into other languages. I adopt a loose definition of "compound word" as any word or a sequence of words that can be decomposed into meaningful subwords, where the subwords may be words or morphemes like derivational affixes. Thus, this definition includes both complex words and phrases. My compounding model uses the combined data from hundreds of languages in Wiktionary, an order of magnitude larger than previous work (Garera and Yarowsky, 2008), and handles many of the world's languages' mechanisms for compounding, including concatenation with epenthesis and elision. This model has important applications for low-resource translation, especially in specialized domains such as science and medicine where compound words are abundant.

Chapter 4: Lexical Relations

This chapter also presents a translation method that bridges through lexically related words: synonyms, hypernyms, hyponyms, and co-hyponyms. For example, the word for WATERMELON in a language is often the same as its hypernym MELON, because a specialized word for WATERMELON simply does not exist in the target language's lexicon. Additionally, WATERMELON is sometimes translated via sense extension as a related co-hypernym (e.g. *honeydew melon* which may be more commonly known in the language, or more unusually as a rather distant but similarly-colored oval-shaped co-hyponym such as CUCUMBER. I model the likelihood of related words being acceptable translations of unknown words, and I show that this model, which does not require any neural component, is simple and effective, especially for low-resource languages.

Chapter 5: Cognate/Sound-Shift Models

Almost all languages are genetically related to other living or attested languages, and these relationships can be seen in their words. For example, the Italian *cavallo* and French *cheval* both originate from the Latin *caballus*, all of which mean *horse*. These cognates, from the Latin *cognatus* ‘related by blood’, are words that share a common etymological origin, and exhibit similar properties, namely that they have similar phonology, orthography, and semantics. This chapter is interested in word formation from related languages, specifically a class of etymological relations involving sound shifts, including cognates, inheritance, borrowing, and transliteration.

In this chapter, I investigate models of cognate and sound-shift word formation in the task of dictionary induction. This work is motivated by the tremendous capacity for humans to generalize during translation, producing forms for words that have not been seen before. This becomes valuable especially for lower-frequency words, which may not have been observed in training data but could be inferred through regular processes such as cognate relationships with related languages. Specifically, I treat the modeling of cognate and sound-shift mechanisms as a sequence transduction problem, using a pragmatic definition of cognacy based on orthographic or phonetic similarity across languages (Kondrak, 2001), which has been adopted by a number of computational cognate research (e.g. Inkpen, O. Frunza, and Kondrak, 2005; Ciobanu and Dinu, 2014; Wu and Yarowsky, 2018b).

Because large-scale aligned cognate lexicons are not readily available for all but the highest-resource of languages, I devise an algorithm to automatically discover cognates by clustering translations from existing multilingual dictionaries. I also develop a notion of weighted edit distance to better capture similarities between cognate words. Finally, using cognate clusters as multiway aligned bitext, I train sequence-to-sequence models for the task of cognate generation on a combination of languages, language families, and word formation mechanisms, showing the success of such models in ensemble and multilingual scenarios.

Chapter 6: Machine Learning for Computational Etymology

Since antiquity, scholars have been fascinated by etymology, the study of words' origins. In modern days, there exist numerous etymological dictionaries for select languages (e.g. English (Partridge, 2006), Albanian (Orel, 1998), or Old Chinese (Schuessler, 2007)) as well as language families (e.g. Italic (De Vaan, 2018), Slavic (Derksen, 2007), or Altaic (Starostin et al., 2003)). Many of these improve and expand upon existing dictionaries as new evidence comes to light about the relationships between languages and their words. However, until very recently, the discovery of these relationships has not been computational driven.

In an era of abundant linguistic data, I seek to address the dearth of computational approaches to modeling etymology. To this end, using etymology data I extracted from Wiktionary using Yawipa, I present several approaches to model from where, how, and when a word enters a language. I employ neural classification models as well as modern neural sequence-to-sequence models to accurately predict a word's formation mechanism, parent language, and year of emergence. For predicting the era of word formation, I also experiment with various data-driven models based on historical word usage. These methods are language-independent and are applicable for improving existing etymology determinations that may be incorrect, as well as providing etymology for words that may not have an existing etymological entry, both in low- and high-resource languages.

Chapter 7: Combined Methods for Unknown Word Generation

In this final chapter, I employ the models for word formation described in this dissertation, namely the cognate, compositional, and lexical relation models, to generate translations of words into target foreign languages. Even though the target language may only possess a small dictionary, I show that these models can effectively predict words in the target language by leveraging information from many other languages. The evaluation is performed on several languages ranging from medium- to low-resource and on a set of concepts spanning the range of coreness, showing the efficacy model combination.

Chapter 8: Conclusion

This chapter summarizes the scientific contributions of this dissertation and proposes avenues of future work, including a large-scale crowdsourcing platform for language documentation and revitalization.

This dissertation contains work published in Wu and Yarowsky (2018c), Wu and Yarowsky (2018b), Wu and Yarowsky (2020b), Wu, Nicolai, and Yarowsky (2020), Wu and Yarowsky (2020a), Wu, Duh, and Yarowsky (2021), and Wu and Yarowsky (2021).

Chapter 2

Prior Work

This section surveys the existing literature relevant to each of the following chapters of this dissertation.

2.1 Comprehensive Dictionary Construction

Perhaps the largest and most prominent effort to build a comprehensive multilingual dictionary is Wiktionary. Though Wiktionary has existed since 2002, only within the last several years has there been a great surge of interest in using the data in Wiktionary for natural language processing tasks. Navarro et al. (2009) was one of the first to examine Wiktionary as a resource for NLP. Since the data in Wiktionary is not readily usable, many researchers as well as hobbyists have developed parsers for Wiktionary. In comparison to my parser Yawipa, these other existing Wiktionary parsing efforts have different goals and scope. Yawipa's goal is to be comprehensive and extensible. To that end, Yawipa goes beyond existing parsers in extracting and normalizing information, such as etymology and translations, that are not encoded in structured Wiktionary markup (and thus easy to parse). Technically, Yawipa is not just a parser, but a parsing framework that facilitates the creation of new parsers for other Wiktionary editions.

In terms of comprehensive extraction from Wiktionary, there are a few similar projects. knoWitiary (Nastase and Strapparava, 2015) extracts data from Wiktionary with the intent of comparing its coverage to that of WordNet. DBnary (Sérasset, 2015) extracts lexical information into a structured database format. ENGLAWI (Sajous, Calderone, and Hathout, 2020) extracts Wiktionary data into XML.

Translations are an important part of my work, and I have made substantial efforts to extract translations from Wiktionary that are not explicitly labeled as such. Most studies on translation extraction have utilized the translation section of an entry: Ács (2014) using a triangulation approach, Kirov, Sylak-Glassman, et al. (2016) for morphological analysis. Perhaps most similar to my work is DBnary Sérasset (2015), which parses certain lexical data, including translations, from Wiktionary and converts it into a structured format.

Yawipa also extracts morphological relations between words. Other projects that parse this type of information include UniMorph (Kirov, Sylak-Glassman, et al., 2016; Kirov, Cotterell, et al., 2018; McCarthy, Kirov, et al., 2020), a large-scale effort to compile a broad-coverage resource of morphological paradigms of nouns, adjectives, and verbs in 118 languages extracted from Wiktionary. Other large-scale parsing efforts for targeted tasks include NULEX (McFate and Forbus, 2011) for parsing, IWNLP (Liebeck and Conrad, 2015) for lemmatization, and WikiPron (Lee et al., 2020) for pronunciations.

Regarding parsing etymology, there are a few existing efforts to parse etymological information from Wiktionary at different granularities. Etymological WordNet (Melo, 2014) contains coarse-grained relations between pairs of words. The relations include is-derived-from, has-derived-form, etymologically-related, etymological-origin-of, etymology, and variant:orthography. This data covers 2.8 million terms. EtymDB (Sagot, 2017; Fourrier and Sagot, 2020) extracted more fine-grained relations including borrowing, compound, cognate, derived, derived-prefix, derived-suffix, and inherited. Both of these projects do not make use of the full range of etymological relationships present in Wiktionary. Thus, there is strong motivation to develop my own Wiktionary parser that is both comprehensive and extensible: it can extract the etymological information and many other types of information annotated in Wiktionary, and it is easy to use and extend for further research.

2.1.1 Core Vocabulary

A word's coreness is an important criterion for dictionary elicitation. Probably the most well-known formulation of a core vocabulary is the Swadesh list (Swadesh, 1952; Swadesh, 1955). This set of concepts, created by linguist Morris Swadesh, originally contains 215 concepts. Swadesh pruned his list to 200 words in 1955, and then a 100-word list was published posthumously in 1971. This list of basic words is used in historical comparative linguistics to determine the relationships between languages, and there have been many attempts to revise or expand these concept lists for this purpose. Rather than enumerating hundreds of these lists here, I refer the reader to Concepticon[1] List, Cysouw, and Forkel (2016), a recent effort to compile such existing lists. It currently contains 392 concept lists.

2.1.2 Dictionary Induction

One major goal of my work is the induction of missing entries in a multilingual dictionary, which can be thought of as a translation matrix. The notion of translation matrices, or concept-aligned words across the world's languages, has a long line of research. Back in the 1950s, Morris Swadesh compiled a list of concepts (Swadesh, 1952; Swadesh, 1955) which he believed were culturally universal for the purposes of establishing relationships between languages (Swadesh, 2017; Dyen, Kruskal, and Black, 1992). Since then, the availability of larger online lexicons have led to more recent studies focused on creating multilingual aligned resources from Wiktionaries and WordNets (e.g. Kazakov and Shahid, 2009; Nastase, Strube, et al., 2010; Bond and R. Foster, 2013).

The task of translation matrix completion, the filling-out of a universal conceptual inventory, has been approached by three broad classes of methods. The first is to manually construct concept inventories, as in (Swadesh, 1952) and followup work. This is unsurprisingly laborious and requires human effort. The second is to automatically identify

[1] https://concepticon.clld.org

cognate relationships. The third is to generate putative cognates by performing transduction in the form of sound or orthographic shifts. See Section 2.3 for related work for the latter two points.

2.2 Compositional Word Formation

The first major word formation mechanism I investigate is compositional word formation. This type of word formation includes complex words, which may be formed via compounding, which has a rich linguistic literature, as well as inflectional and derivational morphology. For a broad survey of linguistic theories of compounding, I refer the reader to Lieber and Stekauer (2011). Following Bauer (2009), I briefly survey the typology of compounds,[2] focusing on aspects relevant to my work.

There are many linguistic and cognitive theories about how humans form compounds. One prominent theory is Construction Grammar, (Fillmore, 1988) which posits that *constructions*, or learned pairings of linguistic patterns with meanings, are the fundamental building blocks of human language. As stated in A. E. Goldberg (2006):

> Any linguistic pattern is recognized as a construction as long as some aspect of its form or function is not strictly predictable from its component parts or from other constructions recognized to exist. In addition, patterns are stored as constructions even if they are fully predictable as long as they occur with sufficient frequency.

In the framework of Construction Grammar, the building blocks of compound words, whether they are words or morphemes, can be viewed as constructions (Booij, 2009).

[2] Bauer (2009) concludes that it is problematic to come up with a definite typology of compounds.

Compounds are often classified semantically into one of three categories, loosely translatable with the formula in quotations:

- subordinate “B-of-A”: *truck driver*, *table leg*
- attributive “B-for-A”: *file cabinet*, *lighthouse*
- coordinate “A-and-B”: *blue-green*, *singer-songwriter*

A compound’s meaning spans a range of predictability, from compositional to idiomatic (Kavka, 2009). For example, the following compounds are increasingly idiomatic and unpredictable.

- *red ink* ‘financial loss’
- *red carpet* ‘celebrity’
- *blue blood* ‘aristocrat’

In addition, some studies show that humans cannot accurately predict the meaning of a compound word from the meaning of its components alone (Štekauer, 2009; Gagné, Marchak, and Spalding, 2010). I show computationally that this is possible to an extent.

2.2.1 Compounds in Natural Language Processing

In NLP, compounds have garnered much interest over the years, with several workshops have been dedicated to compound analysis (Verhoeven et al., 2014) and multiword expressions (Cook et al., 2021). Compound splitting is the predominant task in compound processing, in which a system must identify the component parts of the compound word. One popular approach is to split the word into all possible subwords and rank the resulting splits based on the subwords' frequency in a corpus (e.g. Grefenstette, 1999; Koehn and Knight, 2003). This is a simple but effective approach, which I follow in my work.

However, rather than in splitting compounds, my interests lie more in translating and predicting them. There is a small thread of existing work in this regard. One of the first studies was Rackow, Dagan, and Schwall (1992), who translated German noun-noun compounds into English by individually translating the component parts using a bilingual dictionary and ranking translations using corpus frequency. Grefenstette (1999) performed a similar task with German and Spanish compounds, using frequency in Web corpora, and Tanaka and Baldwin (2003) do the same for Japanese noun-noun compounds into to English. Bungum and Oepen (2009) extend Tanaka and Baldwin (2003)'s approach for Norwegian to English. More recently, a shared task was held on producing paraphrases for English noun compounds (Hendrickx et al., 2013).

These studies, as well as most studies in the linguistics literature, focus on a single language pair, or a handful of languages. Garera and Yarowsky (2008) was one of the first to analyze compounds on a large scale, using a bilingual dictionary of 50 languages. They predict translations of a compound word using the following procedure:

1. Split the compound word into two concatenated parts, accounting for an intermediate "glue" character.

2. Separately translate each component part using a bilingual dictionary, obtaining a literal English gloss of the entire compound word.

3. Look up words in other languages that have the same English glosses.

4. Compute a distribution over the English translation of these other words.

Garera and Yarowsky (2008) call their approach *multipath gloss translation*, because the English translation can be obtained by traveling through words in several other languages. My approach is similar in that I use multiple bilingual dictionaries, but I study and model the compounding phenomenon in more depth as well as on an order of magnitude larger scale (hundreds of languages), with the significant benefits of more reinforcement between unrelated languages. In addition, I perform experiments on compound generation into a foreign language, not covered in their work.

In terms of generating compound words, one line of work (Stymne and Cancedda, 2011; Stymne, Cancedda, and Ahrenberg, 2013) focuses on phrase-based machine translation. In an English to German translation task, they train their model with the target side (German) compound words split. At test time, they use a variety of heuristics to merge words into compound words. Matthews et al. (2016) perform a similar task with two systems: a neural classifier to determine which words should be merged, and a word-to-character phrase-based decoder to generate the merged compound word. My work, targeted at low-resource languages, forgoes these computationally intensive methods which require large amounts of training data. In contrast, my compound generation process generates translations of the component parts using a probabilistic model of component translation, flipped ordering, and linking characters between components, learned from the combination of compounds in hundreds of languages.

Another effort at compiling a multilingual resource of compound words is MorBoComp (Guevara et al., 2006). This project claims to contain a database of word compounds in 20 languages, but the project seems to have stalled, and I was unable to access the data mentioned in their work. My work encompasses a much larger set of languages (by a factor of 15x) and a much larger set of derived instances, and posits compound generation and analysis models absent from their work.

In terms of applications, handling compound words well has been shown to improve machine translation, e.g. into English (Koehn and Knight, 2003) and German (Stymne, Cancedda, and Ahrenberg, 2013) and has helped simplify medical text (Abrahamsson et al., 2014). I expect that my large scale publicly distributed compound-based translation dictionaries and associated generative and analytic models will be useful for out-of-vocabulary handling in downstream machine translation systems, especially for low-resource languages.

2.2.2 Translation via Lexical Relations

I propose another avenue for translating words by going through via lexical relations, such as synonymy and hypernomy. WordNet (Fellbaum, 2010) is a well-known source for synonyms, and using synonyms is a natural choice in machine translation. Even back in the 1990s, researchers investigated whether synonyms can replace in machine translation (Collier, Hirakawa, and Kumano, 1998). Recently, some have shown that synonyms are useful in low-resource MT of Vietnamese (Ngo et al., 2019). Some MT evaluation metrics also use synonyms as part of the metric (e.g. Banerjee and Lavie, 2005; C. Liu, Dahlmeier, and Ng, 2010; He et al., 2010). Andrade et al. (2013) use synonyms to find translations in comparable corpora.

However, translation via other relations is possible and has not been sufficiently investigated. For example, the concept of WATERMELON can be translated in Serbo-Croatian as 'melon' (a hypernym) and in Italian as 'cucumber' (a rather distant co-hyponym). Translation via lexical relations are usually studied in the context of constructing multilingual WordNets (e.g. Huang, Tseng, and Tsai, 2002; Huang, Su, et al., 2005; Nien et al., 2009), where researchers translate the English WordNet in order to bootstrap the construction of a new WordNet in their target language. My work investigates the acceptability of a word's translation in a low-resource language based on lexically related concepts across languages.

2.3 Cognate and Sound-Shift Models

Another major word formation process is cognate/sound-shifting, which accounts for many etymological relations including inheritance, borrowing, and transliteration. Cognate models have been extensively employed to recover missing dictionary translations.

For example, Mann and Yarowsky (2001) generate cognates by a pipeline of dictionary lookup and probabilistic orthographic shifts. Mulloni (2007) uses an SVM-based tagger to label the cognate character sequence for cognate generation. Ciobanu (2016) uses a CRF with reranking to the same end. Beinborn, Zesch, and Gurevych (2013) perform translation matrix completion with extracted cognate lists using character-level statistical machine translation systems trained on separate source-target language pairs. Scherrer and Sagot (2014) perform a task similar to my own; they start with a word list and find plausible cognates using the BI-SIM metric (Kondrak and Dorr, 2004), originally designed for identifying drug names, then perform character-based machine translation on cognates. They experiment with translating cognates from a high-resource language to a low-resource language. My work differs in that my experiments are on a much larger scale, and realize improvements by combining the results of multiple machine translation systems.

This dissertation applies multilingual cognate models to predict related forms of words. Similar approaches have also been applied to the task of proto-language reconstruction (Meloni, Ravfogel, and Y. Goldberg, 2021). Related to cognate prediction is the task of *grapheme-to-phoneme conversion*, which also has a long history of research. Cognate transliteration can be viewed as G2P across languages, where the words are cognates, for example, names (Waxmonsky and Reddy, 2012; Wu, Vyas, and Yarowsky, 2018; Wu and Yarowsky, 2018a). Recently, researchers have studied massively multilingual versions of these tasks, where single (neural) models are trained on the combination of hundreds of languages (e.g. Deri and Knight, 2016; Gorman et al., 2020; Lewis et al., 2020).

One issue with many of these cognate/sound-shift models is that there is little or no cognate data available for training. Thus, researchers have developed methods to automatically identify cognate relationships, sometimes called *cognate detection*. One of the seminal works in this area is Brew, McKelvie, et al. (1996), who investigate the Levenstein edit distance (Levenshtein et al., 1966) and Dice's coefficient to extract "lexicographically interesting word pairs" (i.e. cognates) from aligned bitext. Many others have proposed improvements on the surface level of cognates, including Longest Common Subsequence Ratio (Melamed, 1999), matching at least four consecutive characters or containing digits (Simard, G. F. Foster, and Isabelle, 1992), phonetic features (treating the word as a phonetic sequence) (Kondrak, 2000), semantic features via WordNet (Kondrak, 2001), and n-gram features (Kondrak, 2005). Many of these above features have been incorporated into machine learning approaches for cognate detection, including hidden Markov models (Mackay and Kondrak, 2005; Kondrak and Sherif, 2006), support vector machines (Bergsma and Kondrak, 2007; Rama, 2015), and other various off-the-shelf machine learning algorithms (O. M. Frunza, 2006). I develop a simple and effective multiple-iteration weighted edit distance approach for discovering cognates. Perhaps most similar to my work is Hauer and Kondrak (2011), who also cluster cognates based on a variety of features.

2.4 Machine Learning for Computational Etymology

In the human sense of the word, a dictionary contains more than just translations. One of the most important types of data in a dictionary is a word's etymology, or origin. In recent years, researchers have developed computational methods for determining relationships between languages. For surveys of the field of linguistic phylogenetics, see Nichols and Warnow (2008) and Dunn (2015). However, there is little work on computationally learning the etymological relationships between individual words. There are efforts to construct a Proto-Indo European lexicon (Pyysalo, 2017), and researchers have shown that knowing a word's etymology can help with text classification tasks (Fang, Li, and Ide, 2009; Nastase and Strapparava, 2013) and reconstructing language phylogenies (Nouri and Yangarber, 2016).

The term "computational etymology" has very few existing mentions in the literature. To the best of my knowledge, Yang (2004) was the first to use the term, but his usage of this term only referred to the alignment of cognates. My work defines computational etymology more broadly, and investigating multiple processes of word formation and the relationship between words across languages. My work is pioneering this relatively understudied field, investigating statistical and modern neural models for modeling etymology across thousands of languages.

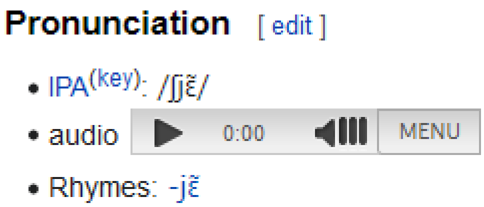
Though some computational etymology tasks defined in this dissertation are new, there are several related threads of work, including cognate prediction, surveyed above in Section 2.3. The etymology of a word can also include when the word entered its language. Identifying the date of first use of a word has historically involved lexicographers scouring old literature and manuscripts. For high-resource languages like English, existing work (e.g. Fischer, 1998) details different processes of forming neologisms, like clipping and borrowing. Dictionaries of neologisms (e.g. J. Algeo and A. S. Algeo, 1993)) list years or even specific dates of the first use of a word. In recent years, there have been some investigations on neologisms computationally (e.g. Ahmad, 2000; Kerremans, Stegmayr, and Schmid, 2011; Ryskina et al., 2020), and a few online dictionaries like Wiktionary and Merriam-Webster contain information about a word's year of first use. However, these resources vary in the amount of information they provide and are often limited to a handful of languages. My work utilizes the Google n-grams Corpus (Michel et al., 2011), which contains word usage over time by capturing the temporal distribution of n-grams derived from millions of scanned books. Most similar to my work is Petersen et al. (2012), who quantify word birth and death using statistical formulas. In contrast, I experiment with several diverse models, including neural networks, to model the birth of words.

Chapter 3

Constructing a Comprehensive Panlinguistic Dictionary

Wiktionary[1] is a free online multilingual dictionary containing a plethora of interesting data. This data does not exist in an immediately useful form, and while there are existing parsers that can extract some of this information (see Section 2.1), other types of information that I am interested in (e.g. etymology) have not been adequately extracted. This chapter presents Yawipa, my comprehensive Wiktionary parser that performs extraction and normalization of data contained in Wiktionary. The latter half of this chapter presents a new dictionary-based criterion for core vocabulary lists using translations extracted from Wiktionary to support the other dictionary induction efforts described in the following several chapters of this dissertation.

[1] www.wiktionary.org



Pronunciation [edit]

- IPA(key): /ʃjɛ̃/
- audio
- Rhymes: -jɛ̃

Figure 3.1: Pronunciation information in the English edition of Wiktionary for the French word *chien*.

This chapter contains some work originally published in Wu and Yarowsky (2020a), Wu and Yarowsky (2020b), and Wu, Nicolai, and Yarowsky (2020).

3.1 Yawipa

As a multilingual resource, Wiktionary exists as a set of *editions* written in a specific language. That is, the English edition is written in English, while the French edition is written in French. Any edition can contain entries for words in any language. For example, Figure 3.1 shows a screenshot of the English Wiktionary's pronunciation information for the French word *chien*. I use the terms *<lang> edition* and *<lang> Wiktionary* interchangeably.

Parsing Wiktionary. The data within Wiktionary exists as semi-structured information. Monthly dumps of all Wiktionary articles is available in XML at this link,[2] where XX is the language code for the Wiktionary edition of interest. Within the XML dump, the content of each Wiktionary page is encoded as MediaWiki markup, a MarkDown-like for-

[2] https://dumps.wikimedia.org/XXwiktionary/latest/XXwiktionary-latest-pages-articles.xml.bz2

mat with some additional features including *templates*, which get expanded via Lua code on the Wiktionary backend before being rendered into HTML. An alternative to parsing the MediaWiki markup is to parse the generated HTML pages that users see in their web browser. Parsing the HTML is more difficult because of the large differences in the generated HTML. However, the HTML sometimes contains additional information that is not present in the MediaWiki markup code. A few existing Wiktionary parsers operate on the HTML, extracting a small set of targeted information (e.g. Kirov, Cotterell, et al., 2018; Lee et al., 2020). Yawipa operates on the MediaWiki markup in the XML dump largely for ease of development and comprehensiveness.

3.1.1 Implementation Details

As Wiktionary is freely editable, the data is constantly being expanded and improved. Thus, one of Yawipa's goals is to be easily extensible so that researchers can write new parsers or edit existing ones to further their own extraction needs. Yawipa is written in the Julia programming language and exists as both a library and a runnable program. It processes the public Wiktionary XML dump.

The Wiktionary XML dump contains much metadata which Yawipa ignores. It only parses the page contents, which is formatted in MediaWiki markup, a format similar to MarkDown but supports *templates*, which Wiktionary expands when rendering the page into HTML. This is the same markup that a user would see when clicking "Edit" in the top right corner of a Wiktionary page. Yawipa splits this markup into "blocks" of contents, each of which have a header. These blocks are realized as sections in the HTML page that the user sees when visiting Wiktionary online. On each block, Yawipa runs a set of parsing functions, each of which is specialized for a specific type of information that the user wishes to extract. For example, a typical parsing function is shown below:

```
function parse_formof(dk::DictKey, heading::String, text::String)
    result = []
    for x in parsetemplates(text)
        if x.tag ∈ FORM_OF_TEMPLATES || endswith(x.tag, " of")
            push!(result, [x.tag, x.lang, x.content..., x.attrs...])
        end
    end
    return result
end
```

This function parses "form-of" relations from the English Wiktionary and is highly readable: *for every template, if it is a form-of template, or its tag ends with " of", add it to the results list.* Form-of is a relation in Wiktionary encompassing variants of a word, such as inflections, abbreviations, and misspellings. Each parsing function takes three arguments: a `DictKey`, the block heading, and the block text content. `DictKey` is a mutable struct defined in Yawipa containing three members:

```
mutable struct DictKey
    lang::String
    word::String
    pos::String
end
```

All results parsed from Wiktionary are keyed on this 3-tuple (language, word, part of speech) indicating the entry of the word from which the information was extracted.[3] Programmatically, this is a struct that is mutable, because certain parsing actions (e.g. parsing

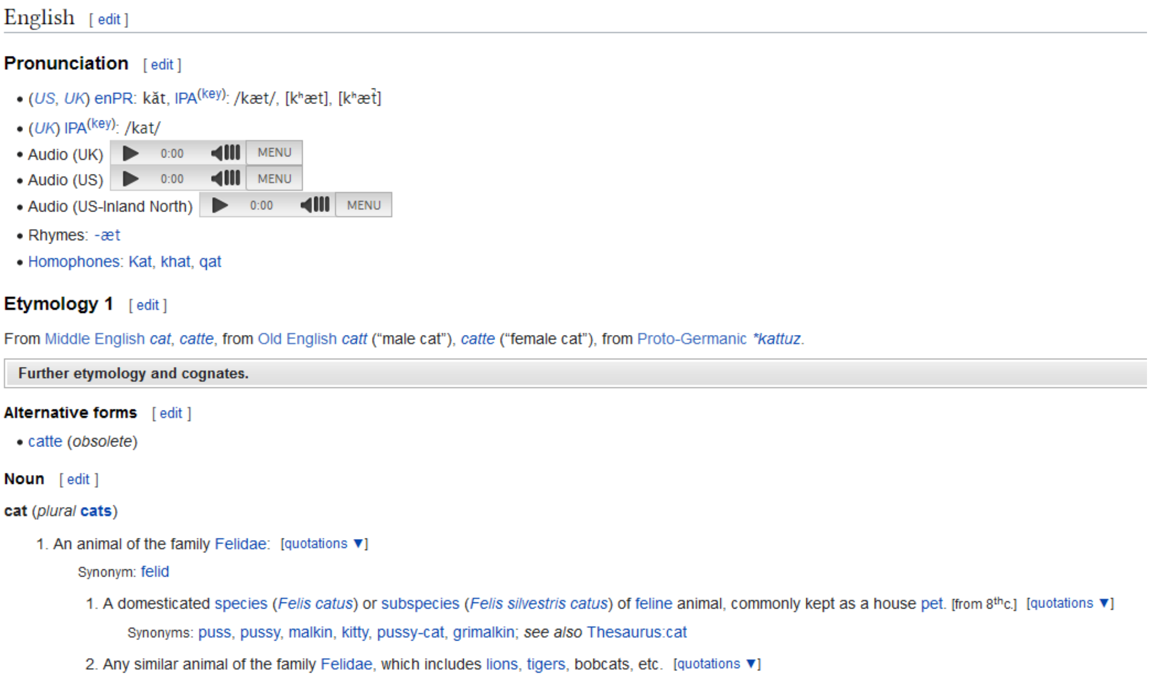
[3] Part of speech is important because of polysemous words, e.g. the noun *refuse* vs. the verb *refuse*.

part of speech) may wish to assign a new value to this key. The `parsetemplates` function does the heavy lifting parsing and extracting fields from the structured Wiktionary templates, allowing Yawipa to understand templates such as `{{der|en|ang|dox|t=dark, swarthy}}`. This template is found in the etymology section of an entry, and a interpretation in plain English would be: "this **En**glish word is **der**ived from the Old English (**ang**) word **dox**, whose **t**ranslation is **dark** or **swarthy**", where the data contained in the original template is bolded. It is the responsibility of each parsing function to handle the information in a template.

Each parsing function returns a list of results, which typically contains the type of information, language of the word, the word itself, and the normalized information. The output of Yawipa is a tab-separated (`.tsv`) file, where the first three columns are the language, word, and part of speech of the entry[4] from which the row's information was extracted. The fourth column is the type of information extracted (pronunciation, translation, etymology, etc.), and the following columns are the normalized output of each parsing function, specific to the type of information extracted and normalized.

In addition to extracting information from nearly every template in Wiktionary, Yawipa also normalizes this information into a usable format. For example, many existing Wiktionary parsers extract translations from translation templates `{{t|...}}`, but Yawipa also extract translations from etymology and definitions. For example, Yawipa normal-

[4] Recall that a Wiktionary page may have multiple entries. For example, *dog* is a word in English, Afrikaans, Danish, Dutch, Kriol, Mbabaram, Navajo, Norwegian Bokmal, Portuguese, Romanian, Swedish, Torres Strait Creole, Volapük, and Westrobothnian. All these entries occur on the same page: https://en.wiktionary.org/wiki/dog.



English [edit]

Pronunciation [edit]

- (*US*, *UK*) enPR: kăt, IPA(key): /kæt/, [kʰæt], [kʰæt̚]
- (*UK*) IPA(key): /kat/
- Audio (UK)
- Audio (US)
- Audio (US-Inland North)
- Rhymes: -æt
- Homophones: Kat, khat, qat

Etymology 1 [edit]

From Middle English *cat*, *catte*, from Old English *catt* (“male cat”), *catte* (“female cat”), from Proto-Germanic **kattuz*.

Further etymology and cognates.

Alternative forms [edit]

- catte (*obsolete*)

Noun [edit]

cat (*plural* **cats**)

1. An animal of the family Felidae: [quotations ▼]

   Synonym: felid

   1. A domesticated species (*Felis catus*) or subspecies (*Felis silvestris catus*) of feline animal, commonly kept as a house pet. [from 8th c.] [quotations ▼]

      Synonyms: puss, pussy, malkin, kitty, pussy-cat, grimalkin; *see also* Thesaurus:cat

   2. Any similar animal of the family Felidae, which includes lions, tigers, bobcats, etc. [quotations ▼]

Figure 3.2: Snippet from the English Wiktionary page for the English word *cat*.

izes *t=dark, swarthy* as two separate translations, *dark* and swarthy, for the Old English word *dox*.

Due to the sequential processing of the Wiktionary XML dump, part of speech in an entry occurs after pronunciation (see Figure 3.2). Thus, the parser will not assign a part of speech when extracting pronunciations. It is necessary to run an additional post-processing script provided by Yawipa to fill in missing part of speech.

3.1.2 Extracted Data

Yawipa extracts and normalizes numerous types of information from Wiktionary, as shown in Figure 3.3. These are all annotated in a Wiktionary page, and may be structured information (e.g. cognates, formof, anagrams, translations), or unstructured (definitions), or a combination of both (etymology, pronunciations). In descending order of frequency, these are:

- def. Definitions.
- pos. Part of Speech.
- formof. Morphological relations, such as inflections, abbreviations, etc.
- deftr. Definition translations. This is one of Yawipa's novel contributions (described below).
- pron. Pronunciation.
- tr. Translations.
- etym. Etymology.
- der. Derived Terms.
- rel. Related Terms.
- anagrams. Anagrams.
- alter. Alternate Terms.
- cog. Cognates.
- syn. Synonyms.
- desc. Descendants.
- ant. Antonyms.
- hypo. Hyponyms.
- coord. Coordinate Terms.
- hyper. Hypernyms.
- noncog. Non cognates.
- mero. Meronyms.
- holo. Holonyms.

3.1.3 Translations

Wiktionary also contains translations, an important component in any dictionary. While Wiktionary provides an API to access translations, this is not convenient for bulk

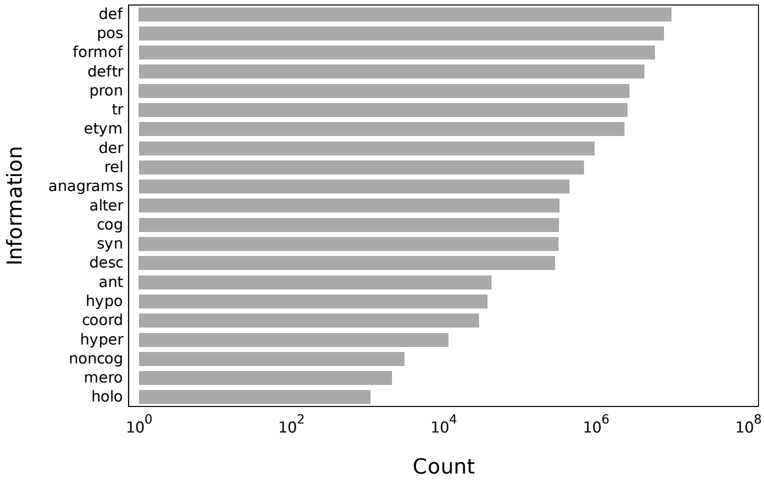


Figure 3.3: Counts of the different types of information extracted and normalized from Wiktionary. Note the log scale on the x-axis.

analysis. Therefore, Yawipa extracts all translations in one go. Within the scientific literature, there are a few projects that have extracted data directly from the Wiktionary dumps: wikt2dict (Ács, Pajkossy, and Kornai, 2013; Ács, 2014) extracts translations from the translation tables in the Wiktionary articles. This codebase supports triangulation between language to discover new translations. Kirov, Sylak-Glassman, et al. (2016) (henceforth Kirov) also extracts translations from translation tables, in addition to morphological paradigms, which were the main focus of their work.

Yawipa extracts translations from translation tables as well as from *definitions* of the word. Definitions are a valuable source of translations, and I am not aware of existing work that extracts lexical translations from freeform definitions. Extracting translations from definitions is a challenging task, since definitions are unstructured and generally freeform text, while translation tables are structured. I utilized a combination of string

Parser	Terms	# Langs
Ács (2014)	1589383	2417
Kirov, Sylak-Glassman, et al. (2016)	1577374	2165
Ours (translations)	1575392	2406
Ours (definitions)	1181666	2800
Ours (both)	2296208	3640

Table 3.1: Number of foreign-English translations extracted by various translation extraction systems.

regular expression matching and other heuristics to convert the definition strings into short lexical translations.

Below, I analyze translations extracted using various systems. In these comparisons, I used the English Wiktionary dump with articles only from May 2019. I ran wikt2dict with a small modification to the code to allow extracting translations for all languages (rather than the small subset that they previously defined). Kirov's parse is from an older (2015) edition of Wiktionary. For each parse, I removed duplicate translations and kept only foreign-English translation pairs.

Wiktionary contain 3931 languages.[5] wikt2dict parse contains 2367 languages, and Kirov's contains 2166. Both share 1640 languages, while separately wikt2dict has 727 not in Kirov, and Kirov has 526. As shown in Table 3.1, extracting translations from definitions covers considerably more languages and terms than just translation tables.

wikt2dict's and Yawipa's translation extraction from translation tables are very similar, which makes sense; both are using the same data. The differences largely come from wikt2dict not postprocessing its output, so it include entries like Finnish *[[puhua]] [[um-*

[5] As of April 2019. https://en.wiktionary.org/wiki/Wiktionary:Statistics

met ja lammet]] (with brackets), or words with unmatched parentheses. There is also some variation in translations, usually in proper nouns: wikt2dict has "Solar System", while Kirov has "the Solar System" as translations for the Zaza word *Sistemê Roci*.

In terms of the number of foreign words and languages where wikt2dict and Yawipa's method extracted more words than Kirov, this is likely due to users simply adding more words since the time Kirov's translations were extracted (we were not able to obtain the code to run their extraction). On the other hand, for some languages, Kirov was able to extract more translations due to parsing morphological information outside of the translation tables. Yawipa's innovation of extracting translation from definitions substantially increases the number of available translations.

3.1.4 Pronunciations

Wiktionary contains a plethora of interesting information, as presented above. In this section, I focus specifically on the pronunciation annotations in Wiktionary, which are relatively understudied. For any given word, Wiktionary may include data about its pronunciation written using the International Phonetic Alphabet (IPA). This pronunciation may be both phonetic and phonemic and may also include additional information like hyphenation, dialectical variation, and even audio files of speakers pronouncing the words. These types of data have been shown to be useful for many tasks, such as grapheme-to-phoneme transduction, e.g. in recent SIGMORPHON shared tasks (Gorman et al., 2020). There are many existing parsing efforts that have extracted pronunciation information from Wiktionary. Recent extractions of data from Wiktionary focus on obtaining high-quality pronunciations from a *single* edition of Wiktionary, usually the English edition (e.g. Wu and Yarowsky, 2020a; Sajous, Calderone, and Hathout, 2020; Lee et al., 2020). However, substantial increases in data can be obtained by parsing other editions of Wiktionary, which have been shown to be helpful for downstream tasks. For example, Schlippe, Ochs, and Schultz (2010) extract pronunciations from the English, French, German, and Spanish editions, and Deri and Knight (2016) extract pronunciations from the English, German, Greek, Japanese, Korean, and Russian editions.

Targeting the larger Wiktionaries for increased coverage and those not dealt with in existing previous work, I construct new pronunciation parsers for the French, Spanish, Malagasy, Italian, and Greek editions of Wiktionary. Combined with pronunciations from the English Wiktionary, this totals to over 5.3 million words, which to my knowledge is the largest pronunciation lexicon to date and also a unique comparable corpora of pronunciations. In Section 3.1.4.1, I show that my extracted pronunciations are a substantial increase in data, covering numerous pronunciations not in the English Wiktionary. This is especially beneficial for low-resource languages. In Section 3.1.4.2, I analyze this data and find that a small portion of these pronunciations may be low-quality and computer-generated. In Section 3.1.4.3, I present a novel visualization technique for analyzing the use of stress in IPA pronunciations. In Section 3.1.4.4, I experiment on the combined task of massively multilingual syllabification and stress detection. My neural sequence-to-sequence model with copy attention outperforms a sequence labeling baseline, especially in very low-resource scenarios, underscoring the contributions of additional languages to the task. In addition, I find that a multitask approach of predicting both stress and syllabification can improve the performance on syllabification alone.

3.1.4.1 Wiktionary Pronunciation Extraction

As a multilingual resource, Wiktionary exists as a set of numerous *editions*. That is, the English Wiktionary is written in English by and for English speakers, while the French Wiktionary is written in French by and for French speakers. Any edition can contain entries for words in any language. For example, Figure 3.1 shows a screenshot of the English Wiktionary's pronunciation information for the French word *chien*. I use the terms *<lang> edition* and *<lang> Wiktionary* interchangeably.

Why parse other editions of Wiktionary? Speakers of different languages have different priorities when annotating data. One can assume that an editor of the Spanish Wiktionary is more likely to provide pronunciations for Spanish words before working on English words. My effort at extracting a new dataset of pronunciations from 6 different editions of Wiktionary resulted in a total of over 5.3 million *unique* IPA pronunciations across 2,177 languages. Note that because the data comes from multiple editions, a word may have multiple annotated pronunciations, making my dataset an interesting comparable corpora. Figure 3.4 shows the 16 languages with the most data in this dataset, along with the contribution of each edition of Wiktionary from which I parsed and extracted IPA pronunciations.

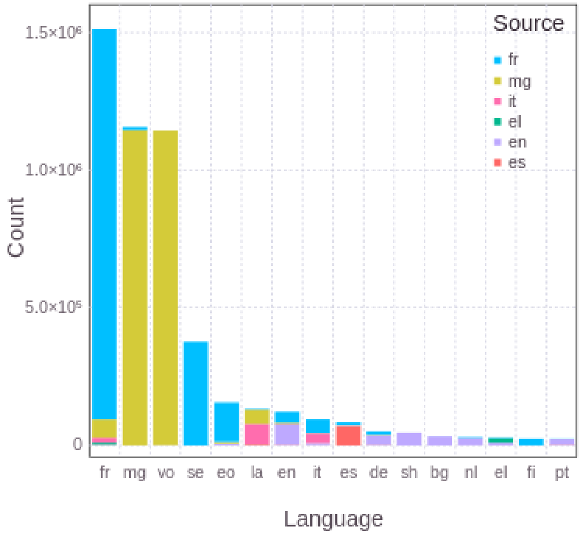


Figure 3.4: The top 16 languages in terms of number of pronunciations, with contributions from multiple editions of Wiktionary.

I draw several insights from Figure 3.4. First, the inclusion of pronunciations from non-English Wiktionaries represents substantial gains over the English edition. Though the English edition is the largest Wiktionary by number of entries,[6] the French edition contains a huge number of pronunciations for French words, dwarfing other editions that I parsed. The French Wiktionary also supplies the entirety of the pronunciations for Northern Sami words (`se`, spoken in Norway, Sweden, and Finland), most of the available pronunciations for Esperanto (`eo`) and Italian (`it`) words, and also words in 1,198 other low-resource languages not shown in the long tail of Figure 3.4. In contrast, the English edition (the second largest supplier) is the sole supplier of pronunciations in 416 languages.

Parsing Implementation. The Yawipa framework (Wu and Yarowsky, 2020a) ex-

[6] `https://meta.wikimedia.org/wiki/Wiktionary`

tracts data from the XML dump of Wiktionary.[7] Every entry is encoded in MediaWiki markup, which is similar to Markdown but includes special *templates* (enclosed in double braces) which programmatically generates HTML that is displayed to a user who visits the Wiktionary website. For example, in the English wiktionary, the entry for the French word *chien* contains the following markup (rendered in Figure 3.1):

```
===Pronunciation===
* {{fr-IPA}}
* {{audio|fr|Fr-chien.ogg|audio}}
* {{rhymes|fr|jɛ̃}}
```

These three templates generate the three bullet points in Figure 3.1. Note that the `{{fr-IPA}}` template generates the IPA pronunciation, so the IPA itself does not exist in the English Wiktionary dump. Thus, one can only extract IPA from the French edition (see below), underscoring the need to parse multiple Wiktionary editions for multiple sources of pronunciations.

```
=== {{S|nom|fr}} ===
{{fr-rég|ʃjɛ̃}}
```

Above is the French Wiktionary's pronunciation for the word *chien*. A template (`fr-rég`) is also used, but the IPA is extractable from the markup. Each edition of Wiktionary has its own conventions on formatting and templates, thus requiring a separate parser specifically for that edition. For implementation details, please see the repository https://github.com/wswu/yawipa.

[7] https://dumps.wikimedia.org/enwiktionary/latest/XXwiktionary-latest-pages-articles.xml.bz2, where XX is replaced with a two-letter ISO 639-1 code.

3.1.4.2 Analysis of the Pronunciation Dataset

For high-resource languages, the home language edition (e.g. English edition for the English language) usually supplies the most pronunciations, but this is not always the case (e.g. the French Wiktionary provides more Italian pronunciations than the Italian edition). In terms of amount of data, two languages are outliers: Malagasy (`mg`, an Austronesian language spoken in Madagascar) and Volapük (`vo`, a constructed language). As relatively less spoken languages, these languages have a disproportionately large amount of data. Why is this so?

The data for these two languages come from the Malagasy edition, which was parsed because of its high ranking in the List of Wiktionaries.[8] Both Malagasy and Volapük are inflected languages[9] whose IPA pronunciations seem to be entirely computer-generated using a regular transduction process from orthography to IPA, which was exploited to create a large set of pronunciations for these two languages.

I also find that some Latin pronunciations may be machine-generated. For example, the Malagasy edition supplies /kontabulawit/ as the pronunciation for the Latin *contabulavit* and /d̪ẽːonstɾat/ for *demonstrat*. These pronunciations lack stress and syllable markings, and in the case of *demonstrat*, do not agree with established pronunciations of Latin. thus leading us to believe that these were machine-generated pronunciations. In contrast, the English edition contains both well-formed classical and ecclesiastical Latin

[8] https://en.wikipedia.org/wiki/List_of_Wiktionaries

[9] Inflected words have their own Wiktionary entry, which can exponentially increase the number of pronunciations.

pronunciations with stress and syllable markers, but only for the dictionary forms *contabulō* /konˈta.bu.loː/ and *dēmōnstrō* /deːˈmon.stroː/.

I must emphasize that I am not condemning the use of machine-generated pronunciations. For many languages, e.g. Spanish and Latin, the spelling of a word reflects its pronunciation, so generated pronunciations are likely to be accurate. Indeed, the existence of pronunciation templates such as `{{fr-IPA}}` are well-researched additions to Wiktionary that alleviate the need for humans to manually input IPA pronunciations, thus reducing the potential for human error. I fully support the use of these templates (though they make my parsing job harder), and I would love to see them standardized across all Wiktionary editions, so that editions such as the Malagasy edition can benefit from contributions to the English edition (or any other edition, for that matter).

I do caution researchers that the data contained in crowd-sourced resources such as Wiktionary may not be thoroughly vetted for accuracy, as I have discovered. Fortunately, the openness of these crowdsourced data allows for community members to quickly intervene when problematic data is found. One especially poignant example in recent news is the Scots Wikipedia, a large portion of which was recently revealed to be written by an American teenager who is not a Scots speaker.[10] Essentially, this teenager translated English articles into "Scots" by systematically rewriting English words to sound as if they were spoken with a Scottish accent, in the same vein as some Latin "IPA" pronunciations in the Malagasy Wiktionary.

[10] https://www.reddit.com/r/Scotland/comments/ig9jia/ive_discovered_that_almost_every_single_article

3.1.4.3 Visualizing Syllabification

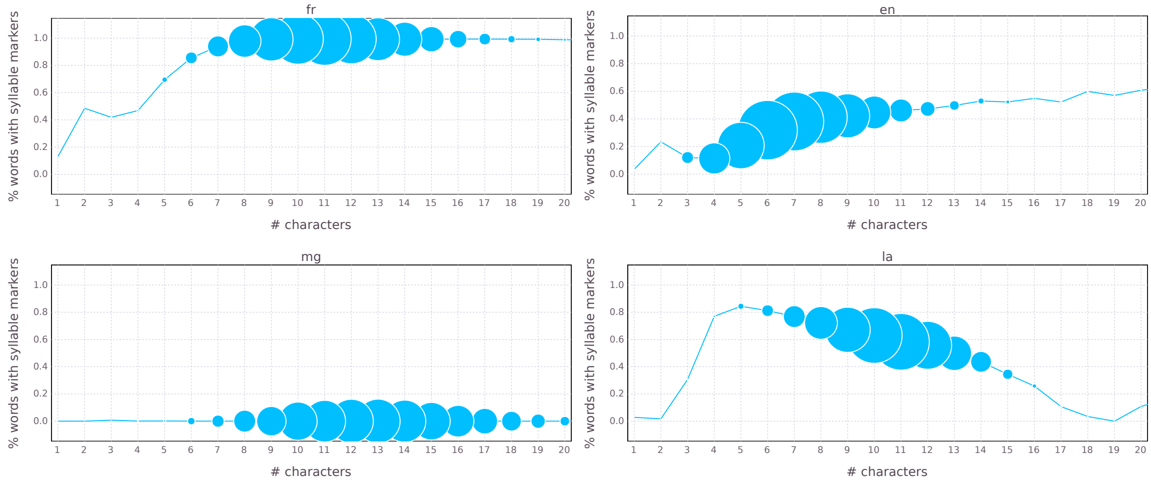


Figure 3.5: Percentage of French, English, Malagasy, and Latin words containing syllable markers, by length of word. The size of the points indicates the number of words and cannot be compared among graphs.

IPA has the ability to mark syllable boundaries (.) as well as primary (ˈ) and secondary (ˌ) stress. Words in some languages, e.g. Malay, do not have stress, and sometimes stress can be double marked (") for extra stress. I first quantify IPA stress and syllabification in my extracted dataset, and then present multilingual experiments on predicting syllabification and stress using this dataset.

I also develop a visualization technique to understand the distribution of words in each language that contain syllable boundaries (Figure 3.5). These bubble charts plot the number of characters in a word (x-axis), the percentage of words containing syllable markers (y-axis), and the number of words in these categories (size of the dot). These charts can help researchers to quickly quantify the presence of syllable markers, one component of high-quality IPA pronunciations. I consider a word to be syllabified if it contains any of the following three symbols: . ˈ ˌ

Ideally, one would expect that the longer the word, the higher the percentage of words that have syllables marked. French is a perfect example of this: once words reach 9–10 characters in length, they all contain syllable markers. By examining these plots, one can easily identify examples of problematic IPA syllabification in Malagasy (`mg`) and Latin (`la`) words. For Malagasy words, syllable boundaries simply do not exist. Latin words follow an unusual negative-sloped curve, where words around 4–6 characters in length are more likely to have syllables marked, but longer words are less likely to have syllable boundaries marked. This analysis actually is consistent with my earlier finding in **??**: because Latin is a highly inflected language, the dictionary forms contain high-quality IPA, but the overwhelming number of pronunciations are actually machine-generated for inflected forms, which may not have the syllables marked. English is a middle ground in terms of quality. While there exists the expected upward slope as the length of the word increases, the percentage of words with syllable markers never approaches 100%. A manual review of several English pronunciations indicates that annotators simply did not include syllable boundaries for many English words. Further analyses could shed light on the reasons for the negligence of the annotators, or other phenomena that might explain the lack of syllable markers.

3.1.4.4 Experiments on Syllabification and Stress Prediction

In this section, I present experiments on multilingual syllable and stress prediction. In the linguistics literature, many studies have shown that awareness of syllable boundaries can improve word recognition performance in children (e.g. McBride-Chang et al., 2004; Plaza and Cohen, 2007; Guldenoglu, 2016). Speech syllabification is also a common step in a speech recognition pipeline. Syllabification of text is not a new task, and has been explored via a variety of methods, including rule-based and grammar-based approaches (e.g. Weerasinghe, Wasala, and Gamage, 2005; Müller, 2006) and data-driven approaches (e.g. Bartlett, Kondrak, and Cherry, 2008; Nicolai, Yao, and Kondrak, 2016; Gyanendro Singh, Laitonjam, and Ranbir Singh, 2016). However, previous work has focused primarily on a handful of languages, and some focus on orthographic syllabification rather than phonemic segmentation. Some use CELEX (Baayen, Piepenbrock, and Gulikers, 1996), a popular dataset containing syllabified text, but it only contains syllabified words in English, German, and Dutch. In contrast, my extracted pronunciation lexicon is a unique multilingual resource that allows for developing and evaluating models and approaches on the new combined task of massively multilingual IPA syllabification *and* stress prediction across hundreds of languages. In this task, given unmarked IPA, a model must insert syllable markers or stress markers at the appropriate locations.

Data. For the experimental tasks, I filter my extracted pronunciation dataset, keeping only IPA containing syllable boundaries or stress markers,[11] so that there is ground truth

[11] A stress marker can server as a syllable boundary, e.g. for the English word *consume* /kənˈsum/.

for training the models. This resulted in 93,206 IPA pronunciations across 174 languages, which are split into a 80-10-10 train-dev-test stratified split (same proportion of languages in each set).

Models. I first build a baseline: a multilingual character BiLSTM sequence tagger with 256 hidden size (B) that predicts both stress and syllabification (Str & Syl) or syllabification alone (Syl). The data is preprocessed such that each IPA character is labelled with 0 for no stress or syllable, 1 for primary stress (ˈ), 2 for secondary stress (ˌ), and 3 for syllable boundary (.). A token specifying the language is included so that the model will incorporate knowledge of the language. For example:

IPA:	/ˌɪn.flu.ˈɛn.zə/
Input:	eng ɪ n f l u ɛ n z ə
Output:	0 2 0 3 0 0 1 0 3 0

For comparison, I experiment with two modern seq2seq models: the default encoder-decoder model (S) in OpenNMT-py (Klein, Kim, Deng, Senellart, et al., 2017), and the same model with copy attention (SC) (See, P. J. Liu, and Manning, 2017). In this scenario, I formulate syllabification and stress prediction as a sequence generation task, where the input is an unstressed, unsyllabified IPA, and the output is the original IPA sequence containing both stress and syllable markers.

I then treat syllabification and stress prediction in a pipelined approach (Syl → Str), where the first model (B or SC) will predict syllable boundaries, and then a second model will predict the stress. Stress classification is a 3-class classification problem: given a syllable, predict primary stress, secondary stress, or no stress. The structure of this stress

Model	Acc1	CED	Acc5	CED5
B Syl	68	.48	—	—
SC Syl	79	.42	96	.11
B Syl → Str	53	.88	—	—
SC Syl → Str	31	1.13	—	—
B Str & Syl	52	.89	—	—
-Str	68	.49	—	—
S Str & Syl	69	.72	89	.25
-Str	77	.47	93	.16
SC Str & Syl	74	.54	92	.17
-Str	81	.35	95	.11

Table 3.2: Results on the syllabification and stress prediction tasks. B is a BiLSTM sequence tagger, S is a sequence-to-sequence encoder-decoder, and SC is the same model with copy attention. Syl indicates the syllabification prediction task, Str indicates the stress prediction task, -Str indicates evaluating by disregarding stress markers. Acc1 is 1-best accuracy, Acc5 is 5-best accuracy (is the gold in the top 5 hypotheses?), CED is mean character edit distance, and CED5 is edit distance of the hypothesis in the top 5 predictions closest to the gold.

classifier is also a BiLSTM, where the hidden state of the syllable in question is passed to a dense feed-forward layer, then a softmax.

A summary of experimental results is in Table 3.2. The baseline BiLSTM model performs consistently worse than the seq2seq models. This is somewhat surprising, since the seq2seq task is a more challenging task: the model must generate the IPA characters along with stress and syllable markers. However, the seq2seq model is able to generate the correct sequence of IPA characters, minus stress and syllable markers, in 95% (for regular attention) and 99% (for copy attention) of test examples, alleviating these concerns and proving the effectiveness of copy attention for this task.

The pipeline approach performs substantially worse than the multitask approach. In the pipeline, the syllabification model first predicts the syllable boundaries, then the stress classifier produces a classification for each syllable. I find that with the pipeline approach, it is impossible to improve upon the first step in the pipeline. Thus, if the syllabification step does not correctly identify syllable boundaries, the final pronunciation will never be correct, even if the stress is correctly predicted for each syllable.

Finally, multitask training on both syllabification and stress marking improves performance over syllabification alone. I believe this is because stress and syllable prediction are two somewhat overlapping tasks. If a model can label stress, then it should have some notion of where syllables are. The (-Str) rows in Table 3.2 show performance on syllabification by evaluating the output of the multitask model preprocessed to replace all stress marks with syllable boundaries.

The large majority of languages in this dataset can be considered low-resource, a specific interest of my experiments. 154 of the 174 languages have much fewer than 466 training examples (0.5% of the entire dataset), yet the average accuracy on these languages is an impressive 67% for syllabification (B Str & Syl - Str) and 51% for both syllabification and stress prediction (B Str & Syl). This highlights the contribution of other languages in a single massively multilingual model trained to do both tasks. Other researchers have found that good performance on syllabification requires much more data than this (Nicolai, Yao, and Kondrak, 2016). I highlight the fact that many of the languages have less than 10 test examples and can be considered truly low-resource; the contribution of many other languages allows the multilingual models to predict the correct pronunciation with minimal training data in a specific language. Though I find that multilingual training helps for low-resource languages, it can also help with high-resource languages: in the SC Str & Syl scenario, a model trained only on French obtained 92.1% on the French test words, compared to the multilingual model at 98.1% accuracy.

3.1.5 Conclusion

I extracted the largest dataset of IPA pronunciations to date, by combining IPA from the French, Spanish, Malagasy, Italian, and Greek editions of Wiktionary along with existing pronunciations from the English edition, totaling to 5.3 million pronunciations. I developed a visualization method for examining syllabification in large datasets, which can give indications about the quality of IPA pronunciations. Finally, I experiment on the new combined task of massively multilingual prediction of syllabification and stress using a variety of models and approaches, showing success with a multitask multilingual sequence-to-sequence model.

I envision this newly extracted pronunciation dataset and the analysis methods presented above to be especially useful for researchers interested in lexicography and spoken language technologies. In terms of lexicography, this dataset is a unique comparable corpus containing annotations from several editions of Wiktionary, each representing a distinct population of speakers. In several cases, the same pronunciation is supplied by multiple editions, and some editions use phonetic rather than phonemic IPA. Future work can address questions such as: When and why might different editions disagree on a pronunciation? Why do some words have pronunciations and others don't? In addition, I would like to investigate the use of this pronunciation dataset in language learning of core vocabulary of low-resource languages (Wu, Nicolai, and Yarowsky, 2020) and modeling etymology relationships between words (Wu, Duh, and Yarowsky, 2021).

3.1.6 Open Source

Yawipa is open-source and is available at `https://github.com/wswu/yawipa`. I solicit improvements and encourage further research with this software package.

3.2 Core Vocabulary

Dictionaries (bilingual translation lexicons) are available for most of the world's languages, but coverage can be sparse for those with fewer resources. In sparse dictionaries, many entries are *core vocabulary* words from lists such as the Swadesh list (Swadesh, 1952; Swadesh, 1955), probably the most well-known formulation of a core vocabulary containing approximately 100–200 words, depending on the version. This list of basic words is used in historical comparative linguistics to determine the relationships between languages, and there have been many attempts to revise or expand these concept lists for this purpose.

Morris Swadesh chose the words in the Swadesh lists based on certain criteria: the words should be culturally universal, stable over time (not likely to change meaning), and not likely to be borrowed. Swadesh lists now exist in over 1000 languages and can be used as a dictionary to perform lexical translations. However, in a low-resource setting, the ability to translate a mere 100 concepts is insufficient for understanding in a language. In addition, the Swadesh list, like many other lists, was manually created and revised through years of experience and extensive fieldwork. Inspired by these shortcomings, I propose a novel data-driven criterion for a core vocabulary list: high coverage in dictionaries of different languages.

This section presents the automatic creation of a core vocabulary list based on the number of entries a concept has in dictionaries. That is, the criterion for inclusion in my list is the consensus of many lexicographers who deemed a word important enough for inclusion in a language's (possibly small) dictionary. The top entries of my list are presented in Table 3.3. I empirically find that roughly 3000 words is an adequate size for the list, which is on par with other major core vocabulary lists. In-depth analysis illustrates that due to substantial overlap with several established lists, my core vocabulary can serve well for downstream tasks such as language phylogenetics and language learning. In terms of low resource languages, my core vocabulary consists of words that should be prioritized for elicitation should they not exist in a dictionary. I also successfully experiment on the task of dictionary induction by generating these core words with cognate prediction models.

1.	one	2.	water	3.	two
4.	dog	5.	fish	6.	tongue
7.	eye	8.	ear	9.	fire
10.	blood	11.	stone	12.	see
13.	bone	14.	skin	15.	name
16.	tooth	17.	nose	18.	star
19.	die	20.	come	21.	head
22.	hear	23.	woman	24.	path
25.	mouth	26.	breast	27.	night
28.	eat	29.	you	30.	moon
31.	smoke	32.	hair	33.	bird
34.	black	35.	fly	36.	sleep
37.	man	38.	egg	39.	new
40.	three	41.	white	42.	I
43.	liver	44.	hand	45.	rain
46.	hide	47.	tail	48.	we
49.	drink	50.	louse	51.	snake
52.	good	53.	say	54.	small
55.	fat	56.	sun	57.	tree
58.	cloud	59.	meat	60.	rock
61.	neck	62.	sand	63.	wind
64.	cold	65.	leaf	66.	dry
67.	earth	68.	four	69.	person
70.	go	71.	kill	72.	bite
73.	that	74.	red	75.	burn
76.	mother	77.	road	78.	big
79.	sit	80.	father	81.	long
82.	five	83.	mountain	84.	male
85.	what	86.	knee	87.	leg
88.	root	89.	soil	90.	large
91.	grind	92.	ashes	93.	fall
94.	who	95.	right	96.	foot
97.	house	98.	all	99.	heavy
100.	back	101.	stand	102.	bad
103.	little	104.	child	105.	hot
106.	know	107.	ten	108.	give
109.	short	110.	walk	111.	dead
112.	female	113.	heart	114.	salt
115.	old	116.	hill	117.	belly
118.	sky	119.	laugh	120.	cut
121.	ash	122.	close	123.	wing
124.	six	125.	shoulder	126.	smell
127.	stick	128.	human being	129.	green
130.	dull	131.	seven	132.	single
133.	eight	134.	many	135.	far
136.	he	137.	breasts	138.	day
139.	the	140.	title	141.	yellow
142.	near	143.	nine	144.	full
145.	this	146.	lie	147.	dig
148.	where	149.	rat	150.	every

Table 3.3: Top 150 words from our core vocabulary list.

3.2.1 Construction

For the construction of my core vocabulary, I utilize LanguageNet,[12] a multilingual lexicon that is a subset of PanLex (Baldwin, Pool, and Colowick, 2010; Kamholz, Pool, and Colowick, 2014), a freely available multilingual dictionary. PanLex contains lexical translations across several thousands of the world's languages and has recently garnered interest in the multilingual research community. Its lexical translations are sourced from existing dictionaries and thesauri such as Wiktionary and WordNet. LanguageNet, as of September 2019, contains 1895 languages.

I employ a simple procedure: using English as a pivot, I collect counts of how many languages have a translation for each English concept. The concepts are then sorted in decreasing order by this count, resulting in a ranking of concepts by coreness. Up until recently, such a computational procedure would have been impossible without the computing resources and datasets available today.

Figure 3.6 shows the top 30 concepts along with the number of dictionaries that contain them.[13] The fact that so many languages' dictionaries contain these words is a strong indicator of the coreness of these words. This point is even more salient for dictionaries of low-resource languages: that so many lexicographers have included these words in their language's dictionary is a testament to the word's importance in the language and thus should be included in a list of core vocabulary. Figure 3.7 shows the rank of each concept

[12] `http://uakari.ling.washington.edu/languagenet`

[13] Here, I use *dictionary* to mean *language*, i.e. every language in PanLex has one dictionary. Each dictionary is represented by a separate ISO 639-3 language code, so this number represents language variants.

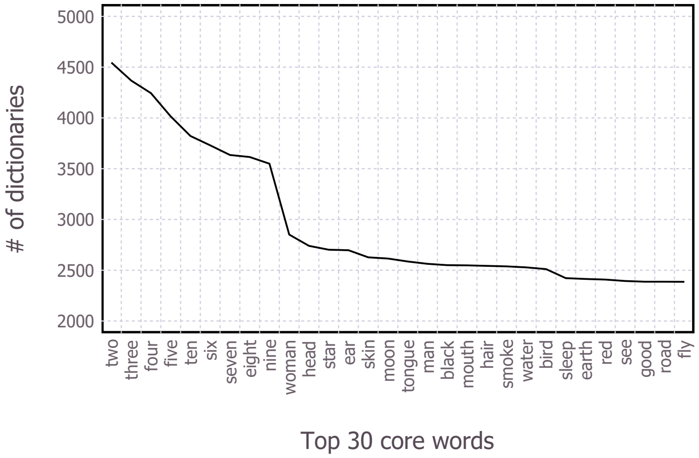


Figure 3.6: Top 30 concepts in the core vocabulary list, and the number of dictionaries containing the concept.

(in the core vocabulary) and the number of languages containing the concept. The curve follows a typical exponential (Zipfian) decay, in which the top 1000 words are (at least) contained in roughly 500 languages. Using this curve, I observe that around rank 3,000 is the point at which the curve begins to drastically flatten out. This indicates a reasonable threshold for the size of a core vocabulary list. For this work, we set a threshold of 3,000 concepts, above which comprise the core vocabulary list. Several other existing lists exhibit a similar vocabulary size.

3.2.2 Analysis

Linguists have always been interested in core vocabulary, and there have been many existing approaches for constructing sets of core words. Many of these lists share a substantial number of words, but the lists differ in the purpose of their construction. I examine two motivations: establishing linguistic relationships, and facilitating language acquisi-

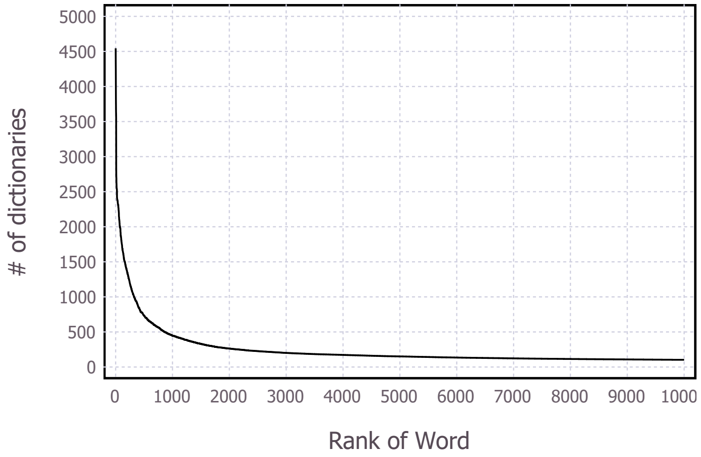


Figure 3.7: Top 10,000 core vocabulary concepts, and the number of dictionaries containing the concept.

tion. The former lists (*à la* Swadesh) are generally composed of words that are universal across cultures and are resistant to borrowing, so that a comparison across language of the words in these lists can help determine linguistic relationships. Words in the latter lists (for language learning) are often chosen for their frequency of use in written and spoken language as well as for their range of use across multiple genres or domains.

In this section, I show that my empirically derived, dictionary coverage–based lists have high overlap with several existing lists that were developed via these motivations and can indeed be used for such purposes. In addition, my core vocabulary list has high coverage over several well-known linguistic corpora which span multiple domains, making this list particularly suited for language learning.

List	Coverage	%
Swadesh	207/207	100
Dogolpolsky	15/15	100
Leipzig-Jakarta	100/100	100
Ogden	698/850	82
Dale–Chall	1669/2942	57
Oxford 3000	1525/2989	51
NGSL	1362/2801	49
Chinese	1518/2462	62
Russian	1243/1817	68

Table 3.4: Overlap with existing core vocabulary lists.

3.2.3 Comparison with Other Lists

I compare my 3000-word core vocabulary list with several other well-known concept lists:

Linguistically Motivated Lists. The Swadesh list (Swadesh, 1952) has already been extensively mentioned. The Dogolpolsky list (Trask, 2000) is a small set of 15 words that were chosen for their resistance to be replaced by other words over time. The Leipzig–Jakarta list (Haspelmath and Tadmor, 2009) is a set of 100 words that are most resistant to borrowing from other languages.

I also investigate the following language-learning lists:

- Ogden's Basic English: (Ogden, 1932) A list of 850 words compiled by C. K. Ogden of simple concepts encountered in everyday life.

- Oxford 3000: A list[14] of 3000 words (2989 unique lemmas) that were selected for their

[14] https://www.oxfordlearnersdictionaries.com/us/about/oxford3000

"importance and usefulness" for English language learners based on their frequency, range of domains, and familiarity in the English language.

- New General Service List (NGSL) (Browne, 2014): A list of 2801 lemmas along with their inflected forms, billed as a list of general words for English language learners. It is based on the Cambridge English Corpus and seeks to improve upon an earlier list, the General Service List (West, 1953).

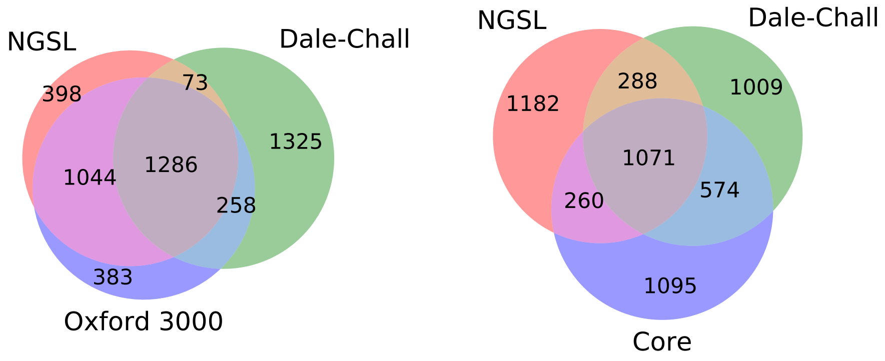
- Dale–Chall (Dale and Chall, 1948): A list of 3000 words that a United States 4th grader would know. This list is used in readability metrics.

In addition, I compare against two lists created for language learning purposes in non-English languages, in order to evaluate the linguistic universality of my core vocabulary list:

- Chinese. A wordlist from the Hanyu Shuiping Kaoshi (pre-2021 edition), the standardized Chinese Proficiency Test. I use words from levels 1–5 (roughly corresponding to B1 or B2 proficiency level), totaling 2500 words.

- Russian. A wordlist from OpenRussian.org containing 1819 words up to a B2 proficiency level.

The analysis in Table 3.4 indicates that my core list has complete coverage over three established core vocabulary lists for historical linguistics: the Swadesh list, Dogolpolsky list, and Leipzig–Jakarta list. This is not surprising: from Table 3.3, we see that many of



(a)

(b)

Figure 3.8: Overlap in core vocabulary lists; (a) compares existing lists, (b) compares existing lists with my own Core Vocabulary list.

these words are indeed Swadesh words. What is more interesting is how my list compares to similarly-sized lists for language learning. Figure 3.8a shows that the NGSL and Oxford 3000 lists have considerable overlap with each other, but less overlap with Dale–Chall. This is possibly because both the NGSL and Oxford 3000 are largely corpus-based, while Dale–Chall is manually curated. In Figure 3.8b, we see that my list covers a little over half of each of the other lists, meaning that there are roughly 1300 words that experts have deemed important for learners that are not commonly found in dictionaries. Conversely, there are roughly 1000 words that lexicographers have deemed important for entry into dictionaries but are not found in language learning lists. What kind of words are these?

In terms of words contained in my core vocabulary but excluded from other lists, I first examine the top ten words, along with their rank in the list, that are not present in any language learning list are: 129 *human being*, 181 *mosquito*, 210 *left hand*, 342 *urine*, 355 *crocodile*, 370 *vein*, 378 *buttock*, 401 *armpit*, 422 *buttocks*, 423 *excrement*. *Human being* shares translations with *human* and *man*, which occur higher in the core list; the same is for *left hand* and *left*. The other words are animals (mosquito, crocodile), and body parts or functions, which also occur in other core lists but might not be relevant for a language learner.

To examine the differences between my core vocabulary list and other lists, I first group the core words into topics based on the topic dictionaries in the Oxford Learner's Dictionary.[15] Table 3.5 presents the top few topics whose words my list contains but other lists do not. These topic dictionaries are not comprehensive, so these counts are underestimates. Nevertheless they give an indication of the types of words missing from language learning lists.

My core list notably contains roughly 160 country names and their adjectival forms (e.g. *Spain* and *Spanish*) not present in the other language learning lists. In an increasingly interconnected society, knowledge of such proper nouns is useful for reading or translating modern text, especially on the web. Many body parts, animals, and family words exist in my list but are missing from existing lists. One explanation is that these lists are mainly for English language learners. Other cultures may place more importance on such topics, and thus knowledge of these terms would be more important for learners of those languages. For example, familial relationships are an important part of Asian cultures, and Asian languages are known for having many specific kinship terms that do not exist as a

[15] https://www.oxfordlearnersdictionaries.com/us/topic/

Topic	#	Example Words
Country	68	Europe, France, French, Spanish
Body	66	abdomen, belly, palm, wrist, nostril
Animal	55	beetle, mosquito, moth, louse, fowl
Family	42	sibling, stepfather, father-in-law, adolescent
Food	30	tasty, herb, acid, garlic
Other		wisdom, noble, merchant, murderer, funeral

Table 3.5: Examples of words in the Core Vocabulary that do not appear in other major core vocabulary lists.

single word in English.

My list contains 112 multiword concepts not present in language learning lists. Along with their associated rank, these include

- multiword expressions (MWEs) and questions (2828 *a lot*, 512 *how many*)
- phrasal verbs (180 *lie down*, 391 *look for*)
- infinitival phrases (532 *be alive*, 1315 *be born*)
- kinship terms (575 *older brother*, 754 *mother-in-law*)
- other multiword nouns (129 *human being*, 1157 *day before yesterday*)

While almost all lists contain a MWEs constituent words (e.g. *day*, *before*, and *yesterday*), a language may not have a single word for the concept of *day before yesterday*. The presence of these MWEs in the core lists highlights the deficiencies of relying on English lists.

For the non-English language lists I examined, the core vocabulary exhibits over 60%

coverage over these lists (Table 3.4). As expected, a few concepts that the core list does not include are culture specific (e.g. for Chinese: *Chinese chess*, *tai chi*, *Beijing*; for Russian: *Leningrad*, *St. Petersburg*, *Soviet*). As observed with the other lists, a large portion of missed concepts (37% for Chinese, 15% for Russian) are multiword concepts (e.g. *can't help but*, *in total*, *of course*). I noticed that many of these phrasal concepts are not content words, which usually have high representation in dictionaries and thus rank highly in my core vocabulary. Anecdotally, proficient usage of adverbs can give the impression of fluency in a foreign language even when knowledge of nouns and verbs is lacking, which might have lead to their inclusion in these language learning lists.

3.2.4 Coverage

I also examine coverage of the core vocabulary list on various corpora which span a wide range of sizes and domains. Note that while these corpora are comprised of English text, I use them not as corpora of words but concepts that are universal across languages and cultures.

3.2.4.0.1 Bible

The Bible is perhaps the most widely translated document in the world. Because of this fact, the Bible can be a useful resource for starting a dictionary in a low-resource language when other resources do not exist. I use the New Simplified English edition which contains both the Old and New Testament.

3.2.4.0.2 UDHR

The Universal Declaration of Human Rights is also a widely translated document. It is considerably smaller than the (already small) Bible.

3.2.4.0.3 British National Corpus (BNC)

(Leech, Rayson, et al., 2014) A multi-domain corpus of written and spoken British English from the late 20th century. I use words with a frequency above 800.

3.2.4.0.4 American National Corpus v2 (ANC)

(Ide and Macleod, 2001) A similar multi-domain corpus. It also contains web-domain text like emails and tweets, which are not included in the British National Corpus. I remove words that occur only once.

3.2.4.0.5 Google N-Grams Corpus (GNG)

(Michel et al., 2011) Google has scanned millions of books and computed frequency statistics per year. I use unigram frequencies from the 2012 version, accumulated over all years.

Coverage on a type and token basis are presented in Table 3.4. I compare against other lists by truncating the core vocabulary list to match the size. I remove proper names using a heuristic if it does not appear in lowercase in the text. I also exclude hapaxes (words that appear only once) from the Bible, and truncate the frequency lists over the larger

	Core-100		Swadesh 100		Core-8414		NGSL		Core-2995		Oxford	
	Type	Token	Type	Token	Type	Token	Type	Token	Type	Token	Type	Token
Bible	0.011	0.069	0.011	0.077	0.40	0.65	0.43	0.69	0.22	0.57	0.23	0.59
UDHR	0.025	0.034	0.036	0.026	0.68	0.62	0.78	0.69	0.43	0.51	0.67	0.63
BNC	0.017	0.055	0.017	0.067	0.71	0.92	0.56	0.94	0.34	0.73	0.51	0.94
ANC	0.010	0.048	0.009	0.053	0.35	0.58	0.51	0.66	0.17	0.45	0.27	0.56
GNG	0.010	0.049	0.010	0.059	0.41	0.78	0.54	0.89	0.19	0.61	0.28	0.75

Figure 3.9: Coverage of lists over various corpora. The number of types and tokens for each corpus is in Table 3.6. Comparisons are only valid between same size lists, i.e. between columns 1 and 2, 3 and 4, and 5 and 6.

corpora, the sizes of which are shown in Table 3.6. To interpret Figure 3.9, we see for example that the top 2995 core vocabulary list gives 22% type and 57% token coverage over the Bible, using 1905 core vocabulary words. This means knowing roughly 2/3 of the core list allows one to read roughly 2/3 of the Bible, an impressive figure. While the NGSL and Oxford have higher coverage over these corpora, this is due to the fact that these lists were constructed in part based on frequency in such corpora. Nevertheless, my multilingual dictionary-based core list only trails slightly behind in coverage relative to other English core lists, indicating that over a thousand lexicographers' stamp of approval across languages tends to work well for specific languages, such as English.

If my core list has high coverage over existing corpora, a natural question is: why not use the corpora themselves as the basis? Large, diverse corpora are hard to find for low-resource languages. Using the Bible, with translations into thousands of languages, as the sole corpus for a language skews the vocabulary to a specific domain and limits the usefulness of the core vocabulary list. The intent of this project is to create a universally applicable core vocabulary list where knowledge of these concepts in any language will enable the comprehension of text across a variety of domains.

Corpus	Types	Tokens
Bible	8,674	790K
UDHR	197	1,773
BNC	5,464	62M
ANC	10,000	20M
GNG	10,000	341B

Table 3.6: Corpus sizes

3.3 Conclusion

In this chapter, I present Yawipa, an extensible, comprehensive Wiktionary parser that improves over several existing parsers in terms of coverage and normalization. My innovations include extracting translations from definitions and etymology glosses, and extracting pronunciations from five non-English editions of Wiktionary, which combined with pronunciations from the English edition, comprises over 5.3 million IPA pronunciations, the largest pronunciation lexicon of its kind. Using this data, I perform experiments on predicting stress and syllable markers, and develop a new visualization technique to quantify syllabification in IPA across a language. My extracted dataset is a unique comparable corpus annotated from multiple sources with many types of data useful for downstream tasks.

To support my dictionary induction efforts, I propose a new functional definition and construction method for core vocabulary sets based on the relative coverage of a target concept in thousands of bilingual dictionaries. My core vocabulary lists derived from dictionary consensus achieves high overlap with existing widely-utilized core vocabulary lists, which are targeted at applications such as first and second language learning or field linguistics. In-depth analysis illustrates multiple desirable properties of this newly proposed core vocabulary set, including their non-compositionality. I argue that this core vocabulary set should be prioritized for elicitation when creating new dictionaries for low-resource languages for multiple downstream tasks including machine translation and language learning, which are pursued in the following chapters.

Chapter 4

Compositional and Lexical Relation Models

In the next two chapters, I present models and algorithms for dictionary induction of low-resource languages. Using no target language resources except for a small bilingual dictionary, these methods exploit the vast resources of many other languages to translate and predict missing dictionary entries in a low-resource language.

This chapter deals with a class of word formation models for concepts that have a known probabilistic pathway for being realized in a specific language. For example, in many languages, the word for HOSPITAL is a combination of the word for SICK and the word for HOUSE (Table 4.1). Danish word for hospital, *sygehus* is composed of *syg* 'sick' and *hus* 'house'. My models learn this as a language-universal recipe: HOSPITAL = SICK + HOUSE. Compositional word formation comprises not only compound words and some instances of inflectional and derivational morphology, as well as some multi-word expressions.[1]

These types of models also allow us to model semantic change during word formation, specifically how a translation for a concept in one language may actually be a valid translation of a related concept. I call this translation via lexical relations. For example, the English word *watermelon* is translated into Italian as *cocomero*, which can also mean 'cucumber' (*cocomero* originated from the Latin *cucumis* 'cucumber'). Both models of compositionality and lexical semantics across languages can be used to predict translations of words in a low-resource language. Because these models share similar computational approaches, I combine the discussion of these models into a single chapter.

4.1 Compositional Word Formation

Compounding is one of the most common and productive methods of word formation across the world's languages (Denning, Kessler, and Leben, 2007). Many common words are compounds, e.g. English *light·house* or *air·port.* Nevertheless, the derivational processes and semantics of compound words can be quite complex.

Consider the semantic concept *hospital*, which can be realized via compound morphology in a remarkable diversity of semantic compositions, as shown in Table 4.1. There are clearly a wide variety of semantic associations constituting this concept (e.g. sick/disease + house/place/institution), a variety of constituent orders (e.g. sick+house vs. house+sick)

[1] My work does not apply to non-concatenative morphology, such as in Semitic languages. I leave this for future work.

Lang.	Compound	Literal Semantics
nld	ziekenhuis	sick + house
nor	sykehus	sick + house
hun	kórház	disease + house
epo	malsanuelejo	sick + place
msa	rumah sakit	house + sick
zho	病院	disease + institution

Table 4.1: Realizations of the concept of *hospital* in several languages.

and potentially a variety of compounding processes beyond simple concatenation (e.g. *sykehus* in Norwegian can be analyzed as *syk* 'sick' + *e* + *hus* 'house'). In linguistics, *syk* and *hus* are referred to as *stems* of the compound *sykehus*. We may also refer to these as *components*, *constituents*, or simply, *parts*.

In this chapter, I present a massively cross-linguistic computational model of both compound morphology compositional processes and compound semantics. This model not only derives an analysis of the compounding process and semantics of compounds within a *single* language, as with much prior related work (see Section 2.2 for prior work), but does so via a joint model across essentially all the world's languages with adequate dictionary resources. This is an unprecedentedly large scale for this class of research, and with significant additional synergistic multilingual power. My compounding model handles not only compounds in the traditional sense (i.e. the combination of independent words), but also derivational morphology (*quickly*, *pretest*) as well as multiword expressions (*fire truck*, *pomme de terre*).

I successfully apply this model to the downstream task of predicting novel translations of compound words, both to English (e.g. *kórház* → *disease+house* → *hospital*) and from

English (e.g. *hospital* → *disease+house, sick+place, etc.* → *kórház etc.*), with valuable applications for translation dictionary expansion and out-of-vocabulary handling in machine translation, again on this uniquely large multilingual scale.

Specifically, this model enables two tasks: *compound analysis* and *compound generation*. In the analysis direction, the goal is to identify the translation of a compound word, by first correctly identifying the word's constituent parts (compound splitting) and then applying a multipath gloss translation algorithm to identify the English translation. In the generation direction, the goal is to predict translations of a given concept, assuming the realization of that concept in a target language is a compound word. Compared with much existing work (see Section 2.2), which focuses on a single language pair or a handful of languages, my model handles on the order of hundreds of languages and is especially applicable for low resource languages for which we do not have much available corpora.

I evaluate the different components of my model on three tasks: compound splitting, compound translation (into English), and compound generation (from English to another language), holding out test words from the dictionary so that they are unseen by the model.

This chapter includes some work originally published in Wu and Yarowsky (2018c). In conjunction with this paper, I released a novel and uniquely large-scale 329-language, 21,000+ example dataset[2] of these compound morphological analyses and their associated compositional and compound translations. This is a valuable resource for training models

[2] github.com/wswu/worcomal

for derivational morphology processes and compound semantics on this massively multilingual scale, with direct application to machine translation.

4.1.1 Compound Discovery from Lexical Resources

While most existing studies (see Section 2.2) require some form of corpus or parallel bitext, I start with only a collection of bilingual dictionaries. Specifically, I use foreign-English translation dictionaries extracted from the open-source dictionary Wiktionary[3] using Yawipa (Wu and Yarowsky, 2020a), my Wiktionary extraction tool (presented in Chapter 3). I extracted translations annotated with the `tr` tag, as well as definition translations and translations from glosses. The major assumption is that these translations contain both substantial examples of compounding in each language (e.g. *sykehus* (Norwegian) = *hospital* (English)) as well as translations of the constituents of these compounds (e.g. *syk* = *sick* and *hus* = *house*). Using these dictionaries, I develop a multi-iteration method for discovering compound translation models motivated across multiple languages that can be used to analyze and construct new compound words that do not exist in available dictionaries.

I extracted from Wiktionary a translation dictionary comprising over 3.1 million words (3.9 including English) across 7.944 languages (as measured by unique ISO 639-3 codes). This translation dictionary contains 5.4 million foreign-English translation pairs. Because this is a foreign-English translation dictionary, I add self-translations (i.e. English-English)

[3] www.wiktionary.org

Lang	Word	Translation	Literal Gloss
fin	rakennustyö	construction	construction + work
nld	ziekenhuis	hospital	sick + house
dan	folk**e**afstemning	referendum	people + vote
nob	informasjon**s**tecknologi	information technology	information + technology
deu	Meuchelnmorder	assassin	assassinate + killer
esp	cantoautor	singer-songwriter	singing + author

Table 4.2: Compounding methods: concatenation, epenthesis, and elision. For epenthesis, the added character is bolded. For elision, the character deleted from the first morpheme is in small font.

for all English translations that do not yet exist in the dictionary, for a total of 6.2 million translation pairs, in order that English can be considered a "foreign" language whose words have an English translation. I also relabel all Mandarin Chinese (cmn) words to use the Chinese macrolanguage zho (~45k words), in order to unify the two and not double count Mandarin words.[4]

4.1.2 Compound Splitting for Automatic Compound Discovery

To discover potential compounds from the dictionary, I perform compound splitting for the compounding mechanisms described in Table 4.2. Existing studies (Koehn and Knight, 2003; Garera and Yarowsky, 2008, e.g.), exhaustively split a word into all possible

[4] Then all Mandarin Chinese words are unified under a single language code. I found that some words listed under cmn did not occur in zho, but often zho words overlapped with other Chinese languages such as Cantonese (yue) and Hakka (hak), so I keep these other Chinese languages separate. This preprocessing step may also be applicable to other macrolanguage codes, but since Chinese is known for its extensive lexicon of compositional words, I felt this action was appropriate for Chinese.

Split	Valid?	Literal Translation
l+acrosse		
la+crosse	✓	the/her + stick/crosier
lac+rosse	✓	lake + bitch/vixen
lacr+osse		
lacro+sse		
lacros+se		
lacross+e		

Table 4.3: Exhaustive splitting for the French word *lacrosse*.

two constituent parts (Table 4.3) and mark the word as a possible compound if both parts occur in a corpus of the word's respective language. Since we may not have corpora available in some languages, I employ dictionaries in place of a corpus.[5] When splitting words, Garera and Yarowsky (2008) limit each component part to be at least three characters in order to avoid components being inflections. My models do not have this restriction, because I would like the models to handle inflectional morphology as a compositional word formation process. In addition, inflectional and derivational affixes often exist as separate entries in Wiktionary that have their own translations. This compound discovery step resulted in 906K potential compound words in 557 languages.

I repeat this compound discovery process for another methods of compound splitting that handle epenthesis, the insertion of a sound between two morphemes. This is a common process in many languages. For example, the Danish word for "referendum", *folkeafstemning*, is a compound of *folk* "people" and *afstemning* "vote" with the addition of an *e* between them. I follow existing work (Koehn and Knight, 2003; Garera and Yarowsky,

[5] Note that this compound discovery step technically only requires a wordlist, not an entire dictionary.

2008) by splitting a word into three parts, where the second part is a "filler" or "glue" between the two constituent parts. I restrict the length of this filler segment to be at most 1/3 the length of the entire word. Note that this filler may be a space or may even contain multiple words, allowing this process to discover multi-word expressions. This compound splitting method resulted in 1.3 potential compounds.

In some cases, instead of concatenating two morphemes or concatenating with epenthesis, the first component may be elided with the second. That is, characters from the end of the first morpheme are deleted before concatenation. For example, the Spanish word *cantautor* "singer-songwriter" is composed of *canto* "singing" and *autor* "author", with the *o* in *canto* deleted. In a third compound splitting method, I allow for elisions up to two characters.

I also propose a new *fuzzy middle* method for compound splitting that exactly matches the beginning and end of the compound but allows for some variation at the site of concatenation. Recall that for simple concatenative compounds, I split a word into all possible two parts and consider the word a potential compound if both component parts occur in the dictionary. In contrast, the fuzzy middle algorithm truncates each component part by removing the last character of the left part and the first character of the right part, looking up these truncated parts in the dictionary, and considering words that contain up to two character additions at the end of the left part, and beginning of the right part, respectively. This allows for up to two character deletions and four character insertions between the two morphemes, effectively combining the concatenation, epenthesis, and elision mechanisms. The following pseudocode illustrates this approach:

```
function fuzzy_middle(word)
    for (left, right) in segment(word)
        trunc_left = left[1 : length(left)-1]
        trunc_right = right[2 : length(right)]
        for L in dictionary that starts with trunc_left
            for R in dictionary that ends with trunc_right
                add L+R to the potential compound list
            end
        end
    end
end
```

To enable efficient search for words that start with the truncated left component and end with the truncated right component, I utilize a trie, an efficient data structure for searching prefixes. I construct two tries, a forward trie to search for the truncated left component, and a backward trie to search for the reversed characters of the truncated right component.

4.1.2.1 Evaluation of Compound Splitting

Compound splitting is not the main focus of this work. However, as it is a step in the compound discovery pipeline, I briefly present an evaluation of the four aforementioned compound splitting algorithms on four datasets extracted from Wiktionary. I use a gold standard dataset of compounds, affixal words, prefixal words, and suffixal words, which were extracted from Wiktionary etymology annotations `com`, `af`, `pre`, and `suf`, respectively.[6] For each of these four categories, I randomly select up to 50 words from each

[6] None of these categories overlap. Though it may seem that `af` subsumes `pre` and `suf`, affixal words may be formed with both a prefix and a suffix, or may contain more than two morphemes.

language so long as that word contains an English translation in Wiktionary. I hold out these words from the dictionary so that they are unseen by the model. I evaluate whether these splitting algorithms can successfully recover the ground truth splits as annotated in Wiktionary for compounds (com), affixal words (af), prefixal words (pre), and suffixal words (suf). A summary of results is in Table 4.4. I evaluate three metrics: 1-best accuracy, 10-best accuracy (is the gold in the top 10 model predictions), and mean reciprocal rank.

I find that many compound words can be discovered by simply splitting a string into all possible two parts and performing a dictionary lookup on each part. In fact, the simple concatenative splitting algorithm can successfully split over a third of all unseen compounds and unseen prefixal words across all 349 languages in the test set. This proposed fuzzy middle approach improves on the compound splitting of the other mechanisms.

From the overall accuracies, one may wonder why these accuracies seem unusually low compared to recent compound splitters, which often report accuracies above 80% (e.g. Ziering and Plas, 2016; Krotova, Aksenov, and Artemova, 2020). First, most studies on compound splitting evaluate on German, which is a high-resource language with copious amounts of available training data. This study is evaluated across over 300 languages, most of which are low-resource.

Second, many words in this test set are composed of more than two components, especially affixal words (af). The splitting methods here are designed to handle compounds with two components. Third, due to the low-resource nature of many languages in the

Dataset	Splitter	Acc1	Acc10	MRR
af	concat	.174	.177	0.0013
com	concat	.296	.298	0.0008
pre	concat	.372	.380	0.0035
suf	concat	.124	.128	0.0019
af	epen	.063	.066	0.0015
com	epen	.140	.145	0.0024
pre	epen	.018	.020	0.0010
suf	epen	.011	.014	0.0014
af	elis	.054	.094	0.0147
com	elis	.039	.056	0.0062
pre	elis	.103	.288	0.0557
suf	elis	.055	.079	0.0100
af	fuzzy	.248	.333	0.0285
com	fuzzy	.429	.537	0.0366
pre	fuzzy	.359	.460	0.0340
suf	fuzzy	.176	.269	0.0306

Table 4.4: Compound splitting results, evaluated with 1-best accuracy, 10-best accuracy, and mean reciprocal rank.

test set, even if the splitting algorithm identifies the correct split point, the decomposition will not be obtained if any component does not exist in Wiktionary.

Finally, for evaluation, I ignore hyphens that occur at the beginning and ends of component parts to account for affixes. However, I do not ignore capitalization and diacritics, because I take the data in Wiktionary as ground truth. This unfairly penalizes the model against certain languages that employ capitalization or diacritics. For example, German *Gegensatz* = *gegen-* + *Satz* is not correctly analyzed, because *Gegen* (capitalized) does not exist in the dictionary. Similarly, Old English *eaþmodlic* = *ēaþmōd* + *-līċ* is not correctly analyzed by any of the compound splitting mechanisms. However, certain cases, for example, Old English *hamsteall* = *hām* + *steall*, are analyzable by the fuzzy middle algorithm, which treats *ham* as a fuzzy match of *hām*.

An example of a particularly problematic example touching on all three points above is the Crow word for "Easter": *Alihkaluusúu*, which is made up of *ala-* 'when' + *ihká* 'egg' + *duusúu* 'they eat'. This is a word exhibiting differences in capitalization as well as diacritics, is a three-component compound, and furthermore, the second component *ihká* does not exist in Wiktionary.

I leave the handling of these issues to future work. Nevertheless, even with low accuracy on this specific compound splitting task, these methods allow us to automatically obtain a large set of high-quality compounds (after filtering, described later) for training a multilingual compounding model.

4.1.3 Multilingual Compound Model

Using potential compounds acquired using the concatenative and epenthesized splitting algorithms described above, I develop an automatic approach to learn a universal model of compounding. This model associates a probabilistic "recipe" with every language-independent concept, with which I can analyze and generate compound words. Throughout this chapter, I will use the concept of *hospital* as a running example. This is an interesting illustrative example since it is not a compound word in English, but occurs as a compound in many other languages.

I begin by considering the compounds of two components, where both components both exist in the dictionary (e.g. kór+ház = SICK+HOUSE). I collect all words with the

HOSPITAL =	ill (58)	+	house (62)
	sick (53)		home (24)
	disease (28)		building (19)
	medicine (25)		place (19)
	patient (23)		institution (14)
	illness (21)		encasing (10)
	doctor (17)		residence (7)
	house (15)		casing (7)
	physician (11)		A house (6)
	building (10)		beat (6)

Recipe	Lang	Word	Segmentation
ill+house	swe	sjukhus	sjuk|hus
ill+house	tgk	касалхона	касал|хона
ill+house	tgk	беморхона	бемор|хона
ill+house	zho	病厝	病|厝
ill+house	ovd	siuokstugu	siuok|stugu
ill+house	afr	siekehuis	siek|e|huis
ill+house	dan	sygehus	syg|e|hus
ill+house	nld	ziekenhuis	ziek|en|huis
ill+house	nld	ziekenhuis	zieke|n|huis
ill+house	nno	sjukehus	sjuk|e|hus
ill+house	ota	خستهخانه	خسته||خانه
ill+house	mak	balla’ garring	balla’| |garring
house+sick	ind	rumah sakit	rumah| |sakit
house+sick	msa	rumah sakit	rumah| |sakit
sick+house	dan	sygehus	syge|hus
sick+house	nld	ziekenhuis	zieken|huis
disease+house	myv	ормакудо	орма|кудо
disease+house	hun	kórház	kór|ház
house+medicine	jra	sang ia jrao	sang| ia |jrao
medicine+house	que	jampina wasi	jampina| |wasi
medicine+house	bod	སྨན་ཁང	སྨན|་|ཁང
house+medicine	tir	ቤት ሕክምና	ቤት| |ሕክምና
illness+house	nno	sjukehus	sjuke|hus
house+illness	tpi	haus sik	haus| |sik
doctor+house	ang	lǣċehūs	lǣċe|hūs

Figure 4.1: Compounding recipe for the concept HOSPITAL learned across all languages. A small portion of the training compounds are shown to the right. The numbers in parentheses indicate the number of compounds whose components translated to the specified word.

same English translation (e.g. *hospital*) that are potential compounds decomposable via concatenation, as described above. For each potential compound, I translate its component parts and accumulate counts of the frequency of each part's translation, forming a probability distribution of component translations for the left and right components of the language-independent concept of HOSPITAL (Figure 4.1).

For any given concept, the semantic ordering of the components in the realization of this concept into a specific language will often vary depending on the language. For example, compound words for the concept *hospital* have different component ordering in different languages:

Dutch:	ziekenhuis	‘ill’+‘house’
Malay:	rumah sakit	‘house’+‘sick’

To account for this variation in ordering, I flip the ordering of the word when con-

Left		Right	
sick	8	house	7
disease	7	home	6
house	6	institution	4
home	5	place	4
ill	4	court	3

Table 4.5: A simplified (for illustration purposes) distribution of component language counts for "hospital" before correcting for ordering.

structing the compositional recipe to match the universal majority ordering. I define the *translational entropy* of a compound model as the sum of the entropy of the component translations on each side, respectively:

$$\text{TE}(\text{concept}) = \text{H}(\text{left translations}) + \text{H}(\text{right translations}) \tag{4.1}$$

where $\text{H}(X) = -\sum_i p(x_i)\ log\ p(x_i)$ is the familiar formula for entropy in information theory. For each compound word, I mark it as "flipped" if flipping the order of the components decreases the overall translation entropy. This process reduces noise in the language-universal model of component part translations.

For a worked out example, consider the simplified distribution of translations in Table 4.5, where the translation counts for the Malay word *rumah sakit* add 1 to the left count for 'house' and 1 to the right count for 'sick' (shown in orange). The translation entropy is thus

$$\mathrm{H}(\left[\frac{8}{31},\frac{7}{31},\frac{6+1}{31},\frac{5}{31},\frac{4}{31}\right])+\mathrm{H}(\left[\frac{7}{25},\frac{6}{25},\frac{4}{25},\frac{4}{25},\frac{3}{25},\frac{1}{25}\right]) \tag{4.2}$$

$$=2.28+2.41 \tag{4.3}$$

$$=4.67 \tag{4.4}$$

Suppose we now flip the ordering of the components in the Malay word *rumah sakit*, such that the component translations sick+house becomes house+sick. Then the translational entropy for this recipe becomes:

$$\mathrm{H}(\left[\frac{8+1}{31},\frac{7}{31},\frac{6}{31},\frac{5}{31},\frac{4}{31}\right])+\mathrm{H}(\left[\frac{7+1}{25},\frac{6}{25},\frac{4}{25},\frac{4}{25},\frac{3}{25}\right]) \tag{4.5}$$

$$=2.27+2.23 \tag{4.6}$$

$$=4.50 \tag{4.7}$$

This flipping operation brings *rumah sakit* in line with the universal ordering of HOSPITAL=ill/sick+house/home, thus improving the recipe for HOSPITAL. The model iterate through each compound associated with the HOSPITAL concept and perform this flipping operation if it reduces the recipe's translational entropy.

Finally, I employ this component part translation distribution to filter out bad compound analyses used to generate this distribution. In a second iteration of model construction, I use only potential compounds whose component translations both have a frequency count greater than 1. This criterion effectively removes bad compound splits such as the Dutch *hospitaal* (decomposed as hospita 'landlady' + al 'even'), thus refining the "recipe" of *hospital.* The component translation distributions for each semantic concept are stored in JSON format for future use.

4.1.3.1 Compound Model Examples and Analysis

In the following pages, I show several examples of universal compounding models learned across all the languages available in the training dictionary. Some decompositions are italicized, indicating that they are not scored highly by the recipe and would be filtered out using the compound score described later. Commentary for each of the recipes is presented in the caption of each figure.

YOLK =		+	
	egg (81)		yellow (23)
	yellow (48)		red (15)
	edge (18)		yolk (14)
	testicle (11)		egg (12)
	ball (9)		pocket (10)
	ovum (7)		plum (6)
	gel (6)		ten (6)
	bead (6)		heart (6)
	arête (6)		diminutive suffix (5)
	roe (5)		diminutive (5)

Recipe	Lang	Word	Segmentation
egg+yellow	afr	eiergeel	eier|geel
egg+yellow	nld	eigeel	ei|geel
egg+yellow	epo	ovoflavo	ovo|flavo
egg+yellow	deu	Eigelb	Ei|gelb
egg+yellow	jpn	卵黄	卵|黄
egg+yellow	jpn	蛋黄	蛋|黄
egg+yellow	ltz	Eegiel	Ee|giel
egg+yellow	zha	gyaeqhenj	gyaeq|henj
egg+yellow	zho	蛋黄	蛋|黄
yellow+egg	ara	أَصْفَر الْبَيْض	أَصْفَر| |الْ|بَيْض
yellow+egg	ind	kuning telur	kuning| |telur
yellow+egg	msa	kuning telur	kuning| |telur
yellow+egg	roh	mellen d'ov	mellen| d'|ov
yellow+egg	roh	mellen d'iev	mellen| d'|iev
yellow+egg	roh	melen d'ov	melen| d'|ov
yellow+egg	roh	mellan d'öv	mellan| d'|öv
yellow+egg	roh	gelg d'öv	gelg| d'|öv
yellow+egg	wln	djaene d'oû	djaene| d'|oû
egg+red	lao	ໄຂ່ແດງ	ໄຂ່|ແດງ
egg+red	shn	ၶႆႇလႅင်	ၶႆႇ| လႅင်
egg+red	tha	ไข่แดง	ไข่|แดง
red+egg	ita	rosso d'uovo	rosso| d'|uovo
egg+yolk	nld	eidooier	ei|dooier
egg+yolk	nld	eierdooier	eier|dooier
egg+yolk	fao	eggjareyði	eggja|reyði
egg+yolk	hun	tojássárgája	tojás|sárgája
egg+yolk	nld	eierdooier	ei|er|dooier
egg+yolk	fao	eggjareyði	egg|ja|reyði
egg+yolk	fin	munankeltuainen	muna|n|keltuainen
egg+yolk	isl	eggjarauða	egg|ja|rauða

Figure 4.2: Recipes for YOLK. While 'egg yellow' is the dominant recipe, 'egg red' also occurs in a few languages. The color of the egg yolk is determined mainly by the hen's diet, but we will leave it to other researchers to determine whether the hens of Southeast Asia and Italy have significantly different diets than hens in the rest of the world.

CORONAVIRUS =		+	
	crown (43)		virus (119)
	corona (33)		computer virus (19)
	choir (15)		Russian (8)
	us (13)		bug (7)
	heart (9)		viral (3)
	chorus (8)		corona (3)
	krone (8)		
	króna (8)		
	COVID-19 (8)		
	wreath (6)		

Recipe	Lang	Word	Segmentation
crown+virus	bul	коронави́рус	корона|ви́рус
crown+virus	cat	coron‘avirus	corona|virus
crown+virus	est	koroonaviirus	koroona|viirus
crown+virus	hun	koronavírus	korona|vírus
crown+virus	isl	kórónaveira	kóróna|veira
crown+virus	gle	coróinvíreas	coróin|víreas
crown+virus	ita	coronavirus	corona|virus
crown+virus	oci	coronavirus	corona|virus
crown+virus	pol	koronawirus	korona|wirus
crown+virus	rus	коронави́рус	корона|ви́рус
crown+virus	spa	coronavirus	corona|virus
crown+virus	ukr	коронаві́рус	корона|ві́рус
crown+virus	bul	коронавирус	корона|вирус
crown+virus	rus	коронавирус	корона|вирус
crown+virus	ukr	коронавірус	корона|вірус
crown+virus	mon	титэм вирус	титэм| |вирус
crown+virus	cym	coronafirws	coron|a|firws
crown+virus	cym	coronafeirws	coron|a|feirws
crown+virus	zho	冠狀病毒	冠|狀|病毒
crown+virus	hye	թագավարակ	թագ|ա|վարակ
crown+virus	zho	冠状病毒	冠|状|病毒
corona+virus	nld	coronavirus	corona|virus
corona+virus	fin	koronavirus	korona|virus
corona+virus	ind	koronavirus	korona|virus
corona+virus	jpn	コロナウイルス	コロナ|ウイルス
corona+virus	kor	코로나바이러스	코로나|바이러스
corona+virus	kor	코로나비루스	코로나|비루스
corona+virus	por	coronavírus	corona|vírus
corona+virus	eng	coronavirus	corona|virus
virus+corona	ind	virus korona	virus| |korona

Figure 4.3: Recipes for CORONAVIRUS.

MAN =	man (135) male (116) husband (46) son (29) -th (28) person (20) baron (20) people (19) -eth (ordinal number suffix)) (16) boy (13)	+	man (94) human (80) person (61) human being (61) people (51) child (30) son (24) male (19) -er (19) character (14)

Recipe	Lang	Word	Segmentation
man+man	tha	ผู้ชาย	ผู้|ชาย
man+man	isl	karlmaður	karl|maður
man+man	zho	丈夫	丈|夫
man+man	ang	maguþegn	magu|þegn
man+man	zho	男人	男|人
man+man	zho	男子漢	男子|漢
man+man	zho	士人	士|人
man+man	zho	男士	男|士
man+man	zho	男丁	男|丁
man+man	chv	арçын	ар|çын
man+man	non	karlmaðr	karl|maðr
man+man	zho	丁男	丁|男
man+man	jpn	男の人	男|の|人
man+man	zho	男子漢	男|子|漢
man+man	zho	男仔人	男|仔|人
man+man	asm	মতা মানুহ	মতা| |মানুহ
man+man	bak	ир кеше	ир| |кеше
man+man	cic	hattak nakni’	hattak| |nakni’
man+man	chv	ар çын	ар| |çын
man+man	kaz	ер адам	ер| |адам
man+man	kaz	ер кісі	ер| |кісі
man+man	mon	эр хүн	эр| |хүн
man+man	tat	ир кеше	ир| |кеше
man+man	tur	er kişi	er| |kişi
man+man	uig	ئەر كىشى	ئەر| |كىشى
man+man	uzb	erkak kişi	erkak| |kişi
man+man	sah	эр киһи	эр| |киһи
human+man	mnw	ၞိဟ်ဗြ	ၞိဟ်|ဗြ
man+human	tyv	эр кижи	эр| |кижи
male+man	lao	ຜູ້ຊາຍ	ຜູ້|ຊາຍ

Figure 4.4: Recipes for MAN. This concept is ambiguous, because *man* can refer to ‘human’ or ‘adult male human’. These compositional words follow the latter interpretation, which is not evident from the recipe man+man but can be seen in the examples, e.g. 男 ‘male, man’ in Chinese and Japanese, and *er*/*ep* ‘male, man, husband’ in Turkic languages

.

ASTRONAUT =	space (43) outer space (21) cosmos (16) universe (11) eaves (8) room (7) celestial body (7) the universe (5) space flight (5) number (4)	+	man (13) human being (10) pilot (9) -er (8) human (7) person (6) sailor (5) airman (5) guy (5) people (5)

Recipe	Lang	Word	Segmentation
space+man	tha	มนุษย์อวกาศ	มนุษย์|อวกาศ
space+man	zho	太空人	太空|人
space+man	ara	رَجُل فَضَاء	رَجُل| |فَضَاء
space+man	cor	den efanvos	den| |efanvos
space+man	fao	rúmdarmaður	rúm|dar|maður
space+man	kaz	ғарышкер	ғарыш|к|ер
space+human being	kor	우주인	우주|인
space+human being	tha	มนุษย์อวกาศ	มนุษย ์|อวกาศ
space+pilot	fin	avaruuslentäjä	avaruus|lentäjä
space+pilot	jpn	宇宙飛行士	宇宙|飛行士
pilot+space	tha	นักบินอวกาศ	นักบิน|อวกาศ
space+pilot	zho	宇宙飛行員	宇宙|飛行員
space+pilot	zho	宇宙飞行员	宇宙|飞行员
space+sailor	nld	ruimtevaarder	ruimte|vaarder
space+sailor	hun	űrhajós	űr|hajós
space+sailor	nld	ruimtevaarder	ruim|te|vaarder
space+automobile	epo	kosmonaŭto	kosmo|n|aŭto
space+female	epo	kosmonaŭtino	kosmo|naŭt|ino
space+cow	jpn	うちゅうひこうし	うちゅう|ひこ|うし
space+chief	gle	spásaire	spás|aire
space+woman	cor	benyn efanvos	benyn| |efanvos
space+of the	nld	ruimtevaarder	ruimte|vaar|der
space+tee	epo	kosmonaŭto	kosmo|naŭ|to
space+fortuneteller	hun	űrhajós	űr|ha|jós
space+exercise	kor	우주비행사	우주|비|행사
space+exercise	jpn	うちゅうひこうし	うちゅう|ひ|こうし
space+traveller	nob	romfarer	rom|farer
space+major	ara	رَائِد فَضَاء	رَائِد| |فَضَاء
space+traveller	hin	अंतरिक्ष यात्री	अंतरिक्ष| |यात्री
space+incantation	fas	فضانورد	فضا|ن|ورد

Figure 4.5: Recipes for ASTRONAUT. The dominant recipes are space+man, space+pilot, and space+sailor (as in English). Here we see several incorrect decompositions due to some characters being interpreted as filler characters.

KITCHEN =	kitchen (26)	+	house (40)
	cook (18)		room (23)
	fire (12)		home (21)
	food (9)		chamber (12)
	room (9)		en (10)
	stove (8)		household (8)
	chef (7)		building (8)
	cue (7)		hen (8)
	kitchen range (7)		shop (7)
	kitchen god (7)		(6)

Recipe	Lang	Word	Segmentation
kitchen+house	hin	रसोईघर	रसोई|घर
kitchen+house	zho	廚房	廚|房
kitchen+house	zho	灶屋	灶|屋
kitchen+house	zho	灶房	灶|房
house+kitchen	vie	nhà bếp	nhà| |bếp
cook+house	asm	ৰান্ধনিঘৰ	ৰান্ধনি|ঘৰ
cook+house	fas	آشپزخانه	آشپز|خانه
cook+house	tgk	ошпазхона	ошпаз|хона
cook+house	zho	爨室	爨|室
cook+house	asm	ৰান্ধনিঘৰ	ৰান্ধ|নি|ঘৰ
house+cook	tpi	haus kuk	haus| |kuk
fire+house	cim	bôarhaus	bôar|haus
kitchen+room	jpn	厨房	厨|房
kitchen+room	mya	စားဖိုဆောင်	စားဖို|ဆောင်
food+house	kaz	асүй	ас|үй
kitchen+room	zho	灶間	灶|間
kitchen+room	zho	廚房間	廚|房間
kitchen+room	zho	廚房間	廚房|間
food+house	kaz	ас үй	ас| |үй
room+kitchen	tgl	silid-lutuan	silid|-|lutuan
kitchen+room	zho	灶披間	灶|披|間
kitchen+room	zho	廚房間	廚|房|間
stove+house	bod	ཐབ་ཚང	ཐབ|་|ཚང
thick Persian-style soup+house	tgk	ошхона	ош|хона
cooking+house	syl	ꠙꠣꠇꠊꠞ	ꠙꠣꠇ|ꠊꠞ
foot+house	asm	পাকঘৰ	পা|ক|ঘৰ
juice+house	hin	रसोईघर	रस|ोई|घर
thick Persian-style soup+house	fas	آشپزخانه	آش|پز|خانه
thick Persian-style soup+house	tgk	ошпазхона	ош|паз|хона
come+house	jpn	くりや	く|り|や

Figure 4.6: Recipes for KITCHEN. Most recipes are kitchen+house or food+house. Some recipes may have the concept also as a component, e.g. kitchen = kitchen + room. For the case of Asian languages, 厨房 = 厨 'kitchen' + 房 'room', 厨 is not a standalone word, but rather a bound morpheme that is commonly used in other kitchen-related words, e.g. 厨师 'chef' (kitchen + master)' or 下厨 'go to the kitchen to cook' (go down + kitchen).

LINGUISTICS =	language (106)	+	science (88)
	linguist (40)		knowledge (47)
	tongue (37)		-logy (41)
	speech (30)		study (33)
	linguistic (19)		-ology (29)
	matter (10)		learning (17)
	word (10)		learn (16)
	goal (8)		studies (14)
	talk (7)		-ics (9)
	clapper (7)		scholarship (8)

Recipe	Lang	Word	Segmentation
language+science	ben	ভাষাবিজ্ঞান	ভাষা|বিজ্ঞান
language+science	mya	ဘာသာဗေဒ	ဘာသာ|ဗေဒ
language+science	dan	sprogvidenskab	sprog|videnskab
language+science	epo	lingvoscienco	lingvo|scienco
language+science	fao	málfrøði	mál|frøði
language+science	fin	kielitiede	kieli|tiede
language+science	hin	भाषाविज्ञान	भाषा|विज्ञान
language+science	hun	nyelvtudomány	nyelv|tudomány
language+science	isl	málvísindi	mál|vísindi
language+science	jpn	言語学	言語|学
language+science	khm	ភាសាសាស្ត្រ	ភាសា|សាស្ត្រ
language+science	san	भाषाविज्ञान	भाषा|विज्ञान
language+science	swe	språkvetenskap	språk|vetenskap
language+science	tha	ภาษาศาสตร์	ภาษา|ศาสตร์
language+science	tur	dilbilim	dil|bilim
language+science	vol	pükav	pük|av
language+science	zho	語言學	語言|學
language+science	jpn	語学	語|学
language+science	fao	málvísindi	mál|vísindi
language+science	nob	språkvitenskap	språk|vitenskap
language+science	asm	ভাষাবিজ্ঞান	ভাষা|বিজ্ঞান
language+science	tgl	aghamwika	agham|wika
language+science	vep	kel'tedo	kel'|tedo
language+science	ces	jazykověda	jazyk|o|věda
language+science	est	keeleteadus	keel|e|teadus
language+science	kat	ენათმეცნიერება	ენა|თ|მეცნიერება
language+science	ind	ilmu bahasa	ilmu| |bahasa
language+science	slk	jazykoveda	jazyk|o|veda
language+science	tel	భాషాశాస్త్రము	భాషా|శాస్త్రము
language+science	tur	dilbilim	dil|b|ilim

Figure 4.7: Recipes for LINGUISTICS. Proof that linguistics is a science!

RAILROAD =

	+	
iron (40)		road (75)
rail (10)		way (35)
weapon (6)		path (34)
irons (4)		route (22)
ruthless (4)		street (22)
solid (4)		journey (9)
hard (4)		method (9)
firm (4)		pattern (6)
intimate (4)		type (6)
arms (4)		kind (6)

Recipe	Lang	Word	Segmentation
iron+road	aze	dəmiryol	dəmir|yol
iron+road	fin	rautatie	rauta|tie
iron+road	mya	သံလမ်း	သံ| လမ်း
road+iron	khm	ផ្លូវដែក	ផ្លូវ|ដែក
iron+road	khm	អយបថ	អយ|បថ
iron+road	zho	鐵路	鐵|路
iron+road	zho	鐵道	鐵|道
iron+road	zho	鐵墿	鐵|墿
iron+road	bod	ལྕགས་ལམ	ལྕགས|་|ལམ
iron+road	kaz	темір жол	темір| |жол
iron+road	zho	鐵枝路	鐵|枝|路
road+iron	spa	camino de hierro	camino| |de hierro
road+iron	spa	camino de hierro	camino| de |hierro
rail+road	zho	鐵枝路	鐵枝|路
rail+road	eng	railroad	rail|road
rail+road	zho	鐵枝仔路	鐵枝|仔|路
weapon+road	zho	火車路	火|車|路
weapon+road	zho	火車墿	火|車|墿
line+road	jpn	線路	線|路
train+road	zho	火車路	火車|路
train+road	zho	火車墿	火車|墿
base+road	kor	기찻길	기|찻|길
euphoria caused by narcotic intoxication+road	aze	dəmiryol	dəm|ir|yol
ferric+road	gle	bóthar iarainn	bóthar| |iarainn
via ferrata+road	ita	strada ferrata	strada| |ferrata
installment+road	ita	strada ferrata	strada| fer|rata
surely+road	vie	đường sắt	đường| s|ắt
road+construct	khm	ផ្លូវដែក	ផ្លូវ|ដែ|ក
the independent deprecated vowel+road	khm	អយបថ	អ|យ|បថ
ra+road	eng	railroad	ra|il|road

Figure 4.8: Recipes for RAILROAD.

RACISM =

	+	
race (50)		-ism (37)
ethnicity (18)		discrimination (20)
species (9)		doctrine (13)
caste (8)		ism (11)
breed (8)		ideology (8)
seed (8)		principle (7)
racist (7)		-ness (7)
skin color (7)		attention (6)
human (7)		difference (5)
type (7)		split (4)

Recipe	Lang	Word	Segmentation
race+-ism	hin	जातिवाद	जाति|वाद
race+-ism	khm	ពូជនិយម	ពូជ|និយម
race+-ism	fas	نژادپرستی	نژاد|پرستی
race+-ism	zho	種族主義	種族|主義
race+-ism	jpn	人種主義	人種|主義
race+-ism	zho	种族主义	种族|主义
race+-ism	mya	လူမျိုးရေးဝါဒ	လူမျိုး|ရေး|ဝါဒ
race+-ism	zho	種族主義	種|族|主義
race+-ism	heb	גזענות	גזע|נ|ות
race+-ism	kor	인종차별주의	인종|차별|주의
race+-ism	zho	种族主义	种|族|主义
race+discrimination	hin	नस्लभेद	नस्ल|भेद
race+discrimination	jpn	人種差別	人種|差別
race+discrimination	kor	인종차별	인종|차별
race+discrimination	zho	種族歧視	種族|歧視
race+discrimination	zho	种族歧视	种族|歧视
race+discrimination	kor	인종 차별	인종| |차별
race+discrimination	uig	ئىرقىي كەمسىتىش	ئىرق|ىي |كەمسىتىش
race+discrimination	zho	種族歧視	種|族|歧視
race+discrimination	zho	种族歧视	种|族|歧视
race+doctrine	fin	rotuoppi	rotu|oppi
race+ism	hun	rasszizmus	rassz|izmus
race+ideology	tha	คตินิยมเชื้อชาติ	คตินิยม|เชื้อชาติ
race+ideology	tha	คตินิยมเชื้อชาติ	คตินิยม|เชื้อ|ชาติ
race+principle	tha	คตินิยมเชื้อชาติ	คติ|นิยม|เชื้อชาติ
race+-ness	tur	ırkçılık	ırk|çı|lık
race+difference	jpn	人種差別	人種|差|別
race+meaning	zho	種族主義	種族|主|義
race+meaning	jpn	人種主義	人種|主|義
race+look	zho	種族歧視	種族|歧|視

Figure 4.9: Recipes for RACISM. Some instances of incorrect decompositions nevertheless result in the same recipe. For example, 种族|主义 'race' + '-ism, ideology', and 种|族|主义 'race, type' + filler + '-ism, ideology'.

SUBWAY =	underground (75)	+	way (58)
	ground (46)		railway (35)
	subterranean (36)		road (32)
	earth (33)		path (28)
	land (27)		passage (19)
	soil (24)		track (18)
	beneath (23)		train (17)
	place (22)		iron (17)
	tunnel (20)		street (14)
	dirt (17)		trajectory (13)

Recipe	Lang	Word	Segmentation
underground+way	jpn	地下道	地下|道
underground+way	jpn	地下鉄道	地下鉄|道
underground+way	zho	地下道	地下|道
underground+way	isl	neðanjarðarbraut	neðanjarðar|braut
underground+way	zho	地下鐵路	地下鐵|路
underground+way	epo	subtera fervojo	subtera| fer|vojo
underground+way	spa	paso subterráneo	paso| |subterráneo
underground+way	jpn	地下鉄道	地下|鉄|道
underground+way	zho	地下鐵路	地下|鐵|路
underground+way	zho	地下铁路	地下|铁|路
underground+railway	mya	မြေအောက်မီးရထား	မြေအောက်| မီးရထား
underground+railway	jpn	地下鉄道	地下|鉄道
underground+railway	jpn	ちかてつどう	ちか|てつどう
underground+railway	zho	地下鐵路	地下|鐵路
underground+railway	zho	地下铁路	地下|铁路
underground+railway	ces	podzemní dráha	podzemní| |dráha
underground+railway	epo	subtera fervojo	subtera| |fervojo
underground+railway	hin	भूमिगत रेल	भूमिगत| |रेल
ground+way	jpn	地下道	地|下|道
ground+way	zho	地下道	地|下|道
underground+passage	mkd	подземен премин	подземен| |премин
underground+passage	ron	pasaj subteran	pasaj| |subteran
underground+passage	rus	подзе́мный перехо́д	подзе́мный| |перехо́д
underground+iron	jpn	地下鉄	地下|鉄
train+underground	tha	รถไฟใต้ดิน	รถไฟ|ใต้ดิน
underground+train	fin	metrojuna	metro|juna
underground+iron	zho	地下鐵	地下|鐵
underground+iron	zho	地下铁	地下|铁
underground+train	mya	မြေအောက်မီးရထား	မြေအောက်| မီး| ရထား
underground+walking	rus	подзе́мный перехо́д	подзе́мный| пере|хо́д

Figure 4.10: Recipes for SUBWAY.

WORKER =	work (210)	+	-er (74)
	labour (76)		person (47)
	job (71)		man (30)
	labor (44)		people (26)
	employment (44)		human being (24)
	business (34)		human (18)
	worker (32)		work (17)
	deed (32)		female (15)
	occupation (30)		-ist (15)
	task (27)		-ian (14)

Recipe	Lang	Word	Segmentation
work+-er	deu	Arbeiter	Arbeit|er
-er+work	ind	pekerja	pe|kerja
work+-er	lat	operator	opera|tor
work+-er	hye	բանվոր	բան|վոր
work+-er	nld	arbeider	arbeid|er
work+-er	jpn	勤労者	勤労|者
-er+work	khm	អ្នកធ្វើការ	អ្នក|ធ្វើការ
work+-er	fas	کارگر	کار|گر
work+-er	ron	muncitor	munci|tor
work+-er	ron	muncitoare	munci|toare
-er+work	tha	คนงาน	คน|งาน
-er+work	tha	คนทำงาน	คน|ทำงาน
work+-er	yid	ארבעטער	ארבעט|ער
work+-er	zho	工人	工|人
work+-er	nld	werker	werk|er
work+-er	zho	工作者	工作|者
work+-er	zho	打工人	打工|人
work+-er	zho	做工的	做工|的
work+-er	deu	Arbeitnehmer	Arbeit|nehm|er
work+-er	isl	starfsmaður	starf|s|maður
work+-er	nld	arbeidster	arbeid|st|er
work+-er	fra	travailleur	travail|l|eur
work+-er	jpn	労働者	労|働|者
-er+work	tha	คนทำงาน	คน|ทำ|งาน
work+-er	zho	勞動者	勞|動|者
work+-er	zho	工作者	工|作|者
person+work	lao	ຄົນການ	ຄົນ|ການ
work+person	vol	voban	vob|an
work+person	nld	werkman	werk|man
work+person	fin	työihminen	työ|ihminen

Figure 4.11: Recipes for WORKER. This is another example where a bound morpheme *-er* is identified as a component, because *-er* exists in our dictionaries as a separate entry. Traditional dictionaries often do not include these affixes as entries.

	book (114)		house (30)
	beech (12)		collection (16)
	room (12)		building (15)
	program (10)		book (13)
LIBRARY =	free (9)	+	library (12)
	letter (8)		room (12)
	writing (8)		chamber (11)
	diagram (8)		notebook (9)
	library (8)		document (8)
	quire (7)		ventricle (8)

Recipe	Lang	Word	Segmentation
book+house	isl	bókahús	bóka|hús
book+house	jpn	図書館	図書|館
book+house	ang	bōchūs	bōc|hūs
book+house	fas	کتابخانه	کتاب|خانه
book+house	gla	leabharlann	leabhar|lann
book+house	tgk	китобхона	китоб|хона
book+house	ang	bochus	boc|hus
book+house	zho	書房	書|房
book+house	isl	bókahús	bók|a|hús
book+collection	hun	könyvtár	könyv|tár
book+collection	isl	bókasafn	bóka|safn
book+collection	est	raamatukogu	raamat|u|kogu
book+collection	fao	bókasavn	bók|a|savn
book+collection	isl	bókasafn	bók|a|safn
book+building	kor	도서관	도서|관
building+book	mri	whare pukapuka	whare| |pukapuka
building+book	tpi	haus buk	haus| |buk
room+book	tha	ห้องสมุด	ห้อง|สมุด
book+room	fin	kirjakammio	kirja|kammio
book+room	jpn	図書室	図書|室
book+place	gle	leabharlann	leabhar|lann
book+place	kir	китепкана	китеп|кана
book+place	tam	நூல் நிலையம்	நூல்| |நிலையம்
book+cupboard	nld	boekenkast	boeken|kast
book+mother	aze	kitabxana	kitab|x|ana
book+mother	kaz	кітапхана	кітап|х|ана
book+storehouse	mon	номын сан	ном|ын |сан
book+cupboard	nld	boekenkast	boek|en|kast
book+place of	eus	liburutegi	liburu|tegi
book+place of	pus	کتابتون	کتاب|تون

Figure 4.12: Recipes for LIBRARY. Here we see the splitting model can handle morphological variants. For example, *bókasavn* is analyzable as bók|a|savn 'book' + genitive plural suffix + 'collection, museum'.

	un- (93)		go (65)
	away (61)		run (57)
	de- (57)		flee (55)
	escape (54)		escape (49)
ESCAPE =	out of (48)	+	leave (28)
	even (32)		move (26)
	dis- (29)		walk (23)
	e (26)		go away (22)
	Wu (20)		to go (22)
	dis (18)		leak (20)

Recipe	Lang	Word	Segmentation
un-+go	deu	entziehen	ent|ziehen
un-+go	deu	entgehen	ent|gehen
un-+go	nld	ontgaan	ont|gaan
un-+go	ell	ξεφεύγω	ξε|φεύγω
un-+go	nld	onttijgen	ont|tijgen
un-+go	deu	entfahren	ent|fahren
un-+go	nld	ontgaan	on|t|gaan
un-+go	nld	onttijgen	on|t|tijgen
un-+run	deu	entrinnen	ent|rinnen
un-+run	ita	svicolare	s|vi|colare
un-+flee	deu	entfliehen	ent|fliehen
un-+flee	ita	sfuggire	s|fuggire
away+go	ang	wiþfaran	wiþ|faran
away+go	rus	уходить	у|ходить
escape+go	kor	도망가다	도망|가다
escape+go	zho	逃走	逃|走
un-+move	deu	entrücken	ent|rücken
escape+go	zho	遁走	遁|走
escape+go	slv	pobegniti	pobeg|n|iti
away+run	rus	убегать	у|бегать
away+run	rus	убежать	у|бежать
away+run	rus	утечь	у|течь
away+flee	hun	elillan	el|illan
un-+to go	nld	ontvaren	ont|varen
un-+to go	nld	ontvaren	on|t|varen
de-+run	jpn	脱走	脱|走
un-+come	nld	ontkomen	ont|komen
out of+go	lat	evado	e|vado
un-+come	deu	entkommen	ent|kommen

Figure 4.13: Recipes for ESCAPE. Most recipes have some form of *un-*. The English word *escape* actually comes from Latin *ex* 'out' + *cappa* 'cape, cloak', with the interpretation of *escape* as leaving your pursuer with only your cloak.

AZURE =	sky (17) azure (16) blue (13) celestial (12) heavenly (8) heavens (7) east (6) goal (6) day (6) dress (6)	+	blue (46) azure (18) -ness (6) slaughter (6) pink (6) The color blue (5) -er (5) -ish (5) -al (5) Lan (5)

Recipe	Lang	Word	Segmentation
sky+blue	zho	天藍色	天|藍色
sky+blue	bul	небе́сносин	небе́|сно|син
sky+blue	bul	небесносин	небе|сно|син
azure+blue	nld	azuurblauw	azuur|blauw
azure+blue	fin	asuurinsininen	asuuri|n|sininen
blue+celestial	msa	biru langit	biru| |langit
blue+celestial	spa	azul celeste	azul| |celeste
east+blue	por	azul celeste	azul| ce|leste
east+blue	por	azul celeste	azul| cel|este
east+blue	spa	azul celeste	azul| cel|este
dress+blue	fin	asuurinsininen	asu|urin|sininen
beautiful+blue	isl	fagurblár	fagur|blár
grand+blue	zho	蔚藍	蔚|藍
of the sky+blue	fin	taivaansininen	taivaan|sininen
clear sky+blue	isl	heiðblár	heið|blár
in every manner+blue	epo	ĉielblua	ĉiel|blua
or+blue	fin	taivaansininen	tai|vaan|sininen
subject+blue	isl	fagurblár	fag|ur|blár
this+blue	epo	ĉielblua	ĉi|el|blua
water+blue	tat	зәңгәрсу	зәңгәр|су
happy+blue	zho	湛藍	湛|藍
his+blue	est	taevasinine	ta|eva|sinine
everywhere+blue	epo	ĉielblua	ĉie|l|blua
Æsir+blue	nob	asurblå	as|ur|blå
fermentation+blue	nno	asurblå	as|ur|blå
blue+diminutive ending	fin	sininen	sini|nen
+navy blue	jpn	紺碧	紺|碧
blue+third person possessive suffix	fin	sininen	sini|n|en
azure+celestial	por	azul celeste	azul| |celeste
azure+-ness	fas	لاجوردی	لاجورد|ی

Figure 4.14: Recipes for AZURE. The English word *azure*, as well as French *azur*, Spanish *azul*, Italian *azzurro*, etc. originate from Arabic لازورد lāzaward 'lapis lazuli', which is from Persian لاجورد lājevard. Lājevard is a region in present-day Afghanistan and Tajikistan where the stone was originally mined.

FLAMINGO =	red (21) go (16) blood tofu (6) crimson (5) flame (5) golden (4) burn down (4) internal heat (4) roasted (4) light (4)	+	goose (12) crane (12) flamingo (6) flaming (6) Cygnus (4)

Recipe	Lang	Word	Segmentation
red+goose	aze	qızılqaz	qızıl|qaz
red+goose	tat	кызылказ	кызыл|каз
red+goose	uzb	qizilgʻoz	qizil|gʻoz
red+crane	zho	紅鶴	紅|鶴
red+crane	zho	紅鸛	紅|鸛
red+crane	zho	火鶴	火|鶴
red+crane	zho	火鹤	火|鹤
red+crane	tur	allı turna	al|lı |turna
red+crane	vie	hồng hạc	hồng| |hạc
red+flamingo	fin	flamingonpunainen	flamingo|n|punainen
red+little	aze	qızılqaz	qızıl|q|az
red+little	uzb	qizilgʻoz	qizil|gʻ|oz
red+bit	tat	кызылказ	кызыл|к|аз
red+pickaxe	ckb	سوورەقوڵنگ	سوور|ە|قوڵنگ
red+bird	zho	火烈鳥	火|烈|鳥
red+prick	zho	火烈鸟	火|烈|鸟
blood tofu+crane	zho	红鹤	红|鹤
flam+go	eng	flamingo	flam|in|go
crimson+crane	jpn	紅鶴	紅|鶴
daughter+goose	aze	qızılqaz	qız|ıl|qaz
daughter+goose	tat	кызылказ	кыз|ыл|каз
daughter+goose	uzb	qizilgʻoz	qiz|il|gʻoz
state+goose	hin	राजहंस	राज|हंस
to+goose	hin	बगहंस	ब|ग|हंस
Ra+goose	hin	राजहंस	रा|ज|हंस
heart+crane	vie	hồng hạc	hồn|g |hạc
blood tofu+stork	zho	红鹳	红|鹳
flamingo+women's	fin	flamingonpunainen	flamingo|npu|nainen
flaming+o	eng	flamingo	flaming|o
flame+bit	cym	fflamingo	fflam|in|go

Figure 4.15: Recipes for FLAMINGO. The first character 红 in 红鹤 means 'red', but because Chinese in Wiktionary is standardized to use traditional characters, simplified Characters like 红 are not fully defined.

REINDEER =

north (25)
northern (21)
reindeer (17)
deer (11)
clean (8)
pure (8)
rein (6)
utter (4)

+

deer (53)
animal (15)
stag (15)
red deer (14)
male deer (10)
buck (8)
beast (6)
Viking (6)
Scandinavian (6)
Norwegian (6)

Recipe	Lang	Word	Segmentation
north+deer	hye	հյուսիսային եղջերու	հյուսիս|ային |եղջերու
north+deer	hye	հյուսիսային եղջերու	հյուսիսային| |եղջերու
north+deer	bel	паўно́чны але́нь	паўно́чны| |але́нь
north+deer	bul	се́верен еле́н	се́вер|ен |еле́н
north+deer	kat	ჩრდილოეთის ირემი	ჩრდილოეთი|ს |ირემი
north+deer	rus	се́верный оле́нь	се́вер|ный |оле́нь
north+deer	rus	се́верный оле́нь	се́верный| |оле́нь
north+deer	hbs	severni jelen	sever|ni |jelen
north+deer	hbs	severni jelen	severni| |jelen
north+deer	hbs	sjeverni jelen	sjever|ni |jelen
north+deer	hbs	sjeverni jelen	sjeverni| |jelen
north+deer	slv	severni jelen	sever|ni |jelen
north+deer	ukr	півні́чний оле́нь	півні́чний| |оле́нь
north+deer	epo	norda cervo	norda| |cervo
northern+deer	bul	се́верен еле́н	се́верен| |еле́н
northern+deer	kat	ჩრდილოეთის ირემი	ჩრდილოეთის| |ირემი
northern+deer	fas	گوزن شمالی	گوزن| |شمالی
reindeer+deer	hun	rénszarvas	rén|szarvas
deer+reindeer	tha	กวางเรนเดียร์	กวาง|เรนเดียร์
reindeer+deer	dan	rensdyr	ren|s|dyr
rein+deer	eng	reindeer	rein|deer
re+deer	dan	rensdyr	re|ns|dyr
re+deer	hun	rénszarvas	ré|n|szarvas
re+deer	eng	reindeer	re|in|deer
deer+snow	gla	fiadh-sneachda	fiadh-|sneachda
snow+deer	eus	elur-orein	elur|-|orein
deer+snow	bre	karv-erc’h	karv|-|erc’h
deer+snow	gla	fiadh-sneachda	fia|dh-|sneachda
deer+snow	gla	fiadh-sneachda	fiadh|-|sneachda
shadow+deer	kat	ჩრდილოეთის ირემი	ჩრდილო|ეთის |ირემი

Figure 4.16: Recipes for REINDEER.

In the above figures, I show several examples of universal compound models learned across all the languages available in the dictionary. We see some general language-universal realizations. For example, occupations often have a word for “man” or “human” as a compound (e.g. astronaut = space + man, worker = work + person). Locations may have a word for “room” or “house” (e.g. hospital = ill + house, kitchen = cook + room, library = book + house). Disciplines of study often have “science” or a translation of “-ology” (e.g. linguistics = language + science, biology = life + science). Other concepts like coronavirus = crown + virus, flamingo = red + goose, and reindeer = north + deer are representative of the head word’s appearance or geographic location.

These are just a handful of examples, but they show a remarkable range of compound processes that are all captured by the compounding model. A full listing of these models recipes can be found at https://github.com/wswu/worcomal. Discovering these patterns across languages can shed insight into how humans construct words for new concepts. In the rest of this chapter, I utilize these models in the practical task of translation of unknown words.

4.1.4 Compound Analysis

Using the universal compound models learned from Wiktionary, I predict the translation of unknown foreign compound words. I largely follow Garera and Yarowsky (2008)'s multipath approach, which is explained in Section 2.2. Their method uses a collection of 50 foreign-English dictionaries acquired online or via optical character recognition. Since then, Wiktionary has grown to be one of the largest sources of bilingual translations, which I utilize here to provide substantially more signal for the component translations. Besides enlarging the source of training translations by over an order of magnitude, I extend their work using several new compound splitting mechanisms detailed in the previous section.

In the analysis direction (as opposed to generation), the task is to analyze a foreign compound word and identify the English translation. The multipath translation model decomposes the foreign word as a compound of known components and builds a distribution of compositional translations. For example, my universal compounding model learns that HOSPITAL = ill/sick/disease + house/home/building. The multipath model applies the knowledge from the compounding model, so that any foreign word that is composed of known components (e.g. ill and house, as in Danish *sygehus*) can potentially be translated

Lang	# Words	Acc1	Acc10	Acc100	AccN
bul	739	.06	.14	.25	.53
gle	502	.07	.18	.26	.60
glg	617	.11	.22	.32	.66
mlt	234	.01	.05	.08	.23
bul	606	.07	.17	.30	.65
gle	443	.08	.21	.30	.68
glg	541	.13	.25	.37	.75
mlt	185	.01	.06	.10	.29

Table 4.6: Evaluation of multipath compound translation. The top section contains results on all test words that exist in the dictionary. The bottom section contains results for which the model generated at least one hypothesis.

as 'hospital', even though the entire word has never been seen during training.

I evaluate the multipath translation model on the task of foreign to English translation, on four languages: Bulgarian, Irish, Galician, and Maltese. This test set contains both medium and low-resource languages and is explained in detail in Chapter 7. In several cases, if the decomposition of the foreign word does not result in an existing compounding recipe, the model does not output any hypotheses. In the bottom half of Table 4.6, I limit the evaluation to words for which the model generated at least one hypothesis, i.e. the decomposition of the foreign word resulted in a compounding recipe that the model had learned.

Table 4.7 shows some model predictions from Irish. I see that in addition to compound words, the model is able to capture suffixes such as *-ach*. I notice that even though in many cases the model does not predict the correct English translation as the first ranked hypothesis, it generates translations that are semantically related (e.g. asteroid/planetoid/minor

Word	Gold Trans.	Idx	Model Hypotheses
Airméanach	Armenian	2	Armenian man, Armenian person, **Armenian**, Armenian woman
mionphláinéad, astaróideach	asteroid	1	asteroidal, **asteroid**, planetoid, Ixion, minor planet, China aster, 1 Ceres, Ceres, bearer of ill luck
féinriail, féinriar	autonomy	0	**autonomy**, individual freedom, self-rule, self-service, self-sufficiency, self-medicate, egotistical, spontaneous
déghnéasach	bisexual	11	parents, two-spirited, two-spirit, be hot, hot, airtight, magnet, demisexual, Horned God, ambiguous
gréasaí	cobbler	0	**cobbler**, shoemaker, hand-made boots, basa, bootmaker, ornamented, embroidered, patterned, ornament
leantóir, lorgaire	follower	2	lawnmower, trailer, **follower**, tracker, detective, pursuer, adherent, seeker, searcher

Table 4.7: Example translations from Irish by the multipath translation model.

Lang	# Words	Acc1	Acc10	Acc100	AccN
bul	739	.12	.27	.42	.63
gle	502	.12	.29	.47	.73
glg	617	.16	.30	.50	.73
mlt	234	.01	.07	.16	.45
bul	606	.15	.33	.51	.76
gle	443	.14	.33	.53	.82
glg	541	.18	.34	.57	.84
mlt	185	.02	.09	.21	.57

Table 4.8: Evaluation of multipath compound translation, with an expanded set of gold English translations using the lexical relation model. The top section contains results on all test words that exist in the dictionary. The bottom section contains results for which the model generated at least one hypothesis.

planet, or follower/tracker/seeker). Interestingly, for asteroid, the model generated Ixion and Ceres, which are names of dwarf planets. Evaluating in this way may also miss correct words that are not listed as gold, because other translations may be acceptable (e.g. *Armenian man* and *Armenian person* should also be acceptable).

Thus, I expand the set of valid English translations using the lexical relations translation model described in Section 4.2. This is useful because the multipath translation model may have learned a compounding recipe for a synonym of the gold word, rather than for the word itself, which limits the performance of this model. In Table 4.8, I present several evaluation metrics on this expanded set.

4.1.4.1 Learning compound morphology

By examining the different processes used in constructing compound words, we obtain a greater understanding of how specific languages perform compounding. Compound analysis with diverse splitting algorithms is able to automatically identify morphology of compound words that appear as epenthesized characters. For example, the following table shows the distribution of linking characters that my model discovered in German (*empty* denotes the empty string, and underscore indicates a space):

Link	Prob
<empty>	0.0735
_	0.0106
s	0.05
n	0.005
e_	0.04

Most languages construct compounds simply by concatenating two words directly without insertions or deletions (although often in variable order). Similarly, many multi-word expressions are simply the concatenation of separate words with a space. For compound words, German favors ‘s’ and ‘n’ as epenthesized characters. As an interfix, *s* is well-known to occur between compounds and indicates the genitive case, e.g. *Bildungsroman*. In contrast, *n* is a genitive suffix appended to the first word, e.g. *Schützengraben* ‘trench’ = *Schütze* ‘shooter’ + *n* (genitive suffix) + *graben* ‘dig’. In contrast to much existing work, my innovation of supporting multi-character glues allows us to discover “e_” as an common epenthesis formula in which the first word is inflected, e.g. *öffentliche Meinung* ‘public opinion’ = *öffentlich* ‘public’ + *e_* (definite feminine suffix) + *Meinung* 'opinion'. This allows my model to parse certain types of multiword expressions. Handling these different compound processes is not only useful for predicting whether a word is a compound, but can also be useful when generating previously unknown compound word translations into the language.

In Figure 4.17 I list the most common epenthesis mechanisms for several languages. I point out several observations. English as many multiword expressions, as evidenced by links such as a space[7] (e.g. *mountain lion*, *couch potato*), _of_ (e.g. *act of Congress*, *type of plant*) and - (e.g. *light-footed*, *gram-positive*. Likewise for _de_, which occurs in French (e.g. *nom de baptême* 'baptismal name', *photo de profil* 'profile picture') and Spanish (e.g. *fin de semana* 'weekend', *barra de equilibrio* 'balance bar'). This type of link is similar to a genitive inflectional ending, but would not be captured by traditional compound word analyses that only deal with single words. Note that the compounding model can also deal with various writing scripts, enabling future compound analysis studies in understudied languages.

Finally, I calculate for each language the probability that a specific word is likely to be used in compound words. I find that the most common components in compound words are often affixes productive. For example, in English, the most frequent components include *er*, *ing*, *ly*, and *ness*, which are all suffixes. In Chinese, 人 and 者 are some of the most common components, analogous to the *-er* suffix in English. This information will be useful in the following section on compound generation.

[7] Which is actually more common than concatenation without epenthesis.

English

Link	Prob
_	0.149
empty	0.063
_of_	0.010
e_	0.008
,_	0.008
e	0.007
_or_	0.007
-	0.007
_a_	0.006
n	0.006

French

Link	Prob
empty	0.144
_	0.065
_de_	0.020
n	0.015
men	0.014
i	0.013
-	0.012
t	0.010
s	0.009
o	0.008

Spanish

Link	Prob
empty	0.146
_	0.051
s	0.020
_de_	0.019
n	0.017
r	0.015
m	0.012
l	0.009
t	0.009
i	0.008

Chinese

Link	Prob
empty	0.699
不	0.003
仔	0.002
人	0.002
頭	0.002
主	0.002
大	0.002
生	0.002
子	0.002
學	0.002

Swedish

Link	Prob
empty	0.216
s	0.038
_	0.024
n	0.017
t	0.015
l	0.013
r	0.011
k	0.010
d	0.009
v	0.008

Russian

Link	Prob
empty	0.153
с	0.036
_	0.035
о	0.020
к	0.011
т	0.009
и	0.009
н	0.008
ст	0.008
ʹ	0.008

Japanese

Link	Prob
empty	0.477
ん	0.011
い	0.010
う	0.009
の	0.007
し	0.007
っ	0.005
り	0.005
く	0.005
か	0.005

Greek

Link	Prob
empty	0.384
πο	0.053
α	0.044
ια	0.038
να	0.036
σ	0.027
_	0.021
ν	0.012
δ	0.010
γ	0.009

Figure 4.17: Epenthesis mechanisms for several languages, along with their associated normalized counts. The underscore _ denotes a space, and *empty* denotes the empty string, i.e. concatenation without epenthesis. Empty filler (simple concatenation) is the most common compounding mechanisms for most languages that I examined.

4.1.5 Compound Generation

A massively multilingual examination of compounding is interesting in and of itself. However, from a practical standpoint, compounding finds applications particularly in machine translation (e.g. Koehn and Knight, 2003; Stymne, Cancedda, and Ahrenberg, 2013). For low-resource languages, where complete lexicons might not be available, one can create possibly valid compound words from known components. This phenomenon has also been documented in second language learners (N. Shqerra and E. Shqerra, 2014).

In the task of compound generation, the goal is to produce a compound word f in a language l, given the concept e. I model the generation of compound words using the following probabilistic model, whose components have been described in the previous sections:

$$p(f \mid l, e) = p(f \mid e) \cdot p(f \mid l) \tag{4.8}$$

$$= p(link \mid l) \cdot p(flip \mid l) \prod_{part \in e} p(part \mid e) \prod_{pt \in tr(part)} p(pt \mid l) \tag{4.9}$$

where

- $part$ are the component parts in the multilingual compounding model
- $tr()$ is a function that translates English to the target language l
- pt is the translation of $part$ in language l

- $p(part \mid e)$ is the probability of $part$ as a component in the compound model for concept e
- $p(pt \mid l)$ is the probability that pt is a component in compounds in language l , defined as $\frac{\text{\# of compounds in } l \text{ containing } pt}{\text{\# of compounds in } l}$
- $p(flip \mid l)$ is the probability of the language flipping the ordering of words in the compound model
- $p(link \mid l)$ is the probability of the link (concatenation, epenthesized characters, etc.) between the component parts

This generative model takes into account various features of compound words described in the above sections of this chapter. In comparison to previous work, e.g. Koehn and Knight (2003), who use the geometric mean of the frequency of each compound part to filter the potential compound list, I assume no access to bitext or other corpora, which is a reasonable assumption for low-resource languages.[8] To generate compound words given a semantic concept, the model iteratively sample from each of these probabilities. For example, this model can generate a realization of the concept *hospital* in Chinese as follows:

1. Select argmax $p(link \mid l)$, the highest probability link in Chinese (concatenation without epenthesis)

[8] Many languages have at least a translation of the Bible, but this is a small text with vocabulary in a narrow domain.

2. Select argmax $p(part_1 \mid \text{hospital})$, the highest probability left component (sick)

3. Select argmax $p(tr(\text{sick}) \mid \texttt{zho})$ the highest probability translation of *sick* in Chinese (病)

4. Select argmax $p(part_2 \mid \text{hospital})$, the highest probability right component (house)

5. Select argmax $p(tr(\text{house}) \mid \texttt{zho})$, the highest probability translation of *house* in Chinese (家)

6. Select argmax $p(flip \mid \texttt{zho})$, whether to flip or not (do not flip)

7. The resulting generated compound is 病家

Interestingly, by performing this compound construction procedure, it is possible to construct entirely new compound words. For example, the above procedure generated an actual word: 病家 'a patient and their family' which does not exist in the training dictionary.[9] This illustrates that even in "comprehensive" dictionaries like Wiktionary, translation between certain terms may only occur one-way, and lexicon expansion techniques discussed in this thesis are useful for improving coverage of Wiktionary and other multilingual dictionaries.

Of course, we need not limit ourselves to the most likely compound according to the model, because in a real-world application, one would generate potentially thousands of hypothetical compounds which would be filtered using a corpus in a target language. This

[9] This entry does exist in the Chinese edition of Wiktionary, but not the English one, presumably because there is no concise English word for 'a patient and their family'.

generation procedure is also straightforard to extend. In future work, one may replace the component translation function $tr()$ with other sources of bilingual translation, such as alignments or online translation software, if these are available for the language. My current work assumes that no such sources are available. As hinted in Section 4.1.4.1, future work could apply morphological rules, such as looking up genitival suffixes of the leftmost component using UniMorph (Kirov, Cotterell, et al., 2018) or other inflectional databases in addition to using learned epenthesis characters, though I have found that the learned filler characters capture these morphological variants.

4.1.5.1 Compositionality Score

I devise a score of the compositionality of a concept across languages to determine the likelihood that any given concept is realized as a compound word. This score can be seen as the model's confidence in the compositionality of a concept. I define this score as follows:

$$\text{compositionality}(\text{concept}) = \frac{log\left(\frac{1}{2}(\text{recipe left count + recipe right count}\right)}{max(\text{recipe count across all recipes})} \tag{4.10}$$

This compositionality score computed for a sample of concepts in the test set is shown in Table 4.9.

If we assume that concepts with a compositionality of greater than 0.5 can be consid-

nineteen	0.98
username	0.84
second person	0.81
unnecessary	0.77
sailing ship	0.76
secondhand	0.74
well	0.73
exclamation mark	0.71
control	0.7
town	0.69
anew	0.68
delay	0.67
redeem	0.66
adverb	0.65
Cold War	0.64
conman	0.63
digestive system	0.62
asymmetrical	0.62
over	0.6
handsome	0.6
furious	0.59
optical illusion	0.58
impudent	0.58
microbe	0.57
supplement	0.56
confess	0.55
serf	0.54
prosody	0.54
boring	0.52
sinusitis	0.52
topple	0.51
basalt	0.5
iPhone	0.49
vestibule	0.48
resin	0.47
Chicago	0.46
bet	0.45
glory	0.44
continuity	0.44
galangal	0.43
Bauhaus	0.42
rug	0.39
cardigan	0.38
amanita	0.37
Palestine	0.34
Sahara	0.32
Europa	0.3
Michigan	0.26
Henry	0.21
kibbutz	0.08

Table 4.9: Compositionality scores for a sample of concepts across the test set.

Lang	#	Acc1	Acc10	Acc100	AccN	Ed1	Ed10	Ed100
bul	740	.00	.01	.03	.10	6.52	5.00	3.87
gle	505	.01	.02	.03	.07	6.60	4.88	3.76
glg	619	.01	.01	.03	.12	6.10	4.46	3.38
mlt	235	.00	.00	.01	.02	5.93	4.25	3.47

Table 4.10: Compound generation task.

ered compositional, then only a little over half of the words in the testset are compositional and amenable for the compositional model. Specifically, in the test set, 472/739 (.64) concepts for Bulgarian, 349/502 (.7) for Irish, 401/617 (.65) for Galician, and 162/234 (.69) for Maltese satisfied this criterion.

4.1.5.2 Evaluation

I evaluate the compound generation model on the task of English to foreign unknown compound translation. I again test on four languages, explained in more detail in Chapter 7. In this task, I assume no source of bilingual translations except for a small bilingual dictionary, which the model is trained solely on. This is precisely the scenario described in the introduction chapter of this dissertation: we wish to communicate with the local people of a low-resource language, but do not have existing machine translation systems nor adequate resources for training them. We may have a native informant who can give us a small dictionary, with which we can exploit the connections with our existing multilingual dictionaries. For the compound generation task, results are shown in Table 4.10. I report both accuracy and mean character edit distance.

From Table 4.10, we can immediately see that the compound generation task is a difficult one. Given only a small seed dictionary in the target language, the compound model generates into a vacuum, using only the knowledge of how compounds are formed in other languages around the world. However, the low accuracies belie the power of the compound generation model. As in the compound analysis direction, the 1-best accuracy is not a useful metric. Examining the 100-best list may even be too restrictive, because in a real-world scenario, the model will precompute a n-best list, where n can be on the order of 10,000 or even 1 million. Then, when we encounter any monolingual text in the target language, we can build a language model, which can be used to prune this n-best list. Thus, in these evaluations, I focus more on recall (AccN), and edit distance to the gold word.

Edit distance is computed as follows: Ed1 is the minimum edit distance between the first-ranked hypothesis and any gold translations. Ed10 is the minimum edit distance between any of the top 10 ranked hypothesis and any gold translations, and so on.

The compound model may have several points of failure that prevent it from generating the correct word. I examine each of these in turn.

Does the recipe exist? Almost every concept in the test set had an associated compound recipe. For each test language, 731/740 recipes exist for Bulgarian, 501/505 for Irish, 612/619 for Galician, and 233/235 for Maltese. Thus, the existence of a recipe did not significantly affect the overall results.

Does the recipe generalize? For concepts that are not universally compound, the recipes may have some noise. In such cases, the compounding model would not be nec-

Concept	Gold	Recipe	Model Hypotheses
Khmer	khmer	Cambodia + language	Cambodjalingua,Cambodjafala,Cambodjalinguaxe,altolingua,altofala,outolingua,altalingua,outofala,altafala
Latin	latín	Roman/Latin + language	romanolingua,latinolingua,romanofala,latinofala,latínlingua,latinoamericanolingua,latínfala

Table 4.11: Certain concepts, like names of languages, are often compositional across languages but not in English.

essarily applicable. For example, concepts such as BLOOD or WHITE are more likely to be amenable to prediction by a cognate model (Chapter 5) than a compositional model. As mentioned in Section 4.1.5.1, only roughly 60% of the test concepts could be amenable to compound analysis.

In Section 4.1.1, I showed that the recipes for concepts often realized as compounds are robust. However, certain realizations are language specific, e.g. FRIDAY = week + five in Chinese.[10] This recipe simply cannot be learned if the only instance of week + five is held out from the training set. Another class of concepts are those that are often compound across languages, but are not in the specific language. For example, Table 4.11 shows that language names can sometimes be better predicted by cognate models.

Do component translations exist? Even if the recipe exists, and it adequately captures the compositional formula for realizing a particular concept, the model may not be able to generate the actual word because the dictionary does not contain a translation for the components.

Is the compound joining process effective? With the correct recipe and component translations, the last step is to join the components. The proposed compound generation model is a brute force solution, enumerating the different translation possibilities and

[10] The common recipes are 'Venus day' and 'Frigg day' (Frigg is the Germanic goddess associated with the Roman goddess Venus), or 'gold day'.

joining them via concatenation, epenthesis of linking chacters, elision of the first component, and flipping the ordering of the components. I found this to be a limiting factor in generating accurate compounds, which motivated a neural model for compound joining.

4.1.5.3 Neural Compound Component Joining

I experiment with neural sequence-to-sequence models on the task of compound component joining: given two components of a compound word (e.g. English *bid* and *able*), generate the compositional word *biddable*. This may involve concatenation, epenthesis, elision, or any other string transduction process. Rather than explicitly modeling this as in Section 4.1.5, I let the sequence-to-sequence model handle the joining process.

I train and test on the prefixal, suffixal, affixal, and compound data from Wiktionary used above in Section 4.1.2, because these words have gold decompositions. I experiment with three common neural sequence-to-sequence models: a LSTM encoder-decoder, the same model with copy attention, and a Transformer model. The input to the model is a sequence containing the language code and the characters of each component, followed by a pipe symbol. The output of the model is the character sequence of the resulting compound word. For example, consider the Old English word *wunung* 'residence' = *wunian* 'to live' + *-ung* (noun-to-verb suffix):

```
Input:   ang w u n i a n | - u n g
Output:  w u n u n g
```

Results on a held out test set are shown in Table 4.12

Model	Acc1	Acc10	AccN	Ed1	Ed10	EdN
LSTM	.72	.85	.88	.76	.43	.35
LSTM Copy	.58	.74	.74	.93	.61	.61
Transformer	.60	.81	.85	.98	.52	.43

Table 4.12: Results on the component joining task on Wiktionary words.

Surprisingly, the LSTM model outperformed the Transformer model, which is currently one of the dominant models in NLP. Further investigation is necessary to determine the exact reasons. I show a random sample of the LSTM model's output in Table 4.13. I find that the neural model is able to handle the concatenation, epenthesis, and elision processes, as well as other types of compound joining, including elision of the right component (e.g. Danish skråne + -ing = skråning) and a change of left component suffix (e.g. Italian vuoto + mente = vuotamente) which were not previously handled.

Inspired by the neural model's successes, I apply this LSTM model to join components in the compound generation algorithm. Specifically, for each test concept, I take the top 100 hypotheses generated by the model before component joining, and apply the neural sequence to sequence model to generate a 50-best list for each hypothesis. I combine these hypotheses by the neural model's decoder score to generate a single n-best list of hypothesized compound words. Evaluation of this list is shown in Table 4.14 as the Neu model. The original compound generation is indicated by BF (brute force) model. In addition, I combine the n-best lists of the BF and Neu models by concatenating the two hypothesis lists and reranking based on their respective model's score. This list is denoted as Combined in Table 4.14.

Source	Gold	Idx	Hypotheses
ang singal | līċe	singallice	0	**singallice**, singalice, singal lice, singalalice, singallis, singallicee, singal licee
ang witlēas | þu	witleast	-1	witleasþu, witleaþu, witleaþur, witleaþul, witleaþun, witleasþul, witleasþulo, witleasþula
bak әрмән | стан	Әрмәнстан	-1	әрмәнстан, әрмән стан, әрмәстан, әрмәістан, әрмән-стан, әрмәнстанм, әрмәнстанмак
ces pár | ek	párek	0	**párek**, páek, pár ek, perek, Párek, párekro, párekre, párekra, párekrar, párekran
cym iaith | adur	ieithadur	-1	iaith adur, iaithadur, iaith-adur, iaithidur, iaithedur, iaith adura, iaith aduro, iaith adurar
dan Polen | sk	polsk	-1	Polensk, Polen, Polen sk, Polenisk, Polentk, Polent, Polensko, Polen sko, Polen skro, Polen skre
dan skråne | -ing	skråning	0	**skråning**, skrening, Skråning, skråneing, skrånting, skråneinge, skråneinging, skråneingro
deu Kasache | isch	kasachisch	1	Kasachisch, **kasachisch**, Kasacheisch, kasacheisch, Kasachenisch, Kasacheischo
deu streiten | ig	streitig	1	streitenig, **streitig**, streiten, streitentig, Streitenig, streitenten, streitenit
eng roller | ed	rollered	0	**rollered**, rolled, roller, rolleded, rollired, rollerer, rollirer, rollered, rollered
eng sailor | ess	sailoress	-1	sailess, sailiess, sailaess, Sailess, sailesss, sailessss, sailessse, sailesssss, sailesssse
eng salesperson | ship	salespersonship	1	salespership, **salespersonship**, salespersonaship, salespershipship, salespershipa
eng shining | ness	shiningness	0	**shiningness**, shininganess, shining ness, shining-ness, shininginess, shininganesss
eng spaceless | ly	spacelessly	0	**spacelessly**, spacelessily, spacelessaly, spacelessoly, spacelessli, spacelessilys
eng spae | craft	spae-craft	-1	spaecraft, spae craft, spaecreft, spaecrift, Spaecraft, spae crafto, spae crafta, spae craftar
eng steel | man	steelman	0	**steelman**, steel man, steeliman, steelaman, steel-man, steel manman, steel manmo
eng street | ness	streetness	0	**streetness**, street ness, streetaness, streetaress, street-ness, street nesss
eng subselect | or	subselector	0	**subselector**, subselectior, subselectaor, subselectur, subselictor, subselectiorpo
eng subtle | ly	subtly	-1	subtlely, subtlily, subtlelly, subtle ly, subtle, subtlelli, subtlellis, subtlellit, subtlellito
eng sulphindigotic | ate	sulphindigotate	1	sulphindigoticate, **sulphindigotate**, sulphindigotete, sulphindigotic ate, sulphindigoteate
eng suppression | ism	suppressionism	0	**suppressionism**, suppressianism, Suppressionism, suppressionaism, suppressiunism
eng surrogate | cy	surrogacy	-1	surrogatcy, surrogaticy, surrogatecy, surrogatacy, surrogatticy, surrogattici, surrogatticyr
eng ternate | ly	ternately	0	**ternately**, ternatily, ternaly, ternatoly, ternatelly, ternatelli, ternatellis
eng theatrical | ly	theatrically	0	**theatrically**, theatricaly, theatricalyy, theatricalle, theatricalli, theatricallily
eng ulcer | able	ulcerable	0	**ulcerable**, ulcer able, ulcirable, ulcer-able, ulcerible, ulcer abler, ulcer-abler
eng uniform | ism	uniformism	0	**uniformism**, uniformaism, uniform-ism, unifonmism, uniforsm, uniformiss, uniformaismo
eng zephyr | like	zephyrlike	0	**zephyrlike**, zephyrike, zephilike, zephyralike, zephyrelike, zephyrliker, zephyrlikepo
epo sincera | e	sincere	0	**sincere**, Sincere, sincire, since, sincre, sinceri, sincerie, sincerier, sinceriere
fin	kuusijalkainen	-1	anto, callo, antor, antoro, callor, callico, callica, callicar, callicino, callicaro
fin liudentua | maton	liudentumaton	0	**liudentumaton**, liudentuamaton, liudentimaton, liudentamaton, liudentomaton
fin tukka | istaa	tukistaa	-1	tukkistaa, tukkaistaa, tukka istaa, tukkuistaa, tukkanistaa, tukka ista, tukka istaaa, tukka istaak
fin uljas | sti	uljaasti	-1	uljasti, uljisti, uljesti, uljusti, uljosti, uljasti, uljasti, uljasti, uljasti, uljasti
fin vaikea | sti	vaikeasti	0	**vaikeasti**, vaikesti, vaikeisti, vaikeesti, vaikea sti, vaikeastik, vaikeastiko
hin लगना | तार	लगातार	4	लगनातार, लगतातार, लगनारार, लगना तार, लगातार, लगनातारा, लगना तारा
ita smaltire | toio	smaltitoio	0	**smaltitoio**, smaltiritoio, smaltirtoio, smaltiretoio, smaltiratoio, smaltiritois
ita vuoto | mente	vuotamente	0	**vuotamente**, vuutamente, vuetamente, vuatamente, Vuotamente, vuotamente, vuotamente
lat vorāgō | ōsus	voraginosus	-1	voragosus, Voragosus, voraganosus, voragagosus, voragato, voraganos, voragagosut, voraganosut
mul Ustilago | mycetes	Ustilaginomycetes	-1	Ustilagamycetes, ustilagamycetes, Ustilagomycetes, Ustilagimycetes, ustilagimycetes
nld programmeren | baar	programmeerbaar	-1	programmerbaar, programmeribaar, programmerabaar, programmerenbaar, programmaarbaar
non vǫldugr | leikr	vǫldugleikr	-1	veldugreikr, vǫldugreikr, veldugraleikr, vǫldugraleikr, veldugrikr, veldugraleik, vǫldugraleik
odt te | *slītan	teslitan	0	**teslitan**, Teslitan, tesliton, teslitin, teslitum, teslitan, teslitan, teslitan, teslitan
rus *kъjь | *byti	кабы	-1	kъjebyti, kъjьbyti, kъjibyti, kъjь byti, kъje byti, kъjь bytij, kъje bytij, kъjь bytibo
rus гла́сный | ость	гласность	0	**гласность**, гласнысть, гласный, гласныйость, гласнасть, гласносте́, гласныйосте́, гласныйост
rus то́шный | твори́ть	тошнотворный	-1	тошныйтворить, тошный творить, тошнытворить, тошныіворить, тошнтворить
rus устоя́ть | -ивать	устаивать	-1	устоятивать, устояивать, устоятывать, устоятьвать, устоятувать, устоятиватьо, устоятиватьок
spa uno | ista	unista	0	**unista**, unoista, unisto, unaista, unistar, unistaro, unistare, unistarer, unistaror
swe tand | a	tanda	0	**tanda**, tandaa, tandia, tandar, tandara, tandanda, tandandi, tandando, tandande, tandandar
vie ăn | mặn	ăn mặn	2	ămmặn, ănmặn, **ăn mặn**, ăm mặn, ăn-mặn, ăn mặnn, ăm mặnn, ăm mặnna, ăn mặnna, ăm mặnno

Table 4.13: Output of the LSTM encoder-decoder component joiner on a random sample of held out test words from Wiktionary.

Lang	#	Model	Acc1	Acc10	Acc100	AccN	Ed1	Ed10	Ed100
bul	740	BF	.00	.01	.03	.10	6.52	5.00	3.87
bul	740	Neu	.00	.00	.03	.20	6.60	5.12	3.60
bul	740	Comb	.00	.00	.03	.24	6.60	5.12	3.59
gle	505	BF	.01	.02	.03	.07	6.60	4.88	3.76
gle	505	Neu	.01	.03	.08	.38	6.45	4.99	3.50
gle	505	Comb	.00	.02	.08	.40	6.48	5.01	3.52
glg	619	BF	.01	.01	.03	.12	6.10	4.46	3.38
glg	619	Neu	.00	.03	.10	.35	6.13	4.50	3.02
glg	619	Comb	.00	.03	.10	.37	6.14	4.50	3.00
mlt	235	BF	.00	.00	.01	.02	5.93	4.25	3.47
mlt	235	Neu	.00	.01	.03	.26	6.00	4.62	3.48
mlt	235	Comb	.00	.01	.03	.26	6.02	4.62	3.47

Table 4.14: Compound generation results, comparing the Brute Force (BF) and Neural (Neu) methods of compound joining, along with Combined (comb) hypotheses.

I find that the neural model substantially outperforms the brute force method while using only the top 100 hypotheses from the brute force approach. This indicates that the component joining process was lacking in the brute force approach. Specifically, this shows that the glue characters and elision in the brute force approach did not handle a large enough set of compounding processes.

I present model generations on four test languages in Tables 4.15 to 4.18. We see that many concepts are simply not compositional, as evidenced by their top recipe, which does not generalize across languages. This is especially noticeable for proper nouns, which are often phonetically borrowed rather than calqued. I will discuss model combination methods to alleviate this issue in Chapter 7. When a robust recipe exists, correct predictions are able to be generated, but they are often quite far down the list. This is due to a combination of the gold translation not following the most likely compound joining process

Concept	Top Recipe	Gold	Gold Idx	Hypotheses
Ajaccio	The Hague + condition	Аячо	-1	заслугазе,вредназе,вреднизток,заслугаз,вредназ,достоенязе,достоенизток
Buckingham Palace	Buckingham + palace	Бъкингамски дворец	-1	дворцадея,дворцлаг,дворцаула,дворцала,дворцорд,дворцопра,дворецъща,дворцодпра
Christmas Eve	Christmas + evening	Бъдни вечер	-1	иоще,поли,ии,полоще,полчак,роди,полдаже,секси,едини,полдори
Gabon	G + bon	Габо́н	-1	солот,солтоз,кравив,солоня,солтоя,солтънък,солонзи,соларе,солфин,солдобър
Grim Reaper	angel + death	а́нгел на смъ́ртта	-1	ипък,икрай,идруг,имлад,ичовек,инов,иедна,иедно,ангелзар,ипресен
Latin	Latin + language	лати́нски ези́к,лати́нски	537	кацо,гласо,гребо,гласс,власто,глася,ходски,ходя,властя,властна
Portugal	Portugal + tusk	Португа́лия	-1	снация,ибой,симе,скрай,скад,икрай,ситзъб,ссуша,скисел,сиск
Sahara	Sakha + Ra	Саха́ра	-1	ио,фирмо,ира,стегля,рото,госто,ита,ипък,нара,нао
St. Elmo’s fire	fire + that which is holy	Огън на свети Елм	-1	наче,пекя,наили,умя,ная,пожаря,пожарче,бияче,пекили,желаня
Xinjiang	new + frontier	Синдзя́н	-1	надруг,намлад,додруг,накрай,домлад,примлад,нарека,принов,придруг,нанов
bird	bird + ten	птица	10162	сюнак,наща,отие,птичи,наче,смалък,нада,сдам,криля,нас
calandra lark	steppe + lark	дебелоклюна чучулига	-1	тукшега,степшега,туклудувам,туклудория,маслошега,степлудория,тукзакачка
confectionery	sweet + diminutive suffix	сладкиши,сладкарница,сладкарство	-1	бодрия,пресния,бодрие,бялия,хубавия,бялория,хубаве,хубавна
cooking	cook + king	кухня	-1	отия,доме,сия,къще,сготвя,отие,наски,сцар,отория,счив
daybreak	day + break	зазоряване,зор	-1	огоня,освия,очас,очупя,есвия,оусетя,напът,елеко,окурс,овидя
doormat	door + mat	изтривалка	-1	футна,често,капияв,капичка,щампав,частна,капие,капиявъв,врату,вратичка
exclamation mark	exclamation + mark	удивителна	-1	виквик,реввик,часобраз,часвид,войвик,опраред,виквикам,плачвик,ералице,викознача
fax	fa + copy	факс	-1	евести,ебой,еписмо,елюбим,екопие,езнача,еброй,ебуква,еслон,еиск
hammer	ham + plus	чукам,вковавам,вкова,кова,разбивам	-1	лоши,тури,бутчук,бути,лили,около,туре,оте,буте,крили
impudent	un- + shamefaced	дързък,безочлив,нахален	-1	непък,нита,нета,неуча,недържа,неумен,нецвят,лошта,недруг,несмел
influenza	in + influenza	грип,инфлуе́нца	-1	нея,неявя,нече,сток,вток,наче,нецял,оголям,оче,сголям
kosher	kay + evil	каше́р	-1	вселош,всезъл,католош,дамзъл,стъклолош,каклош,къдезъл,къделош,дамлош,всесвой
lazy	lazy + lazy	ленив,мързелив,тежък,лени́в	10435	неия,нелош,нее,нещур,ненищ,неханш,невървя,плавния,нежалък,неслаб
mercenary	hire + soldier	нае́мница,нае́мник,нае́мничка	-1	власто,властя,стрелко,служба,властс,подписо,властче,дама,власта,подписс
nineteen	ten + nine	деветна́йсет,деветна́десет	-1	сглас,снам,снас,скаца,данас,занас,иглас,занам,данам,стон
obtuse	blunt + use	тъп	23297	неия,оски,овъже,сия,сшнур,скурс,ошнур,нески,невъже,наски
ocelot	cat + lot	оцелот	-1	котбая,катвам,коткоте,котсъдба,коткот,коткотка,коткотак,котдоста,котучаст,коткабая
oystercatcher	oyster + magpie	стридояд	-1	пясъкдом,стриднеин,стридиск,стридбой,стридсвой,стридазло,морскичас,стридчас
pitch-black	dark + black	черен като катран	-1	черчер,зловра,черчерен,мракчер,черчерно,тъменчер,тъмачер,вакълчер,чернегър
prime minister	first + minister	премие́р,мини́стър-председа́тел	-1	щатшеф,щатбос,шефнос,шефпоп,боспоп,първопоп,простнос,боснос,вождпоп,носнос
rapeseed	rape + seed	рапично семе,рапица	-1	тукза,туче,бялза,масле,маслос,тукс,масличка,бялас,маслоза,мазнине
reliability	reliable + -ness	надеждност,наде́ждност	95	боия,иския,боие,вървия,доверие,екшъния,върврия,иские,действия,искория
snooker	marble + away	сну́кър	-1	тао,топчо,дана,зарадо,отна,изо,порадо,топчу,дав,топчев
strikebreaker	strike + breaker	стачкоизме́нничка,стачкоизме́нник	-1	нефут,негол,немеря,нецел,нерод,бияфут,нестълб,невид,неручей,недело
survey	upon + vision	анкета	-1	отия,сия,оство,огърди,отие,овизия,оцица,отория,сория,наство
virginity	virgin + -ness	де́вственост	-1	силия,момие,мощия,момия,страния,жудия,девичие,девичия,момория,целиния
voter	vote + -er	избира́тел,гласоподава́тел	-1	иски,ио,емъж,смъж,смъжки,тао,емъжки,снация,скрия,гласо
white	white + -ish	бял,благороден	10528,-1	напо,отия,сия,смъж,бяля,колия,кове,пос,ковия,наски

Table 4.15: Compound generation of unknown Bulgarian words.

(concatenation), or that the gold compound does not follow the universal recipe learned from all languages. However, this result is not discouraging. Because around a quarter to a third of test words exist somewhere in the combined hypothesis list, they would be able to be identified by a language model once monolingual text is available for the target language. Chapter 7 presents another method for compensating for non-compositional test concepts.

Concept	Top Recipe	Gold	Gold Idx	Hypotheses
Byzantine	shuffle + wine	Biosántach	None	suaithín,boscáilmhar,boscáilscil,boscálacht,boscálóir,boscáilfíon,boscáilnó,suaithleis,suaithscil
Israel	after and before + Israel	Iosrael,Stát Iosrael	None,None	seaghní,seaná,seam,sealé,calla,blastón,deasa,sealúth,blasta,fíora
aloe	scarlet + open	aló,fóifíneach	None,None	ódóigh,óó,anoscail,áfollas,ailmbain,scealple,fichbain,péintbain,áó,ailmoscail
confidence	self + trust	muinín,iocht,urrús,dóigh,iontaoibh,urrúsacht	None,None,11658,12546,None,None	fadó,asó,ógó,asdé,asmhar,úrmhar,úrra,asra,ógacht,féindé
decade	ten + year	deichniúr	None	asá,asó,asdhá,cáá,asle,aró,cáó,asdó,arle,asdís
geometry	measure + -logy	geoiméadracht,céimseata	None,None	tráthró,áitró,líonró,caseolas,snámhscil,meálógacht,lámhlógacht,meástaidéar,toiseléann,líonlógacht
ibuprofen	cloth + fen	iobúpróifein	None	Eábheanach,bréideanach,spréeanach,faisnéiseanach,éadacheanach,úsáideanach,sceitheanach,sraitheanach
ink	water + black	dúch	12709	assú,ísú,aslíon,dúmhéar,dúchiar,dúghorm,asleis,asáras,báighsú,asbior
inter-	- + re-	idir-	None	aos,aros,réi,éach,míra,ai,ari,lárra,ara,asmhar
linen	flax + cloth	líon,línéadach	10323,None	foró,athlíne,líonró,folíne,bunéadach,líonnead,líonstór,líonscáth,líonlíne,líonábhar
liquidity	liquid + -ity	leachtacht	19	éascacht,tapach,éascach,tapacht,éascín,glasach,leannstát,lionneagar,lionnacht,leannacht
long time no see	long time + see	is fada ná faca thú	None	fadó,óach,fadach,fadógó,binnó,ciandá,crágó,ósea,ódóigh,cianámh
navy	sea + military	cabhlach,dúghorm	None,1	gealó,dúghorm,míthoit,linnarm,ceapbrí,snámharm,fonnaire,uchtaire,capalló,muirarm
negative	negative + negative	diúltach,claonchló	10295,None	míscór,míchlós,míscor,míscéimh,míchóir,mímí,mísceall,míléamh,míléacht,míchead
older brother	large + brother	deartháir mór	None	mórí,ardard,ceapard,ceannard,aireard,móraire,fiafear,tiarnaire,ardaire,ceapbarr
orbit	around + bit	fithis,spéir	None,12305	óré,áré,aá,cróá,aí,aró,aród,aré,alúb,cróré
sentence	sentence + part	abairt	10854	asle,aslog,asfód,asóráid,asáit,aslíon,aslámh,aslann,aschun,asruta
supply	supply + -ing	lón,riar	None,None	asréim,asach,asfís,asis,arréim,assaol,asdlí,astráth,asofráil,asbun
turnip	white + beet	tornapa	None	casú,báná,gealó,ardá,fionnó,gealá,bánacht,bánó,caschló,barrú
upper arm	upper + arm	brac	None	tacarm,tógarm,cúlarm,uaslámh,oilarm,tóglámh,arlámh,aislámh,lámharm,arghéag

Table 4.16: Compound generation of unknown Irish words.

Concept	Top Recipe	Gold	Gold Idx	Hypotheses
Ares	A + s	Marte,Ares	None,None	asi,anun,aen,ana,asur,asuan,lanuns,cativín,lanel,lasi
Friday	Friday + day	venres,sexta feira	None,None	era,oura,roxa,solta,calma,inza,sema,limpa,loura,loira
Independence Day	independence + day	día da independencia	None	ceibidade,diahora,diacarallo,diacona,corodía,americomedio,diafoder,soberanidade,Diadía,soberanivagar
Latin	Latin + language	latín	None	asen,anun,aino,actués,latinés,aho,oen,aen,ana,romanés
Pangaea	Pan + continent	Panxea	None	tien,era,tie,tempa,tina,padia,tuna,tinas,tempe,tinos
Saudi	Saudi + Arabic	saudita	10424	ti,unoso,coi,aboi,arei,apisar,unhoso,apoñer,cochan,oandar
Spanglish	English + Spanglish	spanglish,espanglish	None,None	inglestilla,engrestilla,inglestenda,engrestenda,inglesrocho,inglesdepósito,inglesceleiro
almost	almost + little	case,por pouco	None,None	alga,xunta,penza,guia,brava,linda,cerca,beira,fecha,aspera
annual	year + -ly	anual	13916	uniño,unés,anal,anoso,anera,anosa,aniño,anaño,anano,unoso
assign	toward + sign	outorgar,asinar,asignar,designar	None,10100,15905,None	aben,aillar,aobrar,amandar,aoso,porben,apoñer,adeixar,alevar,aorde
asylum seeker	asylum + asylum	solicitante de asilo	7354	aman,coman,oandar,lapeiro,aista,manista,abuscar,mancata,agorir,coasilo
bisexual	two + sexual	bisexual	13962	berroso,mesmiño,mesmura,mesmeza,mesmoso,douseza,mesmosa,mesmento,doussexual,dousal
caesium	cee + i	cesio	10227	cevez,moiti,cesi,carri,cetempo,torri,cevagar,cetres,gralli,ceabra
carefully	careful + fully	coidadosamente	None	corda,longa,pora,cauta,lenta,larga,calma,picha,aben,aillar
claw	claw + bread	gadoupa,uña,coca,garra	43,14,None,17444	torna,unou,fonda,pina,pata,unha,uñuña,peza,cacha,birla
confectionery	sweet + diminutive suffix	repostaría,confeitaría	None,None	doceza,doce,dociño,docura,doceira,meleire,doceiro,meleiriño,doceito,meleireza
disarmament	arm + armament	desarmamento	None	deseixe,ourever,amilitar,desrever,decompaña,demilitar,ahoste,detropas,ohrever,detropa
enter	inside + go	entrar	None	unhun,porun,aun,abulir,aguiar,apisar,adurar,unun,empegar,afoi
fart	fart + wind	peideiro,peido	None,12809	arar,araire,vellar,vedrar,peidar,lonxar,remotar,cativar,bufarte,ardobar
fathom	fat + om	calar,abrazar	None,10059	aman,amirar,porman,aollar,aobrar,agañar,empegar,amandar,afillar,alevar
frog	frog + child	gavacho,ra	None,None	parar,ahome,aoso,ameniño,alombo,apitar,apescar,apeixe,asilbo,amultar
go away	away + go	partir,tirar,saír	10978,None,None	aben,porun,amirar,desben,aun,amudar,aollar,apasar,abulir,adurar
hyponym	bottom + word	hipónimo	None	xuró,xuroh,borrés,xurah,xurou,cunome,cutermo,cufala,cuprazo,petermo
liberate	free + release	liberar,ceibar	None,12087	senceibo,libri,sendo,senceibe,sensen,sentalla,sengañar,senceibar,sensolta,sensolto
mortality	mortal + -ness	mortalidade,mortaldade	15,25	mortiño,mortura,morteza,mortera,estrelín,estreliño,pasamentiño,mortaliño,obitiño,obitera
nasalization	nasal + -ize	nasalización	None	poxa,nasa,tata,lura,napia,bica,crica,caba,nasizar,fociña
necktie	neck + tie	gravata	None	palló,gotelo,colelo,palliño,palloh,goteixe,corvín,corviño,pallah,coleixe
negative	negative + negative	negativo	10866	deslei,descontar,desxuro,desmocear,desdereito,negativaza,desnegativo,destribunal,leronegativo
now	present + time	agora,actualmente	23,None	inda,ista,esta,nina,-eira,-ista,oura,desda,denda,loga
ogre	raw + animal	coco,orco,urco,papón	None	inaño,varés,berrés,desaño,varino,bruiño,asperaño,hoho,homeu,eideiro
parcel	small + package	parcela	None	iñiño,lumiño,lumeza,lumura,cativazo,benteor,acendeza,cativeza,prendura,acendura
penance	pen + line	penitencia	None	cua,fita,fera,conta,pora,tira,crica,baixa,quera,porable
pick	upon + pick	abranguer	None	asi,cosi,aben,aillar,amirar,aollar,coler,empegar,afillar,porben
regiment	regime + month	bandeira,rexemento	None,None	fora,pora,xira,volta,regra,quenda,baixa,chumba,xeira,media
saw	saw + saw	tronzar,serrar	None,3	parar,porar,amirar,serrar,aollar,agañar,atallar,pitorno,acamiñar,alanzar
sceptre	king + evil	cetro	None	reimal,reipegar,reicana,reicolar,reimao,manaveso,mancana,mamamal,mantirar,manmal
shears	two + knife	tesoira,tesoiras	None,None	doustopa,duaño,dousde,dousseda,dousorde,doustrazar,dousrodal,irmandado,doustrocha,doustresna
span	chip + yes	palmo	None	acha,liña,popa,roda,corda,baña,fia,talla,cana,posta
underwater	beneath + water	submarino	None	demar,demalado,augauga,pemalado,cuauga,humillarco,baixamar,porabaixar,baixauga,subauga
urgent	urgent + urgent	urxente	None	inal,intemer,antal,destemer,antitemer,impequeno,desal,librapurrir,despequeno,libratrigar
vector	century + torus	vector	None	amedir,aformar,ahoste,amodo,adourado,oudourado,acurral,acorpo,seculouro,afumeiro

Table 4.17: Compound generation of unknown Galician words.

Concept	Top Recipe	Gold	Gold Idx	Hypotheses
Brunei	Brown + fallow field	Brunej	None	bik,bilil,bixejn,malma,bilanqas,bi-imma,bi-int,mala,bi-mhux,biebda
Chile	chi + C	Ċilì	None	kielni,mijani,ragelni,mija jew,żewgni,mitt-hinn,mitt-hemm,mitt-jew,kielwieta,kiel-jew
Chinese	Chinese + -ese	Ċiniż	None	nofsuż,nofsiuż,fustaniuż,nofsi jekk,nofsi jew,nofsi ebda,nofsi stat,nofsi kieku,nofsxorta
Maltese	bad + -ese	(il-)Malti,Malti	None,10165	deniuż,Maltiuż,denixorta,denixabla,għaseluż,Maltixorta,Maltixabla,deni jekk,deni tena,deni tavla
New Zealand	new + Zealand	New Zealand	None	ażotu qasir,ażotu Alpi,buttuna qasir,frisk qasir,frisk Alpi,buttuna Alpi,najtrogin qasir,gdid-qasir
Palestine	Pale + bid	Palestina	None	miżienla,xeraqla,ferrex tari,segwa tari,ferrexla,stqarr tari,xeraq tari,segwa ma,pajjiżla,segwa la
Revelation	revelation + record	l-Apokalissi	None	xerqa,kixefa,arja u,skieta,siekta,xandar u,kebbesa,lehema,wicc a,tidwila
Russian Federation	Russian + federation	Federazzjoni Russa	None	kieluż,Russu patt,Russu stat,Russuż,Russu paci,Russu gab,Russu dehen,Russu stqarr,Russu ftehim
Saint George	Saint + George	San Ġorġ	None	reqatel,santu dewwa,santu duwa,santu sema,qaddis duwa,qaddis qatel,resaltan,qaddis regola
alms	give + golden	limosja	None	ala,akiel,ama,amhux,akarità,axemx,aram,fi-ma,anamra,aborma
aloe	scarlet + open	sabbara	None	fi-u,ailu,axebba,abint,aneputi,abi,anom,atifla,ago,aqolla
anniversary	year + day	anniversarju	None	maera,maepoka,magab,manhar,magurnata,masena,majum,mażmien,jum jum,senaxemx
architect	architect + -er	arkitett	None	ras dar,ras re,ras madam,ras ras,fassal re,kap ras,fuqani dar,ras mindu,ras bejt,binja re
belt	waist + belt	ċinturin	None	relok,rekapa,fuqħal,rekap,fuqħand,remadam,repogga,remkien,re-ras,qadd uman
bet	out of + goal	mħatra	None	fi-u,akif,bixemx,akejl,alok,atriq,axemx,bixkaffa,axkaffa,aqies
bird	bird + ten	tajr,għasfur,pizu	14725,None,None	rixa,fi-u,maa,mama,axita,ama,maxita,fuqu,mahuwa,fi-a
contract	contract + act	kuntratt	None	fi-ittra,fi-patt,bifażi,bilingwa,bilsien,fi-ktieb,fi-parti,biftehim,biżmien,bipatt
cooking	cook + king	sajra,sajran	12090,None	brodu re,malre,kokkoka,kokkok,soppa re,ikel ilma,brodu sultan,malsultan,ikel re,kok re
coronavirus	crown + virus	koronavirus	None	ak,abagg,akaskata,aint,qalbk,aintom,ainti,qalba bagg,aintkom,qalba Russu
disperse	one + scatter	xerred	None	mindu dawra,ukollesta,fi-miskin,fi-dawra,fi-fqir,fi-povru,ukoll tarf,anke-mmisja,ukoll tilef
enter	inside + go	daħal	None	ilu dam,ilu mqar,ilu biex,ilu jtul,fuqħola,ilu anke,ilu pogga,fuq sar,ilu wkoll,fuq tarag
fart	fart + wind	fiswa	None	maarja,mabass,qadim bass,qadim arja,mail-,laarja,xieref arja,bass-daqq,mxarrab arja,kbir tarf
fortnight	two + night	ħmistax-il ġurnata	None	erbataxax,nofsax,tnejnax,erbataxa,lejla,sekonda,erbatax bla,hekka,erbataxxemx,tnejnlil
freezer	ice + cupboard	friża	None	toqba re,toqba ma,parka,toqba u,frisk omm,frisk mamà,frisk u,fonda,barda,toqbaa
frying pan	roast + pan	taġen	None	sa u,salil,stad ilu,biex tagen,salewm,sahi,saliżar,sahija,stad borma,saliem
full moon	full + moon	qamar kwinta	None	rix qamar,mimli fuq,mimli qamri,mimli qadef,qamar qamar,mimli qamar,mimli qadfa,mimli dwar
instead of	preceded by and followed by + instead	flok,minflok	17793,None	mama,mamitt,fi-iva,magab,fi-ma,malok,mailu,mamija,maqaleb,mamejda
interaction	mutual + action	interazzjoni	None	ilu lok,fi-mawra,fi-mkien,użazzjoni,ilu mkien,bidla mawra,fi-magra,dwarlok,fi-pogga,ilu pogga
knowledge	know + -ness	għerf,għarfien,għelm	None,None,None	elfuż,lokuż,jafuż,rauż,dehniuż,raqatt,triquż,fehemuż,ramqar,raxoffa
linen	flax + cloth	kittien,għażel	None,None	drappu,abjad ebda,drapple,xoqqa karta,drappla,abjad le,abjad xoqqa,xoqqa mhux,abjad drapp,tnejnlil
nationalism	national + -ism	nazzjonaliżmu,nazionalismu	None,None	poplu tifsira,nazzjon beka,nazzjon tar,nazzjon barra,gensfar,nazzjon fidi,nazzjon reqqa,nazzjon far
necktie	neck + tie	ingravata	None	ras a,flus a,gerżuma,serpa,kap a,ram a,ras banda,qalba,gerżuma rabat,tiben rabat
over	upon + per	fuq	10328	fi-u,mus a,afuq,bilil,ailu,fuqu,fi-a,axifer,afi,axoffa
pullet	pull + let	għattuqa	None	ram a,tawra,traba,tifel a,gendus a,gibedlil,ftit-tarbija,tikka-a,fellus a,gibed bla
regiment	regime + month	riġment	None	akbir,anumru,akejn,aqadfa,axahar,aqamar,aqasba,agabra,aballun,aqadef
scratch	scratch + scratch	barax	11058	faxxa,fi-u,fuqu,fi-a,baraxa,fuqa,ilu u,bi-u,fi-abjad,barxa re
seed	seed + seed	żerriegħa	None	rani,raxitla,raxorta,rafuq,railu,rawild,ralil,ragrad,rakulma,rabla
sentence	sentence + part	sentenza	None	fi-a,fi-iva,fuqa,ilu lok,quddiema,widna,fi-mkien,fi-sid,mindu a,ilu a
sherbet	ice + bet	sorbè	None	hixandar,soru papra,frott alla,hi-meraq,xarba meraq,hikixef,xarba papra,hipapra,hiallat,hi-alla
shovel	shovel + written in the Latin script.”	luħ,pala	None,None	sieqa,xiexa,kiefera,daqlı,deciża,daqu,pedala,moqdiefa,siequ,sod a
single	one + married	fard	None	mament,lament,maa,blament,mhuxa,xemxa,unità,mauż,fuqa,bla-a
south	half + day	sud,nofsinhar,qibla,t’isfel	None,245,None,None	triqa,kliema,mkiena,kelliema,speakera,bniedema,nofsi jum,loka,fomm a,spikera
span	chip + yes	xiber	None	naqqaxa,fi-a,fi-iva,naqaxa,fi-ukoll,fuqa,huma,daqqa,fi-kap,fi-anke
stick	stick + stick	bastun,ħatar	10504,10842	baxxa,id-a,hemeżlil,hemeża,kejna,idlil,injam a,ikel a,boska,bniedema
stink	stink + bad	niten	None	ilu u,sar u,telaqu,spirtu u,xark u,ilu pogga,sarxamm,waraxamm,xandar u,lejn u
tense	time + e	temp	None	akif,axorta,agab,afigura,arota,aforma,amindu,afassal,agawhra,afawwara
thousand	thou + thousand	elf	10786	int-u,intom u,gniena,-ka,intom a,elf-a,intkom u,inti-u,tnejnlil,certa
what	like + written in the Latin script.”	liema	33	mament,maa,ufi,mamqar,mailu,mau,mauż,maanke,mahekk,u a
yell	shout + shout	għajjat	11259	fi-u,fuqu,fi-a,xxewwexa,fuqa,ilu u,fi-ordna,fi-amar,fi-re,sena

Table 4.18: Compound generation of unknown Maltese words.

4.1.5.4 Compound Generation in Practice: A Small Human Evaluation

To test the compound generation model in practice, I perform a small human study with compounds generated into Chinese by the model. In Chinese, any word with more than one character can be considered a compound word. However, many words in Chinese may also not be compositional, e.g. borrowed words are phonological, even though they are composed of multiple characters. Thus, Chinese exhibits multiple processes on which we can test the model.

I recruited a native Mandarin Chinese speaker to predict the translation of twenty test concepts (18 randomly selected, plus HOSPITAL and CORONAVIRUS), given a 10-best list of compound hypotheses, shown in Table 4.19. The annotator was asked to guess the translation, judge how easy it was to guess the translation (easy/medium/hard), identify which hypotheses would be intelligible as a translation by other native speakers (marked in **bold**), and identify which hypotheses were actual Chinese words (underlined). Hypotheses marked in both bold and underlined are correct translations. Results from this study are shown in Table 4.19.

The annotator was quite surprised at the solid performance of the model. Below are brief comments regarding each test word.

HOSPITAL and CORONAVIRUS were chosen because I have worked extensively with these two examples. The model generates understandable compounds in first rank, which is satisfying to see. The correct translation is 医院 ‘medicine institution’. For CORON-

10-best Hypotheses	Gold	Annotator's Translation	Difficulty
病家, 病房, **病室**, 惡家, 不好家, 惡房, 惡室, 不好房, 不好室, 毋鬆爽家	hospital	hospital	easy
冠病毒, **冠电脑病毒**, **冠電腦病毒**, **冕病毒**, 頭上病毒, **齒冠病毒**, 錦標病毒, **齿冠病毒**, 镶假齿冠病毒, 頭殼頂病毒	coronavirus	coronavirus	easy
子黃, 子華, 子黃色, **蛋黃**, 春黃, 蛋華, 蛋黃色, 春華, 春黃色, 子火	yolk	egg yolk	easy
調上, 鍵上, 調板, 調起, 匙上, **鍵板**, 鍵起, 匙板, 匙起, 調眼	keyboard	keyboard	medium
二十, 二拾, **雙十**, 雙拾, **兩十**, 兩拾, 二呀, 雙呀, 兩呀, 二萬乘	twenty	twenty	easy
美國, 美邦, **美國家**, 美國國, 美国國, **美國邦**, **美國國家**, 美国邦, **美国國家**, 美白	United States	America	easy
鐵道, **鐵路**, 鐵川, **铁道**, **铁路**, 燙道, 燙路, 铁川, 燙川, 鐵法	railroad	railroad	easy
書架, **書架子**, **書架仔**, 開架, 開架子, 開架仔, 著架, 著架子, 著架仔, 書停放架	bookshelf	bookshelf	easy
弗蘭克西, **法蘭克人西**, 法兰克人西, 弗蘭克右, 法蘭克人右, 法兰克人右, 弗蘭克西方, 法蘭克人西方, **法兰克人西方**, 弗蘭克金	France	France	easy
巴打, 合打, 可打, 巴格, 合格, **巴箱**, 巴套, 合箱, 合套, 巴盒	suitcase	suitcase	hard
生槳, 生橈, 生桨, 上槳, 上橈, 上桨, 精槳, 精橈, 精桨, 極槳	extreme	oar, ginger	hard
明星日, **金星日**, **太白日**, 明星天, **维纳斯日**, 黃昏星日, **維納斯日**, 太白星日, **金星天**, 太白天	Friday	Venus	easy
屏包, 屏布, 屏紙, 廁包, 廁布, 廁所包, 厕所包, **廁所布**, 馬桶包, **厕所布**	toilet paper	toilet paper	medium
底下行, **之下行**, 下頭行, 下跤行, 腳下行, 下背行, 以下行, 下面行, 下爿行, 下首行	underline	underground railway	medium
疑確, 疑确定, 疑確定, 疑當然, 疑肯定, 疑当然, 疑好阿, 疑沒問題, 疑板上釘釘, 疑一準	doubtless	certainly, doubt, no doubt	hard
字一, 字個, 字乙, 字寡, 字其, 字幺, 條一, 字一個, 字匹, 字1	slippery	word one	hard
後喪, 後屍, 後大體, 後死人, 後鹹魚, 後屍體, 後殭屍, 後尸体, 後屍骨, 後遺容	backwards	dead body, dead face	hard
星煙, 星露, 星霧, **星霞**, 星煙霧, 星霄, 星靄, 星霧水, 星雲氣, 星雰	nebula	stardust	medium
新修, 新補, **新整**, 新彌, 新代謝, **新更新**, **新維修**, **新收拾**, **新修補**, **新修理**	renovate	renovation	easy
冷火, 寒火, **冷戰**, 淡火, 凍火, 涼火, 凝火, 森火, 冷塵, **冷戰爭**	Cold War	cold war	easy

Table 4.19: Results on a human study of generated Chinese compounds. Bold indicates words that are intelligible translations. Underlined words are actual Chinese words.

AVIRUS=crown+virus, the translations of crown have multiple senses: the crown on the head, as well as the crown on a tooth. Nevertheless, the annotator rated these as understandable. The correct translation is 冠状病毒 'crown-shaped virus'.

YOLK was very easy, with 蛋黄 'egg yellow' being the actual correct translation. Similarly, TWENTY as 二十 'two ten' and UNITED STATES as 美国 'beautiful country' are the actual translations.

RAILROAD generated several correct translations: 铁道 'iron way' and 铁路 'iron road', and their counterparts in traditional characters. The dictionary lists 铁道 as 'rail track' while 铁路 is 'railroad'. The annotator informed me that the former is more common in northern Chinese speakers, while the latter is used by southern speakers. Both are acceptable translations for RAILROAD.

FRANCE did not get translated compositionally, but rather phonetically. The first-rank hypothesis is 弗蘭克西 *fu lan ke xi.* In the second-rank hypothesis, 法蘭克人西 *fa lan* *ke ren xi* 'Franks people west', 法蘭克 refers to the Franks, a group of Germanic people from which the word *France* is derived. The annotator believed that 西 was a mistake that native speakers would ignore. The correct translation is 法国 'law country'.

SUITCASE was difficult to identify, with 巴 *ba* being a major distractor. In the hypotheses, 箱 'box, trunk, chest' and 盒 'small box, case' allowed the annotator to guess SUITCASE as a translation. 巴格 *ba ge* may be a phonetic transcription of *bag*, but this was not noticed by the annotator. The correct translation is 箱子 'box diminutive'.

EXTREME and SLIPPERY were not able to be accurately generated by the model. EXTREME did not have a compositional recipe. SLIPPERY's recipe was not robust. The most probable recipe is slip + one, and 字 is an (inaccurate) translation of "slip".

FRIDAY is an interesting case. Across all the world's languages, Chinese one of the few languages where Friday is 'week five'. More common is 'metal/gold day' in Asian languages, and 'Venus day' in Romance languages. Thus, the annotator believed that *Venus* was the intended word. The correct translation is 星期五 'week five'.

TOILET PAPER as 廁所布 'toilet cloth' was only able to be found by looking through the entire n-best list. The correct translation is 卫生纸 'hygiene paper'.

DOUBTLESS was confusing to the annotator. 疑 'to doubt/suspect' is essential to the meaning of the compound, but the annotator remarked that this word is ambiguous, because doubt and suspect are antonyms.

BACKWARDS's recipe was also not robust. The most common recipe was back+corpse.

NEBULA as 星煙 'star smoke/vapor' is reasonable, though the annotator guessed that this meant *stardust* rather than *nebula*. The annotator remarked that this test example was revelatory and caused her to think more deeply about how new words were formed in her native language. The correct translation is 星云 'star cloud'.

RENOVATE as 新修 'new decorate' is also quite reasonable. There are many correct translations for renovate. The annotator prefers 修缮 'decorate repair'.

Finally, COLD WAR as 冷戰 'cold war' is a correct prediction, but the annotator did not guess the translation until reading through the entire n-best list.

In summary, this user study shows the potential application of the compound generation model. Though not perfect, the compound model's hypotheses are recognizable, and more importantly understandable, enabling communication with a speaker of an unknown language. Intelligibility is increased when showning a n-best list, where hypotheses of lower confidence can lead the speaker to get the gist of the meaning through a constructed compound, even if not generating the correct native word.

4.2 Translation via Lexical Relations

In this section, we present another recipe-based translation method in the English-foreign direction that does not require an external machine translation system. The main motivation behind this method is that if one does not know a word in a language, one can use a known related word. Humans do this all the time; this is called circumlocution. Suppose a child who does not have a fully developed vocabulary is trying to express a concept but does not know the word. How would they describe it?

This type of translation is fundamentally different from the previous cognate and compositional models. The previously proposed models generate candidate translations that we have never seen before, and we ask, is this a valid word in the language? On the other hand, in the process of translation via lexical relations, we ask, is this existing word an acceptable translation of another word?

In order to obtain related words, I utilize WordNet (Fellbaum, 2010), a freely-available lexical database of English words. I specifically focus on four types of lexical semantic relations: synonyms, hypernyms, hyponyms, and co-hyponyms. Synonyms share the same meaning. Hyperynms and hyponyms comprise the *is-a* relation, where the hypernym is the supertype (e.g. melon) and the hyponym is the subtype (e.g. watermelon). Co-hyponyms are words that share the same hypernym. Because these relationships are stored in WordNet at the synset level, rather than at the word level, a pair of words may be linked by more than one relation. For example, *dog* is both a synonym and a hypernym of *hound*. These lexical semantic relationships are illustrated in Figure 4.18 using the concept of HOUND.

We wish to find a particular language's word for HOUND without cognate or compositional models available. What can we do with no other bilingual resource but a small dictionary? In English, the word *hound* is used to indicate a hunting dog, so we can intuitively say that *dog* is a perfectly valid replacement for *hound*. Moreover, it is more likely that the word *dog* exists in the dictionary than *hound*, because *hound* is a more specialized

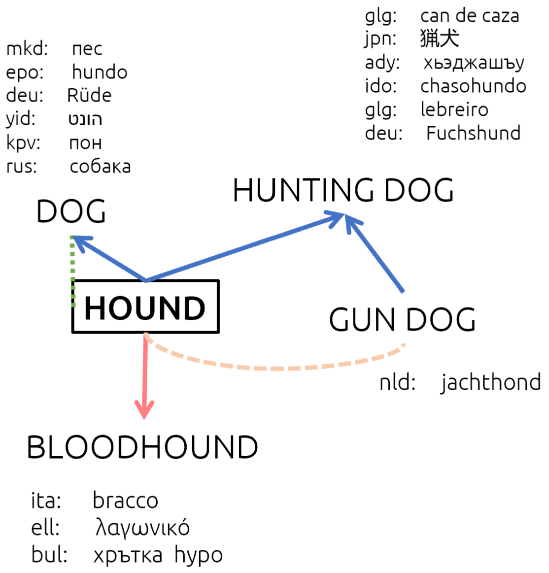


Figure 4.18: Concepts related to HOUND and their corresponding translations in various languages.

word and thus ranks lower in terms of coreness.

To develop a model of translations of related concepts across languages, I translate every English word e in Wiktionary into all other languages and then back into English to obtain a set of back-translations e_{rel}. I then look up each $e \rightarrow e_{rel}$ pair in WordNet to identify the lexical relation (synonym, hypernym, hyponym, and co-hyponym). From these pairs $e \rightarrow e_{rel}$, I compute a probability distribution $p(e_{rel}|e)$ that describes the likelihood that e_{rel} is an acceptable replacement translation of e.

4.2.1 Experiments

I evaluate this model on the task of generating translations from English into a foreign language. Instead of $e \rightarrow f$, this model translates $e \rightarrow e_{rel} \rightarrow f$, reminiscent of translation

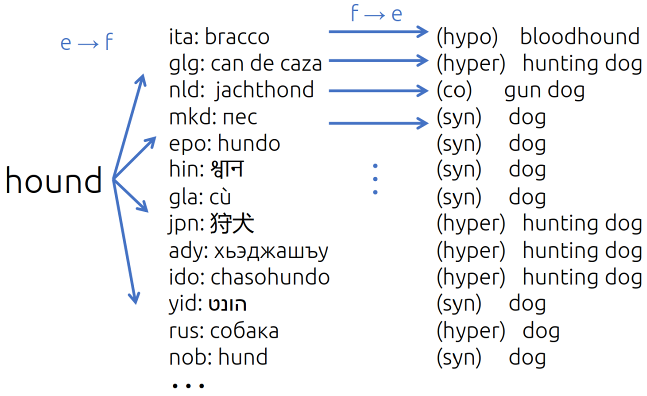


Figure 4.19: Process of computing the probability distribution for the concept HOUND. This involves aggregating the back-translations of the original concept filtered by the lexical relations in WordNet.

e_{rel}	$(e_{rel} \mid e)$
dog	0.54
hunting dog	0.13
gun dog	0.07
bloodhound	0.06
greyhound	0.03
foxhound	0.02
...	...

Table 4.20: Top several translation by lexical relations of HOUND.

Lang	# Test	1-best	10-best	n-best
bul	739	.12	.30	.38
gle	502	.11	.25	.29
glg	617	.10	.22	.31
mlt	234	.14	.26	.27

Table 4.21: Lexical relation translation, all test concepts.

Lang	# Test	1-best	10-best	n-best
bul	412	.21	.54	.69
gle	239	.23	.53	.61
glg	333	.18	.41	.57
mlt	106	.30	.58	.60

Table 4.22: Lexical relation translation, only test concepts that exists in WordNet.

bridging. I evaluate my translation model on the same test set presented in Chapter 7.

Overall results are shown in Table 4.21. I report 1-best, 10-best, and n-best accuracy (whether the gold appears in the top 1, 10, or the entire list). We immediate see that this simple technique shows remarkable performance without any neural model and just a bilingual dictionary plus WordNet. Since WordNet only covers roughly half the concepts in the test set, we also report performance on a subset of test concepts that exist in WordNet in Table 4.22.

I examine several model predictions below. Table 4.23presents Irish predictions. For example, when the Irish words for REMEDY (*leigheas, neart, íoc*) were held out, the model was able to apply the lexical relations REMEDY → MEDICINE, CURE, ANTIDOTE, which did exist in the dictionary, allowing the model to produce an appropriate translation of REMEDY's hypernyms, hyponyms, cohyponyms, and synonyms.

Concept	Gold	Hypotheses
single	aonartha, aonta, singil, aonarach, aonarúil	(syn) unmarried → singil 0.357
		(syn) one → aonta 0.310
remedy	leigheas, neart, íoc	(hyper) medicine → leigheas 0.363
		(co) medicine → leigheas 0.363
		(syn) cure → leigheas 0.171
		(syn) cure → íoc 0.171
		(hypo) antidote → leigheas 0.036
marsh	corcach, seascann, riasc, corrach, eanach	(co) swamp → eanach 0.480
		(co) swamp → corcach 0.480
		(syn) fen → eanach 0.085

Table 4.23: Translation hypotheses in Irish from lexical relations.

Concept	Gold	Hypotheses
she-goat	коза, коза́	(hyper) goat → коза́ 0.917
liberty	свобода́	(hyper) freedom → свобода́ 0.659
cumin	кимион	(co) caraway → кимион 0.667
gradient	склон, градиент, наклон	(syn) slope → склон 0.353
		(co) inclination → склон 0.216
		(co) inclination → наклон 0.216
		(hypo) pitch → наклон 0.098
		(hypo) grade → наклон 0.078
		(hypo) rake → наклон 0.059

Table 4.24: Translation hypotheses in Bulgarian from lexical relations.

For Bulgarian (Table 4.24), we see similar results. SHE-GOAT is a quite specific term, but since the model has learned that GOAT is the hypernym of SHE-GOAT and is an acceptable translation, and that GOAT already exists in the dictionary, the model correctly predicts *коза́*, the translation of *goat*, as the translation for *she-goat*. Caraway being translated as cumin is an interesting successful example. Although they are not the same herb, they are visually similar, and Bulgarian uses the same word for both, *кимион* (kimion). Indeed, caraway is sometimes called Persian cumin.

Concept	Gold	Hypotheses
liberate	liberar, ceibar	(syn) free → liberar 0.427
		(hyper) free → liberar 0.427
		(syn) release → liberar 0.152
		(syn) release → ceibar 0.152
		(syn) loose → ceibar 0.026
		(co) open → ceibar 0.013
quarrel	rifar, cotifar	(hyper) argue → cotifar 0.093
		(hyper) argue → rifar 0.093
azure	blao, azul	(hyper) blue → azul 0.514
claw	garra, uña, coca, gadoupa	(co) nail → uña 0.284
		(co) hoof → uña 0.123

Table 4.25: Translation hypotheses in Galician from lexical relations.

Galician (Table 4.25) also has several examples of words with subtle meanings that could easily be expressed with a more general-purpose word. For example, LIBERATE (*liberar, ceibar*) is adequately translated with FREE or RELEASE. To QUARREL is essentially to ARGUE, albeit in a heated manner. AZURE is a specific shade of BLUE.

Finally, for Maltese (Table 4.26), the lowest resourced language in the test set, we find that the translation with lexical relations approach provides the greatest benefits over the other cognate and compound models. When predicting the word for STICK, *ħatar* and *bastun*, other more specialized sticks (staff, rod, club) also get translated as STICK. Similarly, DECEIVE can be translated as CHEAT or BETRAY.

In addition to these experiments, I also examined the effects of training on only languages in the same family as the test language, versus training on the entire test set. I find that performance is *worse* when trained on all languages, for Bulgarian, Galician, and Maltese. Only for Irish did the performance increase. This is in contrast to the compound

Concept	Gold	Hypotheses
white	bojod, bajda, abjad	(co) pale ⟶ abjad 0.101
stick	ħatar, bastun	(hypo) staff ⟶ bastun 0.089
		(co) rod ⟶ ħatar 0.075
		(hypo) club ⟶ ħatar 0.052
deceive	lagħab, gidem, baram, qarraq	(hypo) cheat ⟶ qarraq 0.283
		(hypo) cheat ⟶ lagħab 0.283
		(co) cheat ⟶ qarraq 0.283
		(co) cheat ⟶ lagħab 0.283
		(hypo) betray ⟶ qarraq 0.103
		(syn) betray ⟶ qarraq 0.103

Table 4.26: Translation hypotheses in Maltese from lexical relations.

model, which I found to be strictly better when training on all the languages available. Table 4.27 shows some Irish examples in which the model trained on all languages was able to outperform the model trained on only Irish-related languages.

Why would training on more languages reduce performance? I found that this introduces more noise. When training the compounding model,more signal from non-related languages is often beneficial, because often it is not the word itself that gets borrowed, but the recipe (this would be a calque, or a loan translation). For example, the English *brainwash* comes from Chinese 洗脑 'wash+brain', due to contact between different languages and cultures. In contrast, lexically related words are often language specific. Translating "watermelon" as "cucumber" only occurs in Italian and Romanian, and there is no reason to believe that any non-Romance language would share this translation. Indeed, other languages use "west melon" (in Chinese) or "Greek melon" (in Hungarian), which is a compositional formation recipe, but not a robust one. Nevertheless, Table 4.27 shows several instances where training on all languages allowed the model to recover translations

Concept	Gold	Hypotheses
die	éag, faigh bás, básaigh, caill	(co) decay → éag 0.007
moment	móimint, nóiméad	(syn) minute → nóiméad 0.087
now	anois, adrásta, anuas	(syn) at present → adrásta 0.150
resin	bí, roisín	(syn) rosin → roisín 0.800
empty	fásach	(co) desert → fásach 0.015
penance	aithrí	(syn) penitence → aithrí 0.233 (syn) repentance → aithrí 0.233
accumulator	bailitheoir	(syn) collector → bailitheoir 0.750

Table 4.27: Translations which Irish learned using all languages but could not using just related languages

compared to training on only related languages.

4.3 Conclusion

Many words can be formed by following certain probabilistic translational "recipes", which I have modeled with compositional and lexical relational models. One such class of words are compositional. While most languages exhibit broad-scale word formation via compounding, they often differ substantially in terms of the diverse processes by which words compound and novel concepts are realized via these compound processes. Using only freely available bilingual dictionaries and no annotated training data, we derived novel models for analyzing and translating compound words and effectively generated novel foreign-language translations of English concepts using these models. In addition, we release a massively multilingual dataset of compound words along with their decompositions, covering over 21,000 instances in 329 languages, a previously unprecedented scale which we believe will both productively support machine translation (especially in low resource languages) and also facilitate researchers in their further analysis and modeling of compounds and compounding processes across the world's languages.

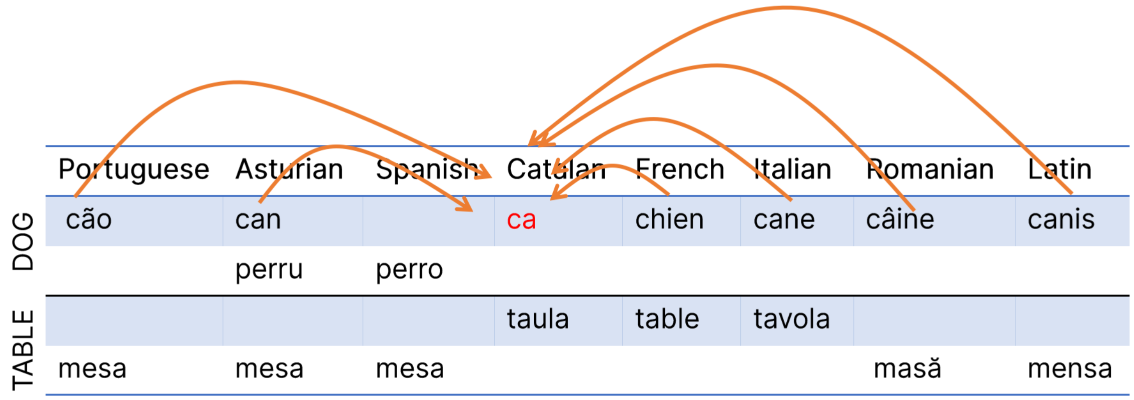
Another class of recipe-based formation is through lexically related concepts. Using only bilingual dictionary and WordNet, we accurately predict the translation of unknown words by bridging through lexically related hypernyms, hyponyms, co-hyponyms, and synonyms. This simple but effective method does not require any neural model and is especially well-suited for extremely low-resource languages for which little resources are available.

Chapter 5

Cognate and Sound-Shift Models

Low-resource languages unsurprisingly often suffer from a lack of high-coverage lexical resources. In this chapter, I propose a method to generate missing cognates or cognate-like words. First, I automatically obtain cognate tables by clustering words in existing lexical resources. I then employ character-based sequence-to-sequence methods to solve the task of cognate cluster completion. I induce missing word translations from lower-coverage dictionaries to fill gaps in the cognate clusters, finding improvements over single language pair baselines when employing simple but novel multi-language system combination on the Romance and Turkic language families.

I define the task of cognate cluster completion. In a multi-way aligned table, such as one shown in Figure 5.1, a cognate cluster is a group of cognates or cognate-like words, typically in the same language family (represented as a single row). Clusters may have empty cells due to dictionary gaps, and the task is to predict these missing entries. In



	Portuguese	Asturian	Spanish	Catalan	French	Italian	Romanian	Latin
DOG	cão	can		ca	chien	cane	câine	canis
		perru	perro					
TABLE				taula	table	tavola		
	mesa	mesa	mesa				masă	mensa

Figure 5.1: The cognate cluster completion task.

this task, any related word within the same row can contribute to the hypothesis of a missing cell. For low-resource languages, generating hypotheses for missing cognates has applications in alignment and resolving unknown words in machine translation. In linguistics, examining cognates across multiple related languages can shed light on how words are borrowed between languages.

Previous approaches to cognate transliteration (Mulloni, 2007; Beinborn, Zesch, and Gurevych, 2013) suffer from the drawback that they require an existing list of cognates, which is infeasible for low-resource languages. In contrast, I automatically generate cognate tables by clustering words from existing lexical resources using a combination of similarity measures. Using these cognate tables, I construct multi-way bitext and train character-based machine translation systems to transliterate cognates to fill in missing entries in the cognate chains. Finally, I evaluate multiple methods of system combination on the cognate chain completion task, showing improvements over single language-pair systems. For the Romance languages, I find that performance-based weight outperforms combining weights derived from a linguistic phylogeny.

This chapter includes work originally published in Wu and Yarowsky (2018b), Wu, Vyas, and Yarowsky (2018), Wu and Yarowsky (2018a), Wu, Nicolai, and Yarowsky (2020), Wu and Yarowsky (2020a), and Lewis et al. (2020).

5.1 Automatic Cognate Clustering

In order to train cognate generation systems, models require aligned cognate lists. However, cognate lists are not widely available for many languages and are time-consuming to create by hand. In many NLP-related applications, including the translating out-of-vocabulary words in machine translation, it is often not necessary that these words be true cognates in the linguistic sense, i.e. they are descendants of a common ancestor. For example, names and loanwords are not technically considered cognates, though they behave as such. Rather, “cognates” only need to meet certain established criteria for cognacy (Kondrak, 2001; Inkpen, O. Frunza, and Kondrak, 2005; Ciobanu and Dinu, 2014), which include individually or a combination of orthographic, phonetic, and semantic similarity between words.

I extract foreign-English translation pairs for all languages from two of the largest multilingual dictionaries, PanLex (Baldwin, Pool, and Colowick, 2010; Kamholz, Pool, and Colowick, 2014) and Wiktionary. To generate multilingual cognate tables, I employ an automatic method of clustering words from lexical resources. In contrast to Scherrer and

Sagot (2014), who compare entire word lists to find possible cognates, I only consider two words to be cognates if they have the same English translation. Pivoting through English removes the need to compute a similarity score between every pair of words in every list, thus reducing the time complexity required to perform alignment. In addition, by introducing a strict semantic similarity constraint, I avoid clustering false cognates, which are orthographically similar by semantically distant.

On each group of words with the same English translation, I perform single-linkage clustering, an agglomerative clustering method where the distance between two clusters X and Y is $D(X, Y) = \min_{x \in X, y \in Y} d(x, y)$ for some distance metric d between two points (in this case, words) x and y. While clusters formed using this linkage method tend to be thin, I found that this method works well for cognates spread out across a language family compared to other linkage methods. I also investigate other linkage methods.

First, I construct lists of plausible cognates from existing dictionaries by running an initial clustering step on each group of words. The distance function for clustering is the Levenshtein distance (Levenshtein et al., 1966), a popular method for computing the edit distance between strings. The pseudocode for computing the Levenshtein distance is shown in Figure 5.2.

Specifically, I use the normalized Levenshtein distance

$$NLD(a, b) = \frac{\text{Levenshtein}(a, b)}{(max(\|a\|, \|b\|)} \tag{5.1}$$

with a clustering threshold of 0.5, i.e. half of the word must match. Treating these clus-

```
function LD(a, b)
    if a == ""
        return length(b)
    elseif b == ""
        return length(a)
    else
        return min(
            1 + LD(a, b[2:end]),  # insertion
            1 + LD(a[2:end], b),  # deletion
            (a[1] == b[1] ? 0 : 1) + LD(a[2:end], b[2:end])  # substitution
        )
    end
end
```

Figure 5.2: Pseudocode for computing the Levenshtein distance between two strings.

ters as multi-way aligned bitext, I run GIZA++ (Och and Ney, 2000) to extract character-to-character substitution probabilities, which are used in a second clustering step. The idea is that a second iteration of clustering should produce better results than a single iteration. This is similar to the two-pass approach employed by (Hauer and Kondrak, 2011).

For the second iteration of clustering, I define the distance function d between two words x and y as a linear combination of the following features, chosen specifically to model both the orthographic and semantic relatedness of cognates.

5.1.1 Weighted Edit Distance

Finally, I repeat the cognate clustering procedure, using a combination of features including both the learned inter-language and intra-family weighted Levenshtein distance. The idea is that a second iteration of clustering should produce better results than a single iteration. This is similar to the two-pass approach employed by Hauer and Kondrak

```
function WED(a, b, ins_cost, del_cost, sub_cost)
    if a == ""
        return length(b)
    elseif b == ""
        return length(a)
    else
        return min(
            ins_cost(b[1]) + WED(a, b[2:end]),
            del_cost(a[1]) + WED(a[2:end], b),
            sub_cost(a[1], b[1]) + WED(a[2:end], b[2:end]),
        )
    end
end
```

Figure 5.3: Pseudocode for computing the weighted Levenshtein distance, a generalization of the Levenshtein distance with custom insertion, deletion, and substitution costs.

(2011).

For the second iteration of clustering, I define the distance function d between two words x and y as a linear combination of the following features, chosen specifically to model both the orthographic and semantic relatedness of cognates.

Inter-Language Distance. A normalized weighted Levenshtein distance, where the insertion, deletion, and substitution costs are specific to the language pair (A, B) and the characters being compared (a, b).

$$\text{Ins}(a) = 1 - p_{A \to B}(\text{NULL} \to a) \tag{5.2}$$

$$\text{Del}(a) = 1 - p_{A \to B}(a \to \text{NULL}) \tag{5.3}$$

$$\text{Sub}(a, b) = 1 - p_{A \to B}(a \to b) \tag{5.4}$$

The character transition probabilities are obtained from alignment using GIZA++. The probabilities are subtracted from 1 to convert them to costs used in the edit distance calculation. I also experiment with adding an addition rule such that the distance between identical characters is zero to account for the noisy nature of alignment.

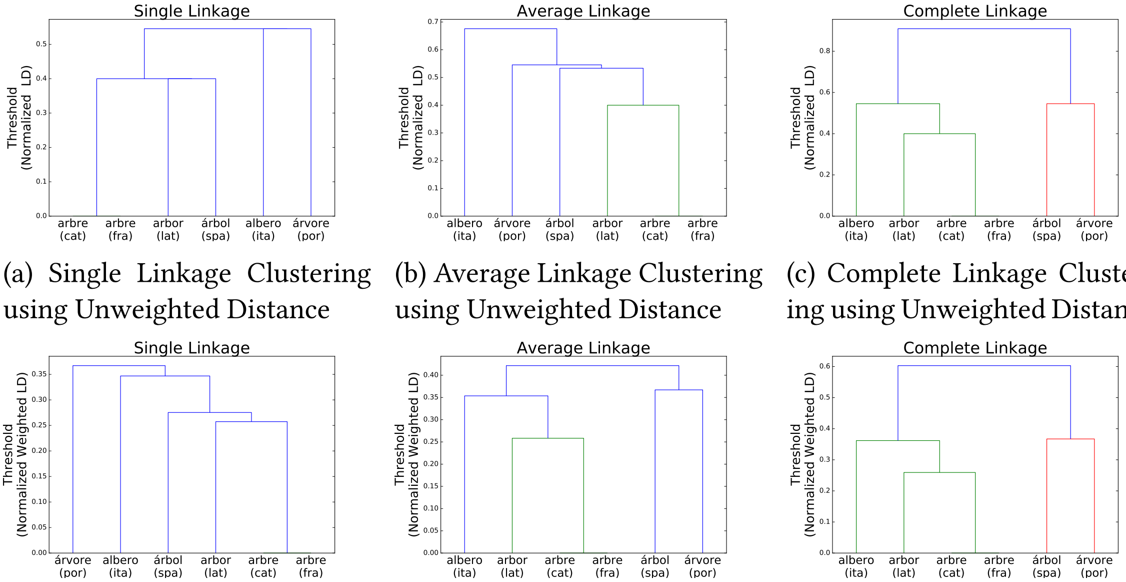
Intra-Family Distance. Same as the inter-language distance, except that the probabilities are obtained by character alignment on the concatenation of all bitexts of every language pair. This is a more universal, non-language-specific distance, and is expected to smooth or counter-balance the inter-language distance if there is not enough data for an accurate measure of inter-language distance. The intra-family distance is also used as a fallback distance in place of the Inter-Language Distance when comparing words of the same language. In practice, I observed that the intra-family distances are very close to the inter-language distance.

Same Backtranslation. A word's backtranslation is the most frequent English translation of that word in PanLex. If a word is in Wiktionary but not in PanLex, I assign the backtranslation to be that word's English translation. This feature is 0 if two words' most common backtranslation is the same, or 1 if they are different.

Same POS. Part of speech is obtained from the English edition of Wiktionary. Polysemous words may have multiple parts-of-speech. If a word is in Panlex but not in Wiktionary, the word will not have a POS.[1] This feature is 0 if two words share a common part of speech, and 1 otherwise.

Same MeaningID. A word from PanLex has a set of possible Meaning IDs that link it

[1] PanLex occasionally contains POS tags for words, but I choose not to use them because they are often incorrect (e.g. due to OCR errors), and words seem to be marked as nouns by default.



(a) Single Linkage Clustering using Unweighted Distance

(b) Average Linkage Clustering using Unweighted Distance

(c) Complete Linkage Clustering using Unweighted Distance

(d) Single Linkage Clustering using Weighted Distance

(e) Average Linkage Clustering using Weighted Distance

(f) Complete Linkage Clustering using Weighted Distance

Figure 5.4: Results of different linkage methods with unweighted and weighted distances

to semantically equivalent words in other languages. If a word exists in PanLex, I include all Meaning IDs that occur with this word. A word in Wiktionary but not in PanLex will not have a Meaning ID. This feature is 0 if two words share a common Meaning ID and 1 otherwise.

5.1.2 Linkage Methods

I motivate the choice of clustering linkage method by illustrating the results of the multiple-iteration clustering approach using hierarchical clustering with different linkage methods: single-linkage, complete-linkage, and average-linkage. These methods differ only in the metric used to merge clusters:

$$\text{Single(X, Y)} = \min_{x \in X, y \in Y} d(x, y) \tag{5.5}$$

$$\text{Complete(X, Y)} = \max_{x \in X, y \in Y} d(x, y) \tag{5.6}$$

$$\text{Average(X, Y)} = \frac{1}{|X||Y|} \sum_{x \in X, y \in Y} d(x, y) \tag{5.7}$$

for some distance function d.

In Figures 5.4a to 5.4c, using an unweighted normalized Levenshtein distance, *arbre* in Catalan and *arbre* in French are immediately grouped into the same cluster because they have a distance of zero. Ideally, these words should all be grouped into the same cluster, because they are true cognates. Single linkage clustering fulfills our needs the best, because the range of distances for merging clusters is the smallest.

When performing a second iteration of clustering using the weighted distances, the dendrograms in Figures 5.4d to 5.4f show similar results. Notably, the range of distances between clusters shrinks, which supports the hypothesis that multiple iterations of clustering are beneficial.

5.1.3 Evaluation

In previous work (Wu and Yarowsky, 2018b), I evaluated cognate clusters on the downstream task of cognate generation. I explore this task in Section 5.2. In this section, I perform an intrinsic evaluation of the cognate clusters using (Batsuren, Bella, and Giunchiglia,

Family	Distance	Clusters	ARI
Italic	unweighted	69,873	0.38
Italic	weighted	65,017	0.32
Oghuz	unweighted	4,279	0.61
Oghuz	weighted	4,067	0.63

Table 5.1: Intrinsic cognate clustering results compared to CogNet.

2019), a large database of cognates which was published shortly after the work on which this section is based (Wu and Yarowsky, 2018b). CogNet contains 3.1 million cognates for 338 languages. I experiment with two language families, Italic (consisting of cat, fra, frp, glg, ita, lat, lld, por, roh, ron, sci, spa, srd) and Oghuz (consisting of aze, gag, tuk, tur). To evaluate the clustering, I first remove all words that do not exist in CogNet, for a total of 164,848 Italic words and 3,321 Oghuz words. I compute the Adjusted Rand Index (ARI), comparing the clusters to the cognate sets in CogNet. Results are shown in Table 5.1.

I find that the second pass of clustering using the weighted edit distance is beneficial: it groups together cognates that existed in separate clusters in the second pass. This results in denser cognate clusters across the language family. It improves the cognate cluster quality as measured by ARI for Oghuz languages, but decreases quality for Italic language. However, considering that the number of gold cognate sets in CogNet is 35,821 and 2,773 for Italic and Oghuz, respectively, additional clustering may be necessary to further condense the cognate clusters. Nevertheless, I find that the multi-pass clustering method is able to successfuly identify cognates across languages when other resources, such as bitext, are not available.

5.2 Multilingual Cognate Generation

This section build upon some of my existing work (Wu and Yarowsky, 2018b; Wu, Vyas, and Yarowsky, 2018; Wu and Yarowsky, 2018a; Wu, Nicolai, and Yarowsky, 2020) in which I experimented with many variations of sequence-to-sequence models (both non-neural and neural) on several language families. One of my notable contributions (Wu and Yarowsky, 2018a) was that a single neural model trained on the combination of multiple languages was more effective at cognate transliteration than separate models trained separately on each language. Here, I extend this work to a larger scale.

Following existing work, I formulate the cognate generation task as a sequence translation task, where the input contains characters of the cognate word (with spaces replaced with underscores), along with source and target language tokens to direct the multilingual model to translate to and from the appropriate languages. An example is shown below, where Latin is the source language and Spanish is the target language:

```
 Input:  lat spa m a t e r
Output:  m a d r e
```

Using CogNet, I train and evaluate multiple multilingual neural cognate generation models, looking spefically at separate language families, as well as on the combination of all languages in the dataset. I have previously shown that multilingual cognate generation models outperform models trained on a single language, because the multilingual model can take advantage of information that is shared across languages, and also benefits from the larger training data. An open question, however, is whether these models benefit from

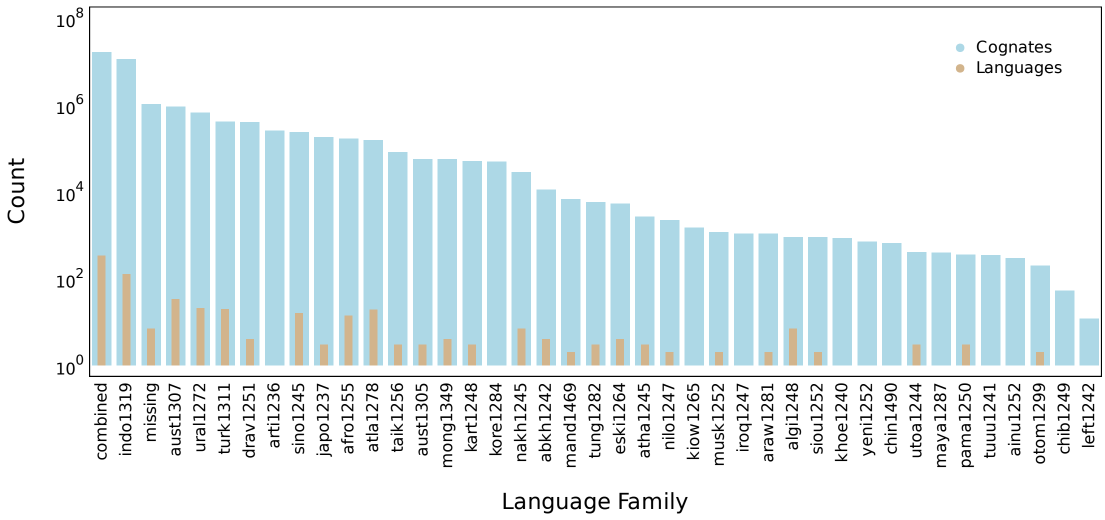


Figure 5.5: The distribution of number of cognates and number of languages within each language family in CogNet. Note the log scale on the y-axis (no bar indicates that the language family contains a single language). The *combined* label indicates all the data combined, and the *missing* label indicates languages that did not have a language family in Glottolog (Basque and several ISO 639-3 macrolanguage codes).

the combination of different language *families*. Within a family, related languages share cognates, but between families, languages may not share cognates, and may also differ in writing scripts.

I group the CogNet 2.0 cognates, which comprises 338 languages, into 44 language families according to the classification in Glottolog 4.4 (Nordhoff and Hammarström, 2011). The distribution of languages is shown in Figure 5.5. For training, I stratify split the data into a 80-10-10 train-dev-test split, where each split contains the same proportion of each language, and ensure that both directions of the cognate relation (i.e. A → B and B → A) exist in the same split.

The model is a two-layer LSTM encoder-decoder with 500 dimension embedding size and hidden size, trained with the ADAM optimizer with early stopping after 10 epochs, and label smoothing of 0.1. This model was implemented using the OpenNMT-py toolkit (Klein, Kim, Deng, Nguyen, et al., 2018). I train separate systems for each language family, as well as a single universal system using the concatenation of the training sets of each language family. I evaluate the performance on several metrics, including accuracy and average character edit distance, for both the models' top prediction and a 5-best list. A full table of results is shown in Table 5.2.

Experimens show very good performance on many language families, including low-resource families such as Oto-Manguean (otom, spoken in the Americas) and Pama-Nyugan (pama, spoken in Australia), which only have on the order of a hundred training examples. This is thanks to the amplified signal from related languages. I briefly comment on several of the lowest-scoring language families: Artificial (arti), Mayan (maya), and Indo-European (indo). The Artificial language family in CogNet contains only Esperanto (`epo`). While performance on generating Esperanto cognates has low accuracy, it has only a moderate average character edit distance, which indicates that the model is getting most of the word correct. Indeed, examining the model output shows that the model typically misses suffixes of the word. Esperanto is known for its highly regular and simplified morphology. A typical example is shown below (spaces are removed to facilitate visualization):

src	gold	predictions
ast epo angulosu	angula	angulo, anglo, anglino, anglio, angulos

The Mayan language family in CogNet consists of only Yucatec Maya (`yua`). Surprisingly, some entries in the test data do not look like cognates at the surface level. For

Family	n	Acc	AED	Acc 10	ED 10
abkh1242	1486	80.55	0.98	86.0	0.58
afro1255	15906	35.91	3.19	49.29	2.24
ainu1252	11	63.64	1.18	63.64	1.09
algi1248	74	71.62	1.43	72.97	1.16
araw1281	73	38.36	2.89	45.21	2.18
arti1236	26625	6.23	2.68	15.76	1.8
atha1245	193	64.25	1.97	77.72	0.96
atla1278	16053	42.58	2.11	54.07	1.39
aust1305	4507	35.03	2.66	45.66	1.88
aust1307	100782	27.08	2.86	37.9	2.08
chib1249	3	100.0	0.0	100.0	0.0
chin1490	37	56.76	2.16	62.16	1.7
drav1251	42391	10.21	5.05	17.87	3.8
eski1264	777	78.76	1.24	88.03	0.61
indo1319	1163944	5.41	3.93	10.72	3.31
iroq1247	44	52.27	1.82	72.73	0.86
japo1237	7681	41.11	1.52	57.35	0.98
kart1248	3983	65.5	1.48	72.88	1.03
khoe1240	70	10.0	2.21	58.57	0.97
kiow1265	161	52.17	2.57	60.87	1.6
kore1284	3444	43.76	1.88	54.15	1.44
left1242	4	25.0	3.0	25.0	1.5
mand1469	605	37.19	2.21	54.05	1.37
maya1287	46	0.0	4.63	4.35	3.04
missing	98524	22.44	2.94	34.07	2.17
mong1349	4935	33.39	4.12	41.05	3.24
musk1252	197	68.53	1.6	70.56	1.39
nakh1245	3031	67.44	1.46	81.56	0.75
nilo1247	149	63.76	1.21	66.44	0.93
otom1299	18	100.0	0.0	100.0	0.0
pama1250	29	62.07	1.62	62.07	1.21
sino1245	25633	35.8	2.49	47.96	1.82
siou1252	74	74.32	0.74	87.84	0.34
taik1256	7575	32.17	2.81	52.77	1.93
tung1282	656	65.55	1.82	73.48	1.14
turk1311	36282	41.78	1.81	56.54	1.16
tuuu1241	47	68.09	1.13	78.72	0.55
ural1272	57755	18.41	2.99	29.38	2.06
utoa1244	28	32.14	3.14	50.0	2.29
yeni1252	61	22.95	2.25	77.05	1.43
combined	1623896	7.23	3.57	13.93	2.86

Table 5.2: Results on multilingual cognate generation.

example:

src	gold
por yua comer	hanal
dsb yua jěsć	hanal
ltz yua iessen	hanal

This may be an error in CogNet, and since *hanal* was not seen during training, the model was not able to recover the correct cognate.

Indo-European is the largest language family in the dataset, and the model for Indo-European performs poorly both with respect to accuracy and character edit distance. Rather than learning to translate cognates, the model learns a very accurate transliteration function. This is likely due to the large amount of training data and large number of languages, which pushes the model to be a more universal transliterator rather than a (sub-)family specific cognate translator. Because of this, the model usually outputs the same word if the word is already in Latin script:

src	gold	model predictions
abk dsb ноиабр	nowember	noiabr, noiabra, nojabr, nojabra, noiabri
afr bre glucose	glukoz	glucose, glukose, gluzose, glukoze, glusose

I also evaluated the models grouped by each cognate word, where different source language's predictions on the target cognate are combined (as in Figure 5.1) using score-based voting, where each source language produces an n-best list of predictions on a target word, and each model gives their predictions a score of $n - rank + 1$ (i.e. for a 5-best list, the top-ranked hypothesis receives a score of 5, the 2nd-ranked hypothesis receives a score of 4, etc.). Results on this experimental scenario are shown in Table 5.3. We find in general that system combination improves over the results of single language systems.

Family	n	Acc	AED	Acc 10	ED 10
abkh1242	47	65.96	1.91	78.72	1.09
afro1255	1400	14.93	6.35	34.29	4.37
ainu1252	6	66.67	1.0	66.67	0.83
algi1248	14	35.71	2.93	35.71	2.5
araw1281	10	40.0	3.0	50.0	1.6
arti1236	2693	6.05	2.86	23.25	1.27
atha1245	22	31.82	4.14	50.0	1.95
atla1278	3444	35.19	1.8	58.94	0.86
aust1305	448	24.55	3.09	43.97	1.67
aust1307	9379	43.86	1.88	82.1	0.37
chib1249	1	100.0	0.0	100.0	0.0
chin1490	3	66.67	2.33	66.67	1.33
drav1251	6150	8.03	5.37	27.38	3.1
eski1264	32	28.12	4.47	56.25	2.31
indo1319	148095	6.03	3.84	28.64	1.99
iroq1247	3	33.33	2.33	100.0	0.0
japo1237	1947	36.83	1.39	59.48	0.75
kart1248	172	28.49	3.73	62.21	1.8
khoe1240	5	0.0	2.2	60.0	1.0
kiow1265	9	33.33	4.56	77.78	0.89
kore1284	463	14.04	4.47	22.25	3.88
left1242	2	50.0	2.0	50.0	1.0
mand1469	28	21.43	3.29	50.0	1.39
maya1287	4	0.0	4.0	25.0	2.5
missing	16077	19.46	2.73	45.84	1.35
mong1349	538	13.75	9.75	24.35	7.89
musk1252	11	45.45	2.82	45.45	2.09
nakh1245	150	37.33	2.84	58.0	1.51
nilo1247	15	33.33	3.27	40.0	2.27
otom1299	2	100.0	0.0	100.0	0.0
pama1250	12	16.67	3.58	16.67	2.67
sino1245	7703	53.2	1.21	69.18	0.72
siou1252	2	50.0	1.5	100.0	0.0
taik1256	1063	25.02	3.16	55.03	1.71
tung1282	33	57.58	2.12	78.79	0.64
turk1311	2348	25.38	2.5	60.95	0.98
tuuu1241	8	37.5	2.62	62.5	0.88
ural1272	5447	12.56	3.37	40.17	1.47
utoa1244	9	33.33	3.22	44.44	2.11
yeni1252	5	20.0	4.8	60.0	4.2

Table 5.3: Results on multilingual cognate generation with system combination, grouped by cognate word.

Finally, I evaluated the single massively multilingual model on each language family separately. Similar to the Indo-European results, I found that the combined model acted more as a transliterator and was unable to correctly predict many cognates. The best performance across language families was around 30% accuracy. Thus, I do not show the full table of metrics but conclude that there may be an upper limit on how many non-related languages to include during training.

5.3 Conclusion

Sound-shifting is a major class of word formation across the world's languages that encompasses, among others, cognates. To train sound shift models, one requires lists of aligned cognates, which are not readily available for all but the largest resource languages. I propose a multi-iteration clustering approach using a weighted edit distance for identifying cognate sets. This method enables the automatic creation of large-scale cognate tables for training multilingual cognate models. I experiment with training such models on 44 language familes, as well as a massively multilingual model trained on hundreds of languages, finding that including additional unrelated languages does not improve performance on cognate generation.

Chapter 6

Machine Learning for Computational Etymology

In an era of abundant linguistic data, I seek to address the dearth of computational approaches to modeling etymology. Using data extracted from Wiktionary, I present several approaches to model from where, how, and when a word enters a language. I employ RNN-based models and sequence-to-sequence models to accurately predict a word's formation mechanism, donor language, and donor word. I also experiment with various historical data-driven models for predicting word emergence. My methods are language-independent and are applicable for improving existing etymology determinations that may be incorrect, as well as providing etymology for words that may not have existing etymological entries, both in low- and high-resource languages.

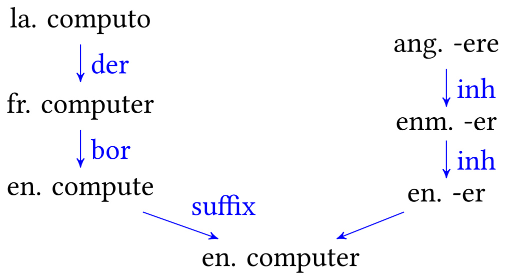


Figure 6.1: Wiktionary etymology graph of the English word *computer*. Etymological relationships are shown in blue.

6.1 Wiktionary Etymology

Wiktionary[1] is a large, free, online multilingual dictionary that is editable by anyone in the world. In addition to containing information found in traditional dictionaries (pronunciations, part of speech, definitions), it is rich source of other information that help one understand a word, including etymology, synonyms, antonyms, translations, derived terms, related terms, and even quotations. In this secion, I focus on etymology.

The etymological relationships between words[2] can be represented as a directed graph, where the nodes are words and the edges are etymological relationships. For example (Figure 6.1), according to Wiktionary, the etymology for the English word *computer* is *compute* + the suffix *-er*. The word *compute* is borrowed from the French *computer*, which is derived from the Latin *computo*. The *-er* suffix is inherited from the Middle English *-er*, which is inherited from the Old English (Anglo-Saxon) *-ere*.

Wiktionary has a set of guidelines[3] for annotators to document etymological relations.

[1] wiktionary.org

[2] Wiktionary contains separate entries for affixes like *-er*, so I informally call them “words” here.

[3] https://en.wiktionary.org/wiki/Wiktionary:Templates#Etymology

<table>
<tr><td>Displayed Text:</td><td>From Middle English cat, catte, from Old English catt (“male cat”), catte (“female cat”), from Proto-Germanic *kattuz.</td></tr>
<tr><td>Wiki Markup:</td><td>From {{inh|en|enm|cat}}, {{m|enm|catte}}, from {{inh|en|ang|catt||male cat}}, {{m|ang|catte||female cat}}, from {{inh|en|gem-pro|*kattuz}}.</td></tr>
</table>

Figure 6.2: Etymology of the English word *cat*.

Label	Count	Label	Count
affix	28366	derived	132404
back-form	24	inherited	159239
blend	144	mention	265220
borrowed	104817	noncognate	188
calque	964	prefix	18169
clipping	44	semantic loan	15
cognate	32095	short for	3
compound	42524	suffix	49505
confix	2185		

Table 6.1: Etymological relationships extracted from Wiktionary. Note that cognate and noncognate relationships are bidirectional relations, while the rest are unidirectional.

Yawipa uses a variety of heuristics to parse the unstructured Wikitext that makes up the the etymology section of a page (see Figure 6.2). Wikitext is a wiki markup language used by Wiktionary and Wikipedia. Table 6.1 summarizes the etymology information extracted.

Besides the challenges of unstructured text, the human element also poses challenges: annotators are sometimes inconsistent in following the Wiktionary guidelines. According to the guidelines, `inherited` is used for words that are from an earlier stage of the same language, while `borrowed` is used for words coming from other languages. The `derived` label is intended as a catch-all label for words that are not borrowed or inherited, whereas a stricter definition of (morphological) derivation would be a word that is formed from

Word	Mechanism	Parent	Correct
analyst	derived	(fr) analyste	borrowed
blind	derived	(ang) blind	inherited
agricultural	affix	agriculture + -al	suffix
peatbog	affix	peat + bog	compound
acetal	compound	acetic + alcohol	blend

Table 6.2: Examples of noisy Wiktionary etymology labels for some English words. ang is Old English

another existing word, often with an affix. The `affix` label is another catch-all for words that do not fit into the other affixal categories prefix, suffix, or confix, or they may have multiple prefixes and/or suffixes. Table 6.2 samples some inconsistencies with the etymology annotations found in Wiktionary. While it is not possible to exactly determine the number of inconsistencies, the large number of etymological relationships labeled as derived and affix indicates that there are many words for which a precise relationship is not known.

6.2 Etymology Prediction

To improve upon and expand the etymology annotations in Wiktionary, a natural solution is to develop a computational model to solve the following task: given a (language, word) pair, this work seeks to predict both the *relationship* of etymology and *which language* the word came from. Using the etymology data parsed with Yawipa, I run three experimental settings spanning different granularities of etymology prediction:

1. Input: Language Code + Word
   Output: Coarse Relationship

$$\texttt{en c o m p u t e r} \rightarrow \begin{cases} 0.13 & \text{affix} \\ 0.08 & \text{bor} \\ 0.07 & \text{cmpd} \\ 0.11 & \text{inh} \\ 0.12 & \text{prefix} \\ 0.56 & \text{suffix} \end{cases}$$

Figure 6.3: Setup of the fine-grained mechanism prediction task. For the language-specific setting, the leading language token (here, `en`) would not be present, and in the parent language prediction task, an additional token for the mechanism (e.g. `suffix`) would be appended.

2. Input: Language Code + Word
   Output: Fine Relationship
3. Input: Language Code + Word + Relationship
   Output: Parent Language

For the fine-grained mechanism prediction, I use the etymology labels affix, borrowing, compound, inherited, prefix, and suffix. Notably, I do not include the `derived` label due to the noise it adds to the dataset.[4] For predicting coarse-grained mechanism, I use two classes: borrowing/inheritance, and compositional, which encompasses compound, affix, prefix, and suffix. For language prediction, to make the problem computationally tractable, I predict the top five most frequent parent languages of a word, or "other" if the parent word's language is not in the top five.

I frame the task of etymology prediction as a multilabel classification task, where the input is a sequence containing the word's ISO 639-3 language code and the individual characters in the word, and the output is a probability that the word belongs to one of

[4] In initial experiments, I included words with the `der` label, but found that the models had trouble distinguishing derivations from borrowings. Further analysis showed that words labeled as derived are noisy, as previously discussed.

	Coarse		Fine		Language	
Lang	Base	Ours	Base	Ours	Base	Ours
af	**0.92**	0.91	0.79	0.79	0.72	**0.81**
en	0.52	**0.76**	0.34	**0.51**	0.42	**0.80**
it	0.51	**0.84**	0.35	**0.57**	0.48	**0.68**
ja	0.89	**0.92**	0.81	**0.85**	0.58	**0.70**
sw	0.70	**0.79**	0.48	**0.59**	0.32	**0.52**
zh	0.98	0.98	0.82	**0.86**	0.36	**0.54**
all	0.66	**0.83**	0.39	**0.53**	0.67	**0.79**

Table 6.3: Results on the etymology prediction tasks. The metric is accuracy.

the etymological relationship labels (note a word can have multiple labels, e.g. "apicide", which is borrowed from the Latin *apis* and contains the *-cide* suffix). The model is a LSTM with an embedding dimension of 128 and hidden dimension of 128. The output of the last hidden state is passed to a fully connected layer with a sigmoid activation function, with binary cross entropy as the loss and Adam as the optimizer with learning rate 0.001. The models were implemented using PyTorch. The data setup is shown in Figure 6.3.

I run these experiments on several languages around the world spanning various levels of resource-ness. In addition, I train a single multilingual system that can handle all the 3146 languages in the dataset by simply adding a language token in the input (Figure 6.3). I employ an 80-10-10 train-dev-test split, and test with the model with the lowest loss on the dev set.

6.2.1 Results and Analysis

Results are in Table 6.3. For almost all languages and settings, the neural method beats a strong majority baseline,[5] though it falls short when the class imbalance is high. Performance on Japanese (ja) beats the high-performing baseline because of a feature of the Japanese writing system: foreign words are written in katakana, while native words are written in hiragana or kanji. Thus foreign words are easily distinguished as borrowing due to differences in the script. For Afrikaans (af) and Chinese (zh), the performance is largely due to the tiny amount of training data (1.1K and 1.7K training examples, respectively), though it is remarkable that with such little data, a neural system can learn to predict etymology with such high accuracy. Equally remarkable is the finding that the spelling of a word alone is adequate to identify a word's etymology. This indicates that a language's prior on whether it prefers borrowing, inheritance, or compositional means for word formation is encoded in the spelling of the word. I will show later that a word's spelling, along with some etymology information, can predict a word's emergence year.

Due to familiarity with the language, I present analyses of some mistakes that the English models made. In the coarse mechanism prediction task (Table 6.4), the incorrect classification of borrowed/inherited words as compositional included borrowed words like *Prachuap Khiri Khan* that contained characters like hyphens or spaces that usually indicate compositionality, or words like *upright* that are technically inherited but could also be compositionally analyzed or were compositionally formed in an ancestor language. For

[5] The majority baseline is to pick the most common etymological class within a language.

Word	Pred	Gold	Confidence
tête-à-tête	comp	borinh	0.58
Prachuap Khiri Khan	comp	borinh	0.56
upright	comp	borinh	0.54
nurturant	borinh	comp	0.70
autovacuum	borinh	comp	0.56
cumulonimbus	borinh	comp	0.64

Table 6.4: Mistakes in the coarse mechanism prediction task.

words incorrectly classified as borrowing/inheritance, these are likely due to character sequences that are not common in the English language (e.g. the two components of *cumulonimbus* are borrowed from Latin).

For the English fine mechanism prediction task (confusion matrix in Table 6.5), the model incorrectly labels a large percentage of compounds as borrowings, and inherited words as borrowing or suffixes. Some mistakes are shown in Table 6.6. Many words incorrectly labeled as suffixed are due to the presence of a suffixal ending (-er or -ly); the suffixation of *drencher* and *gladfully* occurred in Middle English, so they are technically inherited, and words like *unmaidenly* and *macrobiotics* contain both a prefix and suffix. Words like *lesbro* or *Kleinberg* do not have a typical English spelling and are thus incorrectly labeled as borrowings. Other words like *appertain* and *injurious* are hard to distinguish as borrowed or inherited, due to the assimilation of Romance words due to Norman French.

Finally, for the language prediction task (confusion matrix in Table 6.7), the primary mistakes seem to be classifying French as other and other as Middle English. Some examples of misclassifying French borrowings include *sanitary* and *chagrin*. One explanation

	affix	bor	comp	inh	prefix	suffix
affix	27	23	13	0	23	58
bor	0	1108	19	61	24	82
comp	3	132	109	9	20	53
inh	1	137	25	286	19	138
prefix	5	43	6	24	223	39
suffix	4	99	22	21	34	587

Table 6.5: Confusion matrix of predictions for English, where rows are the true labels and columns are predictions. For visualization purposes, this is limited to truth and predictions that only contain a single label.

Word	Pred	Gold	Confidence
drencher	suffix	inh	0.55
gladfully	suffix	inh	0.72
unmaidenly	suffix	affix	0.55
aggrandize	suffix	bor	0.84
macrobiotics	prefix	affix	0.59
lesbro	bor	comp	0.75
Kleinberg	bor	comp	0.82
appertain	bor	inh	0.63
injurious	bor	inh	0.68

Table 6.6: Mistakes in the fine mechanism prediction task.

	en	enm	fr	la	grc	other
en	1822	0	1	11	8	34
enm	2	707	0	0	0	3
fr	34	0	110	2	13	109
grc	13	0	1	47	3	26
la	25	9	7	8	120	82
other	39	101	21	4	38	880

Table 6.7: Confusion matrix for predicting an English word's ancestor language.

for these mistakes is that the presence of so many Romance words has diluted the Germanic spelling pool and thus confuses the model. Many of the misclassifying "other" mistakes included words that were inherited from Old English, like *font* and *cress*. Similar analysis can be performed for other languages, and future work includes collapsing languages of a single line (like Old, Middle, and Modern English) into a single label.

6.2.1.1 Modeling Borrowings

In this section, I specifically examine borrowings, i.e. when a word enters a language from an unrelated language. Unlike inherited words, which arrive from a related language via sound shift mechanisms, borrowed words can be formed through a variety of mechanisms. I focus on six specific types of borrowings (whose Wiktionary label is in monospaced font below) across a spectrum of semantic and phonetic fidelity:

- `calque`: Also called a loan translation. Components of the original word are literally translated into the target language, e.g. the English *brainwash*, from the Chinese 洗脑 *xi* 'wash' + *nao* 'brain'.
- `partial calque`: A calque where not every component is translated, e.g. the English *apple strudel*, from the German *Apfelstrudel*.
- `semantic loan`: A sense extension is borrowed onto an existing word, e.g. the French *souris* 'mouse', which borrowed the computing sense from the English *mouse*.
- `psm`: Phono-semantic matching. Components of the original word are replaced with phonetically and semantically similar words, e.g. 声纳 *sheng* 'sound' + *na* 'receive',

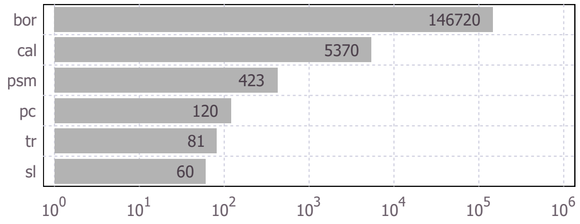


Figure 6.4: Distribution of borrowing relations.

from the English *sonar*.

- `transliteration`: A deterministic process of writing script conversion that seeks to preserve a word's orthography.
- `bor`: A generic borrowing category. The overwhelming majority of borrowings in Wiktionary are labeled as such. In this paper, I distinguish between `bor`, this relation as annotated in Wiktionary, and "borrowing", the word formation process encompassing these six relations.

The borrowing data extracted from Wiktionary consists of over 150K ground-truth annotated borrowing relationships, spanning a total of 837 languages. The top 10 languages are shown in Table 6.8. Note that only 101 languages have more than 100 entries, and 260 languages have more than 10 entries. In this work, I am also specifically interested in the long tail of low-resource languages. The distribution of borrowing relations is shown in Figure 6.4. Note the log scale, and the fact that that the majority class (`bor`) comprises 96% of the entire dataset, which motivates several experimental variants.

Lang	Count	%
eng	23,142	0.15
lat	18,713	0.12
fra	17,556	0.11
spa	7,123	0.05
ara	6,508	0.04
san	6,393	0.04
grc	6,122	0.04
deu	5,390	0.04
rus	5,109	0.03
ita	4,660	0.03

Table 6.8: Distribution of top 10 languages extracted from Wiktionary.

6.2.2 Tasks

I first establish terminology for borrowings: we say etymology is directed relation between a donor word and an incorporated word.[6] I experiment on two tasks in etymology prediction:

6.2.3 Task 1: Incorporation Prediction

Given a donor word and a target language, how would the word be incorporated into that language? And by what means? This task is motivated by a real-world example[7]: when deep learning was gaining popularity, researchers were considering how to best render the term into Japanese. Should it be a loanword and written in katakana (ディープラーニング *dīpurāningu*), or translated using a calque (深層学習 *shinsō gakushū*

[6] I eschew the established terms "loanword" and "borrowing" because loaning and borrowing imply an obligation to return the item being borrowed. In contrast, "borrowed" words are fully incorporated into the language.

[7] Thanks to Kevin Duh for this example.

'deep' + 'learning')? Besides terminology standardization, this task has applications in language revitalization and unknown word translation.

6.2.4 Task 2: Donor Prediction

In the opposite direction, given a word, from where and how did it come into the language? If we view Wiktionary as a directed graph, where the nodes are words and the edges are etymological relationships, there are missing edges. The task is to reconstruct these missing edges. As Wiktionary is a human-annotated resource, there is much variance in the quality and completeness of annotations, and good performance on this task can help fill in etymology even in high-resource languages like English.

6.2.5 Experiments

To tackle these two tasks, I employ character neural sequence-to-sequence models. For Task 1, predicting the incorporated word, the input is a sequence containing: the donor language, each character of the donor word, the etymological relation, and the target language. The output is the characters of the incorporated word.

```
In:   eng c a b b a g e bor abe
Out:  k a b i j
```

For Task 2, the input is a sequence containing the word's language and each character of the word, while the output is the donor language, donor word characters, and relation.

```
In:   abe k a b i j
Out:  eng c a b b a g e bor
```

For Task 1, I experiment with separate LSTM models trained for each borrowing relation (LSTM-sep), a single multi-task LSTM model trained on the combined data (LSTM), the same model trained with both the source and target data preprocessed by the unigram SentencePiece method (Kudo and Richardson, 2018) with a vocabulary size of 4000 (LSTM-spm), the same model with copy attention (See, P. J. Liu, and Manning, 2017) (LSTM-copy), a Transformer Vaswani et al., 2017 model (TF), and an ensembling method (Ensemble). This method is a score-based voting procedure that combines the output of the LSTM-sep, LSTM, and TF models. Each model gives 5 votes for their top prediction, 4 votes for their second place prediction, and so on (1 vote for fifth place). For each test instance, the votes are tallied up, and the prediction with the highest number of votes is the prediction of the ensemble. Ties are broken by picking the prediction with the highest model decoder score among all the models.

For Task 2, I experiment with a baseline LSTM model and the same model with copy attention.

All models were trained using the OpenNMT-py framework (Klein, Hernandez, et al., 2020). The LSTM models are two-layer encoder-decoders with 500-dimension hidden state, trained with the ADAM optimizer. The Transformer model has a 6-layer encoder and decoder with 8 heads, trained with ADAM with learning rate scheduling. For reproducibility, we provide the training scripts which include the full model details. Accounting for the extreme imbalance in our dataset, we performed a stratified split of the dataset into a 80-10-10 train-dev-test split, where each split contains the same proportion of languages

Model	BLEU	Acc	CED	5Acc	5CED
LSTM-sep	53.77	20.00	2.42	33.51	1.82
LSTM	55.83	21.43	2.31	34.98	1.71
LSTM-copy	55.90	19.92	2.32	34.46	1.69
LSTM-spm	45.62	10.68	2.85	20.31	2.13
Transformer	61.30	22.19	2.06	41.54	1.43
Ensemble	60.32	25.67	2.05	49.24	1.18

Table 6.9: Results for Task 1. Acc is accuracy (higher is better), CED is average character edit distance (lower is better). 5 indicates 5-best results.

and borrowing relations.

6.2.6 Results and Analysis

6.2.6.1 Task 1

I evaluate each model on a held-out 15,288 example test set. Table 6.9 presents character BLEU (computed with SacreBLEU Post (2018)) as well as accuracy and character edit distance from the gold (CED). I also report 5-best results for accuracy (was the correct answer in the top 5 results?) and CED (within the top 5 results, what is the minimum edit distance to the correct answer?)

At a cursory glance, the single models trained on all the data performs slightly better compared to the separate relation-specific models, following a trend of multi-task training performing better than models trained on a single task. The Transformer model performs the best, likely due to its innovative attention mechanism that has proven successful in other tasks. However, by examining the results for each borrowing relation, we see that the successes of the models are largely on the `bor` relations. All the models perform poorly in correctly predicting any non-`bor` relations, though we find that the calque-specific model performs slightly better than the jointly trained LSTM on calques. For example, the separate calque model correctly predicted the German *vollschlank* borrowed into Dutch as *volslank*, which the LSTM model could not do. And even when it generates incorrect answers, often the predictions look like "good attempts" at calqueing. For example, the French *Pays d'en Haut* gets translated as *Land of the Roud* (correct is *upcountry*), whereas the jointly trained models often do character substitutions instead.

Copy attention (LSTM-copy), which allows the model the option to copy characters from the source, was intended to help the model with similarly spelled borrowings, but overall it did not perform as well as a simple LSTM model. The subword model (LSTM-spm) also unexpectedly did not perform well. The goal of using subwords was to encourage the model to translate larger character sequences, the idea being that translational relations such as calques would consist of two subwords rather than several individual characters. Indeed, the LSTM-spm model treats most words as calques, often translating when it should instead perform character substitutions or sound shifts. Ensembling of three models' outputs is a simple but effective method resulting in a large increase in prediction performance. The score-based voting effectively combines the strengths of individual models, especially when all models have the same word in their n-best predictions.

Error Analysis. Due to the small quantities of available training data for partial calques, semantic loans, phonosemantic matches, and transliterations, the models cannot accurately learn to predict words incorporated by the aforementioned processes. This data shortage is exacerbated for the separately trained systems. Models largely treat these translational borrowings as generic `bor`s and perform character substitutions and sound shifts. This approach, exemplified by cognate transliteration systems, works for the majority of test examples, because `bor`s are essentially cognates with small edit distance. All phonosemantic matches are Chinese, so models will output Chinese characters, but due to the sparsity of the characters, the model cannot produce the correct answer. For the remainder of this analysis, I will focus on `bor` and `cal` as the main two borrowing relations. All models show similar patterns of prediction; the following examples are from the multi-task LSTM model.

In many cases, the incorporated word is similar to the donor, so the model can correctly predict the borrowing. For example, for the Latin *vanitas* borrowed into French, the model predicts *vanita*; the correct *vanité* is its second choice. The model can also handle different writing scripts. For example, it correctly predicts the Greek *πυρῖτις* borrowed into Latin as *pyritis*. Unfortunately, sound shifts do not work for the other borrowing relations, like calques, that require translation of morphemes. In many cases, the model does not seem to distinguish between non-`bor` relations and merely performs sound shifting. For example, the model predicts that the English *shopping center* calqued into Afrikaans is *schoppingsentre* (correct is *winkelsentrum*).

When encountering calques, the model sometimes recognizes that it should translate rather than transliterate. However, the lack of sufficient training data prevents the model from learning to accurately translate component morphemes. For example, the model predicts the English *download* calqued into German is *Dunnleut* (correct is *herunterladen*). Here, we see that the model picks up on the fact that German words tend to start with a capital letter, though in this case the word in question is a verb which does not need capitalization. The model also often cannot recover the correct word order when languages have different adjective-noun ordering. For example, the model incorrectly predicts that the French *mariage blanc* borrowed into English is *marriage mank* (correct is *white marriage*).

Broken down by language, the data contains numerous low-resource languages, many of which have just 1-10 words. Training a single model on such data for a single language would yield low performance, but the massively multilingual borrowing models can successfully handle many of these low-resource languages.

6.2.6.2 Task 2

For Task 2, I follow Wu and Yarowsky (2020a), who used an LSTM model to predict both the language and formation mechanism of a word. While they attempted to predict broader categories of inheritance vs borrowing, I focus on six specific borrowing relations. Because many borrowings have small edit distance, I also employed an LSTM model with copy attention. This model's performance was slightly worse than the baseline LSTM, a trend also observed in Task 1. This indicates that borrowings are fundamentally different from inherited and cognate words, where copy attention models have seen good performance. Results grouped by word, language, and relation are presented in Table 6.10.

The models for Task 2 are inherently multi-task: they must predict the donor language, donor word, as well as the relation. As such, prediction of donor language and relation can be evaluated as classification tasks. The models were able to generate valid languages and relations in 98% cases, showing that sequence-to-sequence models can also be successful in classification tasks.

I briefly analyze the errors of the LSTM model. Perhaps unsurprisingly, the model gets over 96% accuracy on predicting the relation by always guessing `bor`, the majority class. Yet it is able to beat a strong majority baseline (always predicting `bor`, the majority class). The model is also able to successfully predict the language of the borrowing in almost half of the test instances (guessing the majority donor language, English, would only achieve 14.8% accuracy). Thus a word's language and spelling provide sufficient information for identifying how and from where it entered the language. In terms of errors, some instances where the model predicts a donor language that is actually related to the correct language. For example, the Dutch *tabak* is borrowed from the Spanish *tabaco*, rather than the model's prediction of the French *tabac*, and many Dutch words originally from English were predicted to come from German, and vice versa. In addition, several words like English *specify* were predicted to come from French, but are actually from Old French. Future work can address a custom loss function that gives "partial credit" to such predictions rather than marking them as completely incorrect.

Model	Rel	Lang	Word	CED
Majority	96.0	14.8	–	–
LSTM	96.1	47.9	23.2	2.9
LSTM-Copy	96.1	47.7	20.8	3.0

Table 6.10: Results for Task 2: 1-best accuracy grouped by Relation, Language, and Word. CED is average character edit distance for Word prediction.

In terms of word prediction, the seemingly low accuracy of the model is not discouraging. Supported by the low character edit distance, there are many examples where the model's prediction is close enough to be recognized by a human. For example, the Chinese ⊠⊠⊠⊠ is borrowed from English *a cappella*, but the model predicts *acapara*, and the Jersey French *thiâtre* was predicted to be borrowed from Latin *thiatrum* (correct is *theātrum*). When providing new entries to an impoverished etymology dictionary, the prediction model can suggest possible etymology and even plausible unknown word forms, which can then be verified by a human lexicographer.

6.2.6.3 Conclusion

I model word borrowings from a donor to an incorporated word, and vice versa, using neural sequence models in a variety of experimental scenarios. I find that a single model trained to predict multiple types of borrowings performs better than separate models trained for each borrowing. A Transformer model performs better than an LSTM model, and a simple ensembling method results in superior performance, though the amount of training data is a limiting factor in the performance of these models. Predicting the donor language and word is a slightly easier task, where the LSTM model is able to beat a strong majority baseline.

6.3 Predicting Word Birth

One aspect of etymology that Wiktionary does not specifically contain is information about *when* a word entered the language. Based on a word's parent language, one can approximate the date of entry, e.g. a word borrowed into English from Middle French would have entered sometime around 1300–1600, the lifespan of Middle French. However, this is imprecise.

In the remainder of this chapter, I present work on modeling word emergence, an integral part of a word's etymology. I distinguish between, word birth, the year a word was first recorded as being used, and word *emergence*, the year in which the word starts gaining popularity in usage, and I argue that the latter is more informative than the former. I examine two datasets of historical word usage, the Google N-Grams corpus (Michel et al., 2011) and Merriam-Webster's Dictionary (Dictionary, 2006), and propose several methods for predicting the year of emergence in any language.

6.3.1 Historical Word Data

There are few existing sources of historical word usage, especially for languages other than English. This work utilizes data from two sources:

Google N-Grams (GNG). The Google N-Grams project (Michel et al., 2011) collects

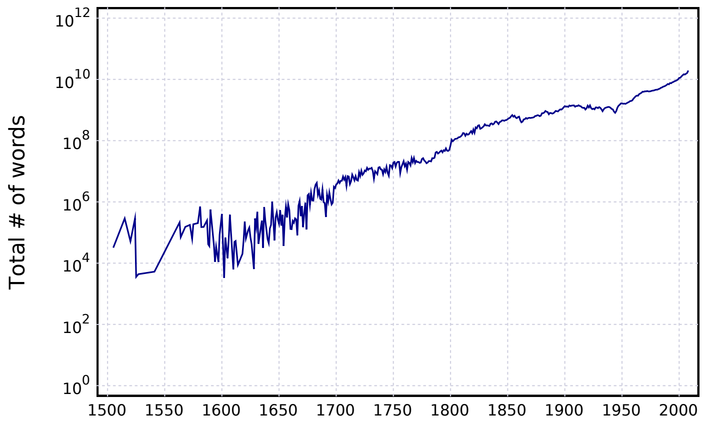


Figure 6.5: Total number of words in GNG per year. Note the log scale on the y-axis.

statistics of how many times a particular n-gram appears in how many books published in a given year. Data are available for 1- to 5-grams, and the languages covered are English, Chinese, French, German, Italian, Russian, and Spanish. The oldest books date from the 1500s, while the most recent are from 2008. GNG was constructed by using OCR to extract text. This process is not perfect, and I present methods that can potentially detect these errors. The total number of words in GNG per year is shown in Figure 6.5.

Merriam-Webster Dictionary (MW). This dictionary contains the year of first use for words in the English language. Before 1500, the data is more coarse-grained, and years are grouped by century; the oldest designation is *before 12th century*. The most recent words are from 2016. The data contained in MW is the first recorded year the word was used in print or writing.[8]

[8] Which is not necessarily when it was added to the dictionary. And the first attestation in print is also not necessarily the first strict usage of the word. Generally, words are introduced in speech before they are written down.

6.3.2 Models and Experiments

6.3.2.1 RNN-based

I first employ the same RNN-based approach as for modeling etymology, as a sanity-check to verify that modeling word birth is indeed possible. In this experiment, I use MW as the training data, restricting the words to those for which extracted etymology information exists (19,081 words). Different time periods in a language's history are characterized by different distributions of word formation (Figure 6.6). I am interested in assessing the contribution of etymology to the task of predicting word birth. I train a character-based neural model in a 70-15-15 train-dev-test split using the same setup and hyperparameters as in Section 6.2. An ablation study is conducted with four settings: only characters, characters + the parent language, characters + the word formation mechanism (bor, inh, etc.), and characters + mechanism + parent language. I experiment on these words and a reduced set whose birth year is $\geq$ 1500 (a total of 11,494 words), because in the MW dataset, years before 1500 are grouped by century. Results are presented in Table 6.11 (the metric is mean average error between the true year and the predicted year) and example predictions in Table 6.12.

Restricting the data to words born after 1500 results in a noticeable improvement, though even with the added noise of old words, the LSTM model can predict a word's birth year within two centuries. The models see improvements in performance when adding etymological information, which demonstrates that while a word's spelling en-

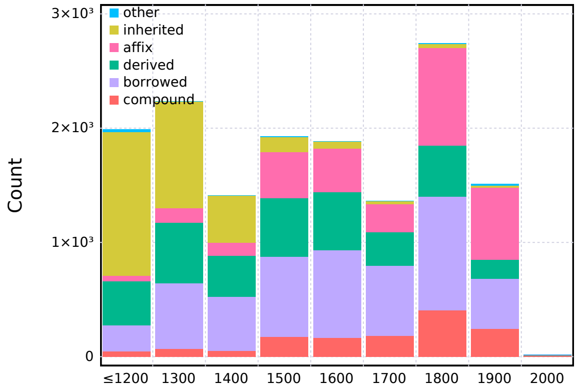


Figure 6.6: Sources of word formation for English words by century of word birth.

Setting	MAE (all)	MAE (year ≥ 1500)
Chars	253.0	118.9
Chars + Mechanism	180.9	112.8
Chars + Parent Language	157.9	103.2
Chars + Mech + Lang	157.3	101.9

Table 6.11: Ablation study of predicting word birth.

codes at least some information about a word's birth year, and knowing how and what language a word came from can help narrow the predicted time range of a word, allowing an average prediction within a century. Specific examples in Table 6.12 reveal that adding more etymology information tends to, but does not always improve predictions. These results indicate that word birth is modelable, but there are potentially better methods for doing so.

6.3.3 Examining Historical Data

The year of first use is somewhat problematic. I already noted that older words have a less precise birth year. OCR errors are also common; the classic example is the long

Word	True	C	CM	CL	CML
hippopotamus (bor, la)	1563	1682	1673	1662	1650
macrobiotic (affix, en)	1965	1804	1886	1819	1852
manucure (bor, fr)	1877	1723	1718	1739	1771
tae kwon do (bor, ko)	1967	1791	1937	1878	1955
eureka (der, grc)	1603	1750	1711	1783	1731

Table 6.12: A sample of predictions of birth year. C, CM, CL, and CML correspond to the settings in Table 6.11.

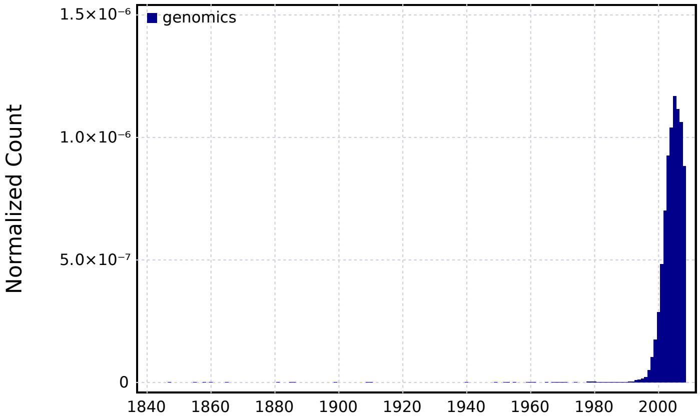


Figure 6.7: Normalized counts of the word “genomics” in GNG. Note the tiny bar at year 1847.

s (ʃ), which was used up until around 1800. OCR software have difficulty distinguishing between this letter and the letter ‘f’, so words like “funk” would appear to have a much earlier year of first use than in reality. And a word’s birth year is not necessarily informative: the word genomics (Figure 6.7) was first used in 1847, but did not gain popularity until the late 1900s.[9] Thus, I am interested in when a word gains traction, or emerges into the language, rather than the absolute first use. I devise several models of word emergence, following some preprocessing:

First, the GNG count data is smoothed by averaging the counts of the current year with those of the immediately preceding and following year. Then these counts are normalized by dividing by the total number of words in that year. This represents the percentage of the total number of words that a given word contributed in any given year.[10] I propose several data-driven formulas for extracting a word’s emergence year from GNG data:

- GNG First Attestation. Perhaps the simplest model: use the first year a word was attested in GNG. This may be problematic for younger (more recent) words, e.g. *genomics*.

- % of median threshold. Petersen et al. (2012) used a threshold of 0.05 × the median normalized count. They consider the first year a word’s count crosses this threshold as its emergence year.

[9] The term was coined in 1986 (Yadav, 2007).

[10] One observation with normalizing by the total number of words is that the usage of an old word may be diluted over time. For example, the normalized count of the Spanish word “agua” was 0.00298 in 1522 and 0.00023 in 2009. While in 1522, there was a smaller total number of words, the occurrences of “agua” made up a larger percentage of the total than in 2009, when the Spanish language had a much larger vocabulary size. Petersen et al. (2012) describes this phenomenon as “competing actors in a system of finite resources.”

- % of max threshold. A similar threshold heuristic: the first year in which the normalized count crosses 1% of a word's maximum normalized count is considered the emergence year.

- Curve Fitting. The above heuristics are simple but they do not utilize all the data. To take into account trends in the data, I employ locally estimated scatterplot smoothing (LOESS) to fit a curve to the data. LOESS is a non-parametric regression method that fits a low-degree polynomial (in this case, degree 2) to a sliding window of the data. This model was selected because, in many cases, humans can look at a graph of word usage and easily identify a word's emergence year just by noticing where there is a sudden change in the shape of the curve. This curve-fitting model predicts the emergence year of a word as the most recent year[11] where the LOESS curve crosses from negative to positive. If the curve never dips below the x-axis, then it designates the emergence year as the year at the curve's minimum value. I experimented with different settings for the span parameter, which controls the size of the sliding window.

- Derivative. The final model also exploits trends in the data: it takes the derivative of the LOESS regression curve and identifies the first year where it becomes positive. This indicates the beginning of an upward trend in the number of occurrences.

[11] There are cases where the curve may cross multiple times, especially if the word is older.

Year	# Words	First	Median	Max	C 0.3	C 0.4	C 0.5	C 0.6	C 0.7	Der	# Words	C+M+L
1500-1549	2360	**96.7**	96.7	96.8	299.5	311.4	319.3	326.4	337.6	145.3	39	199.2
1550-1599	4491	**89.9**	90.2	90.1	255.8	268.3	275.4	281.3	289.3	126.6	181	149.3
1600-1649	4230	**88.2**	88.6	88.6	214.3	225.7	232.5	236.6	240.8	111.2	288	129.4
1650-1699	3003	**81.9**	82.6	82.7	164.7	173.0	178.3	181.5	184.9	89.6	160	95.1
1700-1749	2108	80.8	81.9	81.8	117.8	127.3	132.6	135.5	138.6	**70.3**	104	65.2
1750-1799	3030	80.8	81.8	81.7	79.3	85.9	89.4	91.5	94.8	**53.1**	121	64.4
1800-1849	6053	77.8	78.9	78.7	47.4	52.8	55.3	57.2	58.6	**46.3**	195	56.2
1850-1899	8001	75.3	73.5	73.7	**34.5**	34.3	35.3	36.3	38.1	45.2	228	74.0
1900-1949	6801	83.6	75.5	75.6	30.2	**26.6**	26.7	27.0	28.0	51.6	229	95.4
1950-1999	3420	101.0	89.2	87.3	32.6	27.9	26.2	25.2	**23.4**	66.5	156	130.5
2000-2049	47	133.5	131.4	123.9	41.4	40.9	42.4	41.5	**38.7**	104.4	24	166.4

Table 6.13: Mean absolute error in years for different models. C 0.3 denotes the curve fitting model with span of 0.3.

6.3.4 Results and Analysis

As far as I am aware, there are no existing datasets for word emergence. Thus, I evaluate each of the above models in predicting a word's birth year as a proxy for emergence year. I utilize the intersection of MW words with unigrams from GNG, for a total of 57,015 words. Each model was evaluated on mean absolute error (in years) with respect to the gold birth years of MW.

I examine the performance of each model in 50-year increments (Table 6.13), revealing noticeable differences in model performance. On average, the simple heuristic models (First, Median, and Max) predict birth year within a century, though accuracy decreases for more recent words. On the contrary, the curve fitting models perform poorly on older words but greatly outperform the heuristic models on recent words. The derivative model, which uses the fitted curve, performs best around 1700-1800, but accuracy falls off for older and younger words. The RNN model exhibits a similar U-shaped performance curve.[12] For the non-neural models, First, Median, and Max are consistently within 100 years of

[12] Results for the best RNN-based model (chars + mechanism + language) were included in this table for comparison, but the results are not directly comparable because unlike the other models, the neural model uses a training and development set, so the test set is substantially smaller.

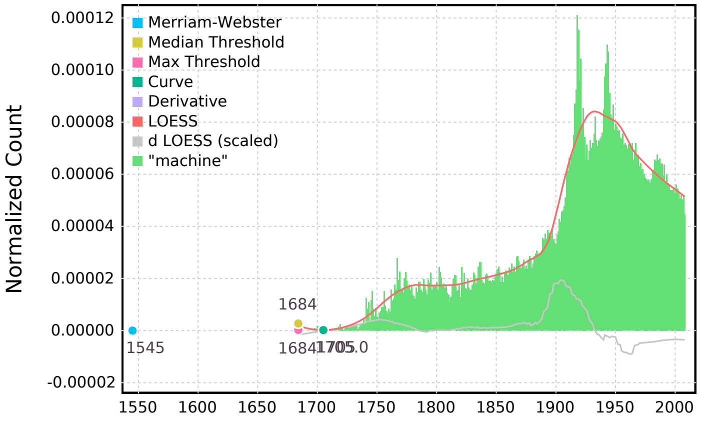


Figure 6.8: Plots of each model's birth year predictions on the word "machine".

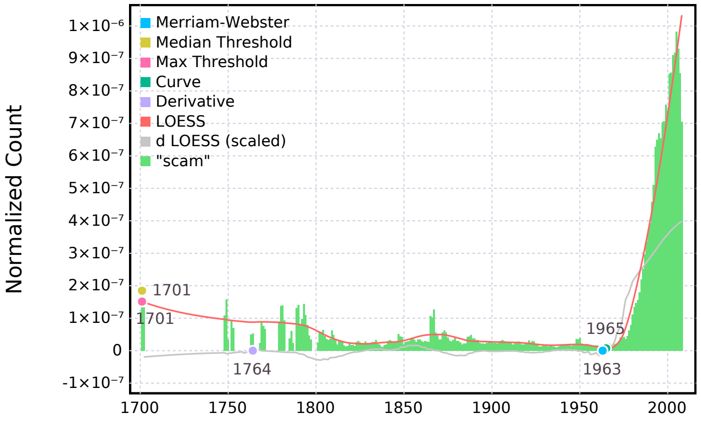


Figure 6.9: Plots of each model's birth year predictions on the word "scam".

the gold, the curve fitting and derivative models can greatly improve upon these simpler models. While Median and Max do not perform as well, they more accurately model the phenomenon of word emergence than First.

Figures 6.8 and 6.9 show each model's predictions on an older word *machine* and a younger word *scam*, respectively. MW lists the first use of *machine* as 1545, though it was not found in GNG until after 1700. For *scam*, MW lists the first use year as 1963, though the word seems to have been in use at a low frequency since 1700.[13] Because of this,

[13] The etymology of *scam* is uncertain. The earlier usages in Google N-grams are likely OCR errors of the

the simpler models give an incorrect birth year, while the curve fitting model correctly identifies the start of a period of exponential grow around 1960. Thus the curve-fitting model works well as a model for word emergence. Similar results were observed for GNG Spanish and French data, though there is no gold data to formally compare against.

6.4 Conclusion

I presented a Wiktionary parser with comprehensive support for parsing etymology and translations. I introduced the task of etymology prediction, where given a word, one should predict its parent word and language. I performed preliminary experiments on this task, showing the effectiveness of multilingual models. Regarding word emergence, an aspect not found in Wiktionary etymology, I experimented with numerous models in modeling word emergence using historical word data. All of the methods are language independent, and I see future application of these techniques in correcting misannotations and increasing coverage of etymological dictionaries for low-resource languages.

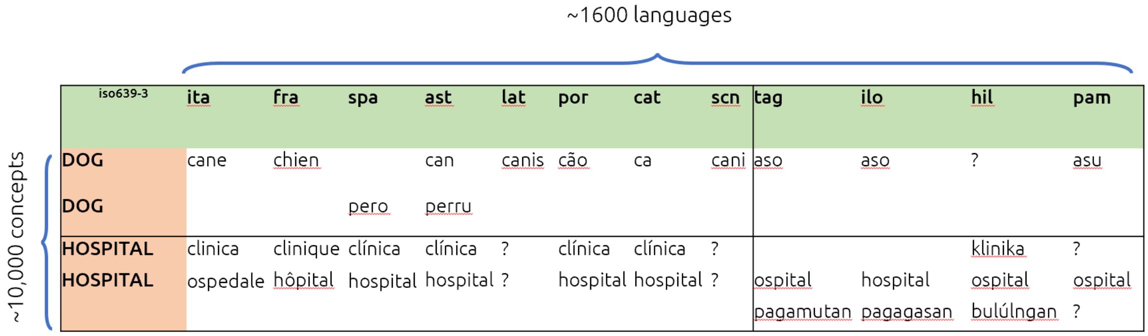
word *seam.*

Chapter 7

Model Combination for Generation of Unknown Words

This chapter combines the existing systems described in the previous chapters to realize the goal of constructing a comprehensive panlingual dictionary. Visually, this dictionary can be represented as a dense translation matrix, whose columns are the languages, and rows are realizations of the concepts in their respective languages (Figure 7.1).

An accurate, massive, dense translation matrix across the world's languages would be useful for many applications, first and foremost machine translation of low-resource languages. The combined efforts in this dissertation enable the construction of this matrix at such a scale that was not possible in the past.


~1600 languages
~10,000 concepts

iso639-3	ita	fra	spa	ast	lat	por	cat	scn	tag	ilo	hil	pam
DOG	cane	chien		can	canis	cão	ca	cani	aso	aso	?	asu
DOG			pero	perru								
HOSPITAL	clinica	clinique	clínica	clínica	?	clínica	clínica	?			klinika	?
HOSPITAL	ospedale	hôpital	hospital	hospital	?	hospital	hospital	?	ospital pagamutan	hospital pagagasan	ospital bulúlngan	ospital ?

Figure 7.1: A large translation matrix for core vocabulary. The bottom right quadrant represents low-resource scenarios with missing dictionary entries, for which my models are most applicable.

7.1 A Unified Test Set

Naturally, all the models proposed in this dissertation can be applied to generate large n-best lists to fill in every cell in this translation matrix. The issue is that we must also evaluate how good is this matrix; evaluating the models' hypothesized translations requires ground truth. Throughout this dissertation, I deal with extremely low-resource languages; there is no source of monolingual or bilingual data available besides a small bilingual dictionary. Thus for our purposes, I assume Wiktionary is the only data available. To evaluate a panlingual matrix, I hold out from the training dictionaries a portion of words from each test languages.

One major question is which words to hold out. In Chapter 3, I suggested that one should prioritize core vocabulary words when predicting novel word forms, because these words have important societal and cultural value. However, core vocabulary words are also less likely to be borrowed (thus useful for training sound-shift models), and are also

Language	Family	Speakers	Wiktionary Entries	Test Concepts
Galician	Italic	2.4M	55K	619
Bulgarian	Slavic	8M	27K	735
Irish	Celtic	170K	2856	504
Maltese	Semitic	500K	1967	233

Table 7.1: Summary of languages in test set.

more likely to be in the dictionary in the first place (thus valuable training data for low-resource languages). Depriving models of this training data may limit the model's performance. Therefore, I select a set of test concepts across the range of coreness (defined in Chapter 3), such that the test words span a range of frequency of usage, domains, and compositionality.

Concretely, I evaluate the hypothesized matrix on a set of four test languages: Bulgarian (`bul`), Irish (`gle`), Galician (`glg`), and Maltese (`mlt`). These languages range from medium resource to low resource and are members of different language families. I hold out every 20 concepts in the ranked core vocabulary list, i.e. the concepts at rank 20, 40, 60, ..., 20000, from the dictionaries of the aforementioned languages, for a test set of 1000 concepts. Note that not all 1000 test concepts are present in the dictionaries of the test languages; after all, these test languages are not high-resource. Thus, we can only evaluate on the concepts for which we have ground truth.[1]

Table 7.1 shows summary statistics about this test set. This test set contains words from a variety of domains and parts of speech,[2] making it a realistic, diverse, and general

[1] Studies in low-resource machine translation often evaluate on high-resource languages in a low-resource scenario: they artificially limit the amount of training data of the high resource language to simulate the effect of evaluating on low-resource languages. This is somewhat unrealistic.

[2] Note that the models are not specifically designed to handle all these parts of speech, e.g. prepositions

POS	Count
Noun	610
Adjective	122
Proper noun	111
Verb	94
Adverb	15
Phrase	14
Numeral	7
Preposition	7
Proverb	5
Interjection	3
Pronoun	2
Suffix	2
Determiner	2
Number	2
Prepositional phrase	2
Conjunction	1
Prefix	1
Total	1000

Table 7.2: Distribution of part of speech for concepts in the unified test set.

test set that encapsulates concepts that are likely to be encountered in real life. To illustrate the variety of concepts, a histogram of part of speech for the test concepts is shown in Table 7.2. The entire test set is shown in Table 7.3, in descending order of coreness.

Table 7.3: The 1000-concept test set.

1	blood	2	white	3	light	4	tea
5	frog	6	seed	7	Friday	8	die
9	deer	10	thousand	11	go	12	lung
13	whale	14	now	15	pine	16	give
17	fork	18	south	19	laugh	20	nineteen
21	thumb	22	dew	23	weapon	24	well
25	want	26	box	27	sickle	28	vulva
29	ink	30	bird	31	Israel	32	knowledge
33	stick	34	New Zealand	35	student	36	belt
37	fig	38	ice cream	39	enter	40	bride
41	saliva	42	pronoun	43	bubble	44	Russian Federation
45	adverb	46	Romania	47	Jordan	48	sport
49	ruler	50	mercury	51	easy	52	do you speak English
53	Christianity	54	mobile phone	55	fart	56	where
57	length	58	Portugal	59	spade	60	lazy
61	Libya	62	tall	63	example	64	work
65	sentence	66	gender	67	top	68	good
69	answer	70	shovel	71	invite	72	Palestine
73	necktie	74	Chile	75	frying pan	76	turnip
77	claw	78	moment	79	Brunei	80	hope
81	Confucius	82	coronavirus	83	prime minister	84	alms
85	happen	86	string	87	furrow	88	silicon
89	almost	90	organ	91	Prague	92	kilometre
93	Bahamas	94	drive	95	scrotum	96	base

or phrases.

97	mammal	98	strike	99	acceleration	100	hang
101	strange	102	Naples	103	geometry	104	sushi
105	architect	106	idol	107	starling	108	big
109	liberty	110	website	111	catch	112	governor
113	pistol	114	toilet paper	115	beast	116	gas station
117	resin	118	Chinese	119	clever	120	marsh
121	speed	122	Joan of Arc	123	contract	124	prepare
125	Armenian	126	arthropod	127	handle	128	nationalism
129	three	130	Kathmandu	131	deceive	132	instrument
133	photosynthesis	134	traitor	135	Sahara	136	drag
137	marmot	138	suddenly	139	Judas	140	etc.
141	nude	142	someone	143	Burkina Faso	144	asteroid
145	fur	146	slippery	147	Cold War	148	anniversary
149	dirt	150	mechanics	151	scratch	152	Danish
153	above	154	driver’s license	155	orbit	156	sow
157	Gabon	158	ballpoint pen	159	digestion	160	intention
161	resistance	162	werewolf	163	Revelation	164	clown
165	haematology	166	proc	167	voter	168	Latin
169	caesium	170	function	171	older brother	172	telephone
173	Kurdish	174	basalt	175	diameter	176	grateful
177	mother-of-pearl	178	regiment	179	thrush	180	USSR
181	carp	182	full moon	183	living room	184	policy
185	snooker	186	Samarkand	187	client	188	fishing
189	note	190	snot	191	Belgian	192	Vishnu
193	decade	194	grater	195	microbe	196	seashell
197	vegetable garden	198	Macau	199	berkelium	200	glory
201	lunar eclipse	202	remind	203	thulium	204	adultery
205	central bank	206	fax	207	mailman	208	public
209	tense	210	Father’s Day	211	Zechariah	212	circumstance
213	handsome	214	navy	215	saw	216	uprising
217	Mount Everest	218	cobbler	219	harem	220	parcel
221	spinning top	222	Buckingham Palace	223	ace	224	complete
225	geographic	226	nebula	227	porch	228	surprise
229	Afghan	230	among	231	consequence	232	hawthorn
233	pool	234	stair	235	-ism	236	Latvian
237	autonomy	238	enclosure	239	imperialism	240	necrosis
241	splinter	242	Cancer	243	Swede	244	capitulation
245	dynamite	246	goldsmith	247	liberate	248	pestle
249	stink	250	Chicago	251	Ukrainian	252	career
253	exclamation mark	254	insult	255	occur	256	schooner
257	threat	258	Habakkuk	259	annual	260	cumin
261	glad	262	lonely	263	quarrel	264	to see
265	Comoros	266	Saint Vincent and the Grenadines	267	ascend	268	cranberry
269	flamethrower	270	kiosk	271	olive tree	272	samurai
273	unknown	274	Old Testament	275	bold	276	cowardice
277	handcuffs	278	loan	279	panther	280	rug
281	thirty-five	282	Catherine	283	Titanic	284	ark
285	crescent	286	freeway	287	instead of	288	over
289	sherbet	290	traffic jam	291	Khmer	292	act
293	boring	294	criterion	295	freezer	296	influenza
297	noble	298	predator	299	single	300	tiny
301	Brexit	302	Toronto	303	blessed	304	cowshed
305	forget-me-not	306	humility	307	mow	308	puff pastry
309	sour cream	310	virginity	311	Pangaea	312	any
313	caracal	314	democrat	315	forty-eight	316	linen
317	o’clock	318	purchase	319	six	320	variable
321	Marx	322	ache	323	chef	324	domain
325	goalkeeper	326	itch	327	penalty	328	sceptre
329	that	330	yell	331	Saint George	332	arrangement
333	charge	334	diocese	335	forty-two	336	kibbutz
337	nutcracker	338	roast	339	third person	340	yellow
341	Prince of Wales	342	bankruptcy	343	chiaroscuro	344	delay
345	guillotine	346	melancholy	347	oud	348	remedy
349	slide	350	trachea	351	Calliope	352	Moravia
353	Uzbek	354	benzene	355	chlorophyll	356	delta
357	fetter	358	how do you say ... in English	359	lesser spotted woodpecker	360	oakwood
361	proletarian	362	serf	363	trace	364	Bluetooth
365	Quidditch	366	aloe	367	bet	368	confidence
369	empty	370	grab	371	iguana	372	long time no see
373	nearsightedness	374	repression	375	sixty-nine	376	vector
377	Christmas Eve	378	OK	379	albatross	380	blouse
381	chard	382	daybreak	383	fleece	384	hourglass
385	light	386	part	387	reply	388	spades
389	upper arm	390	Canadian	391	Margaret	392	absurd
393	bisexual	394	control	395	dumbbell	396	go away
397	interaction	398	mercenary	399	oystercatcher	400	privatization
401	second person	402	symphony	403	witch doctor	404	Crimean Tatar
405	Lviv	406	Xinjiang	407	bond	408	confectionery
409	ear lobe	410	fuck you	411	hockey puck	412	limousine
413	moderate	414	phrase book	415	sarcasm	416	supernatural
417	tyrant	418	Ajaccio	419	I’m in love with you	420	Ural Mountains
421	assemble	422	bubonic plague	423	copula	424	epicentre
425	froth	426	herd immunity	427	kefir	428	merciful
429	past	430	rapeseed	431	socialist	432	tie
433	urgent	434	Argonaut	435	Henry	436	People’s Democratic Republic of Algeria
437	age	438	bogatyr	439	confess	440	doorbell

441	feed	442	handbook	443	ischemia	444	mica
445	notion	446	plus	447	sailing ship	448	stay
449	to burn	450	working class	451	Herod	452	Samsung
453	appointment	454	blue screen of death	455	clutch	456	decimetre
457	empathy	458	furious	459	iPhone	460	lackey
461	mortality	462	persecution	463	record	464	secondhand
465	span	466	to carry	467	what	468	Aristotle
469	Melanesia	470	Turkish bath	471	anew	472	cabbage roll
473	cooking	474	disperse	475	follower	476	handcuff
477	income tax	478	living	479	moo	480	ogre
481	pick	482	redundant	483	shine	484	suckle
485	to err is human	486	virtuous	487	Ares	488	Grim Reaper
489	Nuremberg	490	adze	491	beating	492	carefully
493	coworker	494	distress	495	fearless	496	from time to time
497	hammer	498	inter-	499	low tide	500	mild
501	obtuse	502	prejudice	503	ruminate	504	slander
505	to sell	506	will o’ the wisp	507	Cambrian explosion	508	Leyden jar
509	Saudi	510	agnosticism	511	autumnal	512	calandra lark
513	comedian	514	destination	515	embroider	516	frugal
517	gym	518	lake	519	mourning	520	optimistic
521	procedure	522	restrict	523	shock	524	star
525	theocratic	526	unnecessary	527	yellowish	528	Christadelphian
529	Judea	530	Tibetan	531	amino acid	532	azure
533	brood	534	concise	535	croak	536	eighty-nine
537	fishing cat	538	gestation	539	homeopathy	540	intellect
541	large	542	meiosis	543	on behalf of	544	pot calling the kettle black
545	red currant	546	semiconductor	547	stilt sandpiper	548	time
549	username	550	without	551	Buddhist	552	Guelph
553	Michigan	554	Thumbelina	555	acute angle	556	auscultation
557	booger	558	chorus	559	credit	560	distinguish
561	et al.	562	gelding	563	hybrid	564	juror
565	minus	566	object	567	paranoia	568	printing
569	requirement	570	slip	571	supply	572	to wash
573	wax	574	Balkan	575	Lena	576	Scandinavian
577	abomination	578	assign	579	board game	580	chanterelle
581	continuity	582	diacritical mark	583	earache	584	export
585	galangal	586	headland	587	impatient	588	jeep
589	main	590	monolingual	591	omnipresent	592	pierce
593	relax	594	sexual harassment	595	squeegee	596	temptation
597	town	598	vagabond	599	zander	600	Byzantine
601	I’m cold	602	Pandora	603	Yenisei	604	alliteration
605	backward	606	cache	607	compliment	608	curved
609	discord	610	essential	611	fortnight	612	great-granddaughter
613	impudent	614	it’s too expensive	615	long pepper	616	median
617	multimillionaire	618	outstanding	619	plane	620	pullet
621	related	622	seedling	623	smorgasbord	624	suspend
625	to sing	626	vestibule	627	-ist	628	Chita
629	Harry	630	Ottoman	631	St. Elmo’s fire	632	administrative
633	ar	634	birdie	635	calmness	636	choke
637	configuration	638	custody	639	dry ice	640	every cloud has a silver lining
641	fleeting	642	gibbon	643	hand	644	hyponym
645	land	646	mechanical pencil	647	national park	648	particle accelerator
649	produce	650	reproach	651	savanna	652	soursop
653	survey	654	to pour	655	umlaut	656	vigilance
657	yellowhammer	658	Caesar salad	659	Holy Grail	660	Maltese
661	Stalinist	662	accessory	663	asymmetrical	664	behaviorism
665	bottle	666	chemical reaction	667	contain	668	decapitation
669	doormat	670	exciting	671	footnote	672	great-great-grandfather
673	henceforth	674	inflammable	675	landowner	676	macaroni
677	military service	678	obelisk	679	patron	680	pogrom
681	pyrite	682	rest in peace	683	she-goat	684	special
685	supplement	686	tidal wave	687	travel agency	688	vulnerability
689	worsen	690	Bashkir	691	Independence Day	692	Northern Marianas
693	Shakespeare	694	abaca	695	are you allergic to any medications	696	baksheesh
697	borax	698	cardigan	699	cocoa powder	700	covet
701	desktop	702	ebb	703	evacuation	704	foam
705	giant panda	706	herbivorous	707	implementation	708	khanjar
709	loquacious	710	marshmallow	711	mugwort	712	opposite
713	pitch-black	714	psychological	715	reflexive pronoun	716	scraper
717	slag	718	stem cell	719	thanks for your help	720	transgender
721	ventricle	722	where are you	723	American English	724	Cassiopeia
725	Gordian knot	726	Navalny	727	Scandinavian	728	accomplish
729	analog	730	ask for	731	binding	732	bridge
733	cherry blossom	734	coppersmith	735	deen	736	drug addiction
737	esoterism	738	firebrand	739	geometric	740	hangnail
741	ibuprofen	742	intelligent design	743	kosher	744	lobe
745	money changer	746	one another	747	personnel	748	prosody
749	restlessness	750	sexton	751	so-so	752	stud
753	three thousand	754	underwater	755	weeping willow	756	Anatoli
757	Europa	758	Laurasia	759	Oriental Republic of Uruguay	760	Stanislaus
761	accumulator	762	amanita	763	asylum seeker	764	bird of paradise
765	bureaucratic	766	catechism	767	collage	768	corncockle
769	decomposition	770	disarmament	771	epilogue	772	feign
773	foreign currency	774	glottal stop	775	heathen	776	if I were you
777	insatiable	778	knave	779	mash	780	minimal pair
781	nautical mile	782	ominous	783	pardon me	784	plot

785	putsch	786	revive	787	scrutinize	788	shears
789	sodium hydroxide	790	strikebreaker	791	tempo	792	to show
793	unanimously	794	vocal cords	795	-ous	796	Confucianism
797	Gulf Stream	798	Lebanese	799	Odysseus	800	Thrace
801	accord	802	anonymity	803	audit	804	biryani
805	burbot	806	cf.	807	common shrew	808	cram
809	derogatory	810	dovecote	811	equilateral	812	fathom
813	free kick	814	greatest common divisor	815	hitman	816	informatics
817	joie de vivre	818	lion’s share	819	merger	820	negative
821	on	822	patronymic	823	playlist	824	pull
825	reliability	826	screw	827	skua	828	stagger
829	symbolism	830	to ask	831	uhlan	832	vortex
833	yeti	834	Basmachi	835	Democritus	836	Hiroshima
837	La Paz	838	Old French	839	Spanglish	840	acquittal
841	arable	842	baht	843	biographer	844	brunette
845	ceramic	846	color blind	847	coordinate	848	daring
849	digestive system	850	dubious	851	enteritis	852	far
853	foretell	854	gold mine	855	haste makes waste	856	hooray
857	inconceivable	858	jack-o’-lantern	859	libretto	860	masculine
861	moisten	862	nematode	863	optical illusion	864	penance
865	please turn left	866	proboscis	867	readiness	868	residence permit
869	scavenger	870	sinusitis	871	spout	872	supersonic
873	thanatology	874	to learn	875	udarnik	876	vibraphone
877	wolf spider	878	Bauhaus	879	Dominican	880	House of Lords
881	Luxembourger	882	People’s Liberation Army	883	Tibetan	884	accentuate
885	altruistic	886	arid	887	bandage	888	bier
889	brigadier	890	caries	891	chubby	892	compass point
893	courtesan	894	deaf-mute	895	discretion	896	dramatic
897	electromagnet	898	ester	899	fire	900	full
901	gradient	902	happily	903	hospice	904	impotence
905	invalid	906	landfill	907	liquidity	908	mendacious
909	name	910	obstetrics	911	parliamentary	912	phonological
913	postal	914	ptomaine	915	redeem	916	rock
917	sedative	918	smoked	919	spotlight	920	suburban
921	temporarily	922	to breathe	923	topple	924	underline
925	wand	926	willingly	927	zabaglione	928	Bhutanese
929	Draco	930	Hesiod	931	Kama Sutra	932	Neapolitan
933	Stockholm syndrome	934	Xanthi	935	admissible	936	angstrom
937	assailant	938	barrister	939	blacklist	940	brusque
941	cash desk	942	clientele	943	consequently	944	cross out
945	deem	946	dissolution	947	eligible	948	exclamation
949	fleur-de-lis	950	gamble	951	go nuts	952	grown-up
953	hippodrome	954	impulsive	955	intifada	956	layout
957	lymphoma	958	minuet	959	nasalization	960	ocelot
961	paper money	962	photocopy	963	pood	964	prone
965	radiology	966	renovate	967	sandhi	968	shawarma
969	slip of the tongue	970	stateless	971	superintendent	972	the more the merrier
973	to rub	974	troubadour	975	vigorous	976	whaler
977	yashmak	978	Angolan	979	Channel Islands	980	Gerona
981	I want to go to the toilet	982	Lakshadweep	983	Pandora’s box	984	Shenzhen
985	Toki Pona	986	ableism	987	all cats are grey in the dark	988	antepenultimate
989	atomic clock	990	binomial	991	bosom friend	992	bullseye
993	cartographer	994	child prodigy	995	cog	996	conman
997	crevice	998	deport	999	documentary	1000	ebony

7.2 Coverage in the Bible

I have previously mentioned the Bible as the most translated document in the world. The JHU Bible Corpus (McCarthy, Wicks, et al., 2020) contains word alignments with English on thousands of translations of the Bible. In this section, I analyze the coverage of these bibles and their respective alignments as a source of translation for my test set.

The English version of the Bible[3] contains 382 concepts in the test set. The concepts are listed below, in descending order of coreness:

> blood, white, light, seed, die, thousand, go, now, give, south, laugh, nineteen, thumb, dew, well, want, sickle, ink, bird, Israel, knowledge, stick, belt, fig, enter, bride, saliva, Jordan, sport, ruler, easy, where, length, lazy, Libya, example, work, gender, top, good, answer, shovel, invite, moment, hope, alms, happen, furrow, almost, organ, drive, base, strike, hang, strange, idol, liberty, catch, governor, beast, marsh, prepare, handle, three, deceive, instrument, traitor, drag, suddenly, Judas, someone, slippery, dirt, above, sow, Latin, fishing, note, glory, remind, adultery, public, Zechariah, saw, complete, porch, surprise, among, pool, pestle, stink, insult, Habakkuk, glad, quarrel, unknown, bold, loan, ark, over, act, noble, blessed, humility, virginity, any, linen, purchase, six, itch, sceptre, that, charge, roast, yellow, delay, remedy, slide, confidence, empty, fleece, light, part, control, bond, merciful, past, tie, urgent, age, confess, feed, stay, Herod, appointment, furious, persecution, record, span, what, disperse, living, shine, beating, carefully, distress, hammer, prejudice, slander, lake, mourning, star, Judea, brood, large, time, without, object, slip, supply, wax, abomination, pierce, temptation, town, backward, plane, choke, hand, land, produce, reproach, contain, special, covet, accomplish, binding, feign, plot, revive, on, pull, stagger, far, readiness, bandage, bier, discretion, fire, full, name, redeem, rock, willingly, ebony

The JHU Bible Corpus contains Bible translations in Bulgarian and Maltese, but not Irish nor Galician. For Bulgarian-English, 195 test concepts exist in the Bible alignments, and 61 of these alignments (31%) resulted in a gold translation that existed in Wiktionary. Correctly aligned words in the test set are presented in Table 7.4. For Maltese-English word alignments, 126 test concepts exist, and 14 of these concepts (11%) were aligned to a gold translation. Correctly aligned words in the test set are presented in Table 7.5.

Here I make several observations about using the Bible alignments for translation. First, my test set is a general test set. While the Bible covers only roughly a third of these words, it remains an excellent starting point for further dictionary development on lan-

[3] The King James Version, with archaism like *thou*, *-est* and *-eth* forms replaced with their modern equivalents.

Concept	Gold Idx	Top 10 Most Probable Alignments
blood	0	**кръв**, кръвта, кръвно, жертвената, Кръв, кървавочервена, кръвопролития, проляната, кръвопролитие, Кръвта
white	2	бели, бяло, **бял**, бяла, белтъка, обелиха, избелили, побеляха, побелее, избелят
light	1	виделина, **светлина**, светлината, виделината, просветление, Виделото, утрешната, светене, осветлено, видело
seed	4	Цяло, семеносно, посеял, Разграби, **семе**, семеносна, семето, Потомството, потомството, Семе
die	11	друго-яче, умри, измре, измрат, Умират, измрем, умреш, умрат, умрем, умрете
thousand	1	хиляди, **хиляда**, милион, Хиляда, хилядата, милиарди, строй
go	16	напредне, Насърчи, иди, изкачите, тръгни, възлизайте, отидеш, бежешком, пътуваме, Отиваш
now	0	**сега**, Царуваш, побледнее, състезават, заемем, отсега, Досега, досега, Отсега, Засега
give	4	приклонете, раздайте, отдадат, отдадете, **дам**, песнословят, въздайте, дайте, отдай, давай
south	3	южни, юга, южният, **юг**, югът, южната, освирепее, южно, южна, разсвирепее
dew	1	росата, **роса**, Наросявани, росен
well	1	кладенецът, **кладенец**, кладенче, Кладенец, благоденстваше, Кладенецът, оздравееш, кладенеца, благоденствуваш, Здрав
sickle	0	**сърп**, сърпа
ink	0	**мастило**
bird	1	птиче, **птица**, птицата, птичи, птичка, пернато
knowledge	0	**знание**, познанието, познание, знанието, просветена, познаване, Знание, познаването, корабници, знания
belt	0	**колан**, пояс, колана, пояса
bride	0	**невяста**, невястата, невестата, невеста
Jordan	3	Иордане, йорданската, Иордана, **Йордан**, Иордан, Йордане, Йорданските, Иорданската, Иорданска, Иордановото
ruler	1	властител, **властник**, вождът, владетеля, Алоисовият, началник, водача, владетел, управител, главатаря
where	3	пребиваването, где, къде, **където**, гдето, накъде, садил, приливът, приемната, осветена
lazy	0	**ленив**, лениви, ленивия, ленивецо, мързеливи
example	0	**пример**, примера, Подложени, наблюдавайте
work	7	престъпване, изработена, Делото, изхитруват, навезеш, дърворезбата, извезани, **направа**, работата, изработката
top	2	върхът, върха, **връх**
good	2	добри, добър, **добро**, добрите, доброто, добрата, благ, добра, добрия, добрини
answer	2	откликне, отговорът, **отговоря**, отзова, отговорите, отговаряш, отговориш, отговарям, сърдито, отговаряте
shovel	0	**лопата**
moment	4	минута, мигновена, минутна, Погинаха, **миг**
hope	9	закоравявай, надежда, надеждата, надеят, Надеждата, обнадеждени, 147, надей, уповай, **надявам**
alms	0	**милостиня**, милостини
almost	0	**почти**, свършване, превалил
beast	2	звяра, звярът, **звяр**, скот, Звярът, зверовия, животно, животното
three	0	**три**, трима, трите, тримата, триста, двама-трима, тризъбна, Три, тридневен, тригодишен
instrument	1	инструмент, **оръдие**
traitor	0	**предател**
suddenly	3	Неочаквано, лихоимство, ненадейно, **внезапно**, наближавах, внезапна, изведнъж, неочаквано, Внезапно
slippery	2	плъзгави, хлъзгави, **плъзгав**, хлъзгав, Опетнен, опетнен
above	6	вишния, по-висока, всевишния, горе, изработената, височайши, **отгоре**, по-горе, отличаваше, горното
sow	4	посяват, засейте, засея, посейте, **сея**, насея, сейте, сеете, посея, засяваш
glory	1	славенето, **слава**, славата, славо, прослава, вдигнатите, Славата, похвалиш, украшението, пестеливо
remind	4	припомни, припомня, напомни, напомням, **напомня**
surprise	0	**изненада**
among	6	смесите, предизвиквах, вникнете, смесвате, между, одумник, **сред**, най-силен, корейците, корят
bold	1	осмелява, **дързък**, смелост
over	4	привеждай, превеждай, домоуправителят, наводнят, **над**, настоятели, широкия, обиколка, премини, преминахте
blessed	8	Благослових, благословиха, благословил, благословени, благословените, най-благословена, благослових, благослови, **благословен**
any	24	бодлива, чертаете, никакъв, тъжба, кое-да-било, принесло, някое, Повярвал, жалост, някому
six	0	**шест**, шестима, шестте, шестстотин, шестстотинте, Шестимата, третите, шестстотната, шестимата, Шест
delay	5	отлагаш, забавяш, забавиш, бавих, бавене, **отлагане**, забави, бави
confidence	3	упование, дръзновението, увереността, **доверие**, смелостта, увереност, дръзновение, пристъпим
fleece	1	руното, **руно**
light	14	виделина, светлина, светлината, виделината, просветление, Виделото, утрешната, светене, осветлено, видело
merciful	2	милосърден, състрадателен, **милостив**, милостивите, милосърдни
persecution	0	**гонение**, гонението, напаст
span	0	**педя**
what	0	**какво**, каква, какъв, жадуващи, последиците, благоугодното, страхуващи, какви, що, мъдрувате
carefully	0	**внимателно**, изследвах
distress	5	утесня, досаждай, притесня, утеснението, притеснение, **бедствие**, наскърбя, утеснение, неволя
prejudice	1	предразсъдъци, **предразсъдък**
produce	0	**добив**, произведения

Table 7.4: Instances where the Bulgarian-English Bible word alignments recovered the correct Bulgarian word. Hypotheses are sorted by alignment probability. Bolded hypotheses indicate a correct prediction.

Concept	Gold Idx	Top 10 Most Probable Alignments
blood	1	d-demm, **demm**, mad-demm, tad-demm, demmi, bid-demm, mid-demm, id-demm, b'demmu, demmu
white	0	**abjad**, bajda, l-abjad, tçammex, bojod, b'dija
light	2	id-dawl, tad-dawl, **dawl**, çad-dawl, fid-dawl, d-dawl, f'dawl, bid-dawl, mid-dawl, mhijiex
thousand	0	**elf**, elef, l-elf, miljun
now	0	**issa**, bħalissa, çalissa, bis-serqa, mil-lum, ksibna, Bħalissa, iħammrulkom, tifilħux
ink	1	l-pinna, **linka**
Israel	0	**Iżrael**, f'Iżrael, jżommux
where	0	**fejn**, jitmermer, fejnhom, ssemma, mnejn
example	0	**eżempju**, mera
liberty	0	**ħelsien**, tal-ħelsien
beast	9	l-Bhima, mal-Bhima, lill-Bhima, il-Bhima, Il-Bhima, tal-Bhima, bil-Bhima, bħall-Bhima, daçmieni, **bhima**
three	6	tliet, tlitt, Tlieta, it-tlieta, Sewwasew, fid-disa', **tlieta**, jaqblu, t-tlieta, t-tliet
among	1	qawwietu, **fost**, f'nofskom, nofskom, çamiltx, Fosthom, Appostli, qalb, firdiet, it-tilwim
merciful	1	jħennu, **ħanin**

Table 7.5: Instances where the Maltese-English Bible word alignments recovered the correct Maltese word. Hypotheses are sorted by alignment probability. Bolded hypotheses indicate a correct prediction.

guages for which a Bible translation exists. Indeed, the existence of *Israel*, *Jordan*, *Judas*, and other proper names of religious significance in the core vocabulary list indicates that many dictionaries already use the Bible as a source of translations. From another angle, the Bible is a domain-specific text that is not general enough for daily conversation, as evidenced by the Bible's lack of modern technology and science terms, or geopolitical entities relevant in the modern world. This motivates the methods developed in this dissertation.

Second, the process of word alignment is noisy and may not achieve optimal word translation results. Running a morphological analyzer such as that of Nicolai, Lewis, et al. (2020) to obtain lemmas may help reduce the space of inflected forms to enable better translation from alignments.

7.3 Direct Neural Models

To validate the efficacy of the translation models proposed in the previous chapters of this dissertation, I first apply standard well-known machine translation models on

Input	Output
gle u n a n i m i t y	a o n t o i l í o c h t
fin t u r n i n g	s o r v a a m i n e n
vol b l o n d	h i b l o n a n
rus r a d i u m	р а д и й
ita s o m e t i m e s	o g n i _ t a n t o

Table 7.6: Data for the character-based direct neural model.

Input	Output
gle un@@ anim@@ ity	a@@ onto@@ il@@ íocht
fin turning	sor@@ va@@ aminen
vol blond	hi@@ bl@@ on@@ an
rus radi@@ um	ра@@ ди@@ й
ita sometimes	o@@ gn@@ i tan@@ to

Table 7.7: Data for the BPE-processed direct neural model.

the task, which I call the direct neural approach. These models are neural sequence-to-sequence machine translation models trained to predict the form of unknown words, given only a sequence containing the target language code, and the English concept. I use the same model and setup as in the cognate experiments but train with two data variants: (1) character-based (with spaces replaced with underscores), and (2) processed with byte-pair encoding (BPE) (Sennrich, Haddow, and Birch, 2016). The BPE was trained for 16K merge operations on the concatenation of the source and target side of the training data. An example of the data for each variant is shown in Tables 7.6 and 7.7.

Lang	Acc1	Acc10	Acc100	Ed1	Ed10	Ed100
bul	.098	.217	.274	3.52	2.52	1.86
gle	.016	.074	.147	3.68	2.48	1.81
glg	.160	.288	.366	2.19	1.32	0.91
mlt	.022	.049	.069	1.35	0.94	0.72

Table 7.8: Accuracy and edit distance evaluations for the direct neural approach using character neural models.

Lang	Acc1	Acc10	Acc100	Ed1	Ed10	Ed100
bul	.055	.163	.262	2.86	1.86	1.28
gle	.010	.034	.083	2.65	1.87	1.43
glg	.159	.281	.367	1.46	0.80	0.49
mlt	.018	.033	.043	1.08	0.79	0.64

Table 7.9: Accuracy and edit distance evaluations for the direct neural approach using BPE neural models.

7.3.1 Results

Accuracy and edit distance metrics for 1-best, 10-best, and 100-best lists are shown in Tables 7.8 and 7.9. Overall, the character-based direct neural model performs slightly better than the BPE-based model in terms of accuracy, but the BPE model has slightly lower (better) average edit distance. Why is this the case?

In the character-based model, the direct neural approach essentially models transliteration from English. This is beneficial for higher resource languages that may have borrowed from English or a related Germanic language. On the other hand, the BPE model seems to learn translations rather than transliterations.

For example, when predicting the Maltese word for STRANGE (gold is *għarib*):

Model	Top model hypotheses
Character	stran, strang, stranġ, għal, għar, stranż, strana, stren
BPE	ħar, għar, għarb, ħħar, għarda, għarra, għanja, għerb

the BPE model learns an underlying representation of STRANGE from a combination of exposure to other languages (Arabic: ḡarīb, Turkish: garip) as well as associations from within the same language (għarib is a translation of STRANGER, FOREIGNER, WEIRD, and ODD), thereby performing a similar function to the lexical relation model we proposed in Chapter 4. In other cases, the BPE model tries to predict words that look plausibly in the target language, but do not have any correspondence in meaning, for example, when translating SALIVA into Irish (gold is *seile*):

Model	Top model hypotheses
Character	sailí, salaí, saile, sala, sáile, saoil, sáil, sal
BPE	caol, lán, lus, glac, glas, saol, slis, slán

We see that while the first few hypotheses are nowhere close to the gold, the next few do have some semblance (with the *s* and *l*), but it is tenuous. This shows that while the direct neural approach is a decent first attempt at this task, more specialized models are needed to tackle the challenges posed by specific words.

7.4 Cognate and Sound Shift Models

A natural extension of the direct neural model is the cognate/sound-shift models presented in Chapter 5. I apply the multilingual methodology proposed in that chapter on the test languages across several values of clustering threshold. I train the same neural

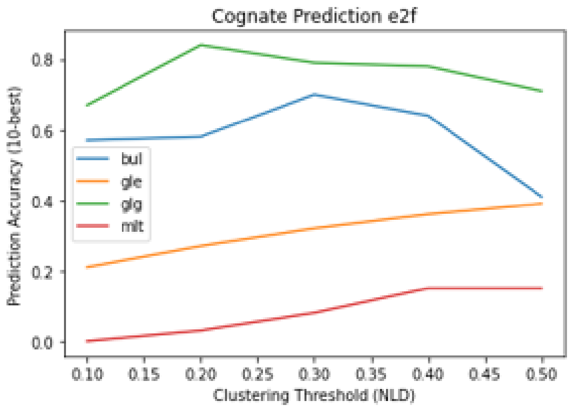


Figure 7.2: Clustering threshold for cognate experiments.

Language	Test	Acc1	Acc10	Acc100
Bulgarian	735	.27	.58	.78
Irish	602	.14	.27	.32
Galician	619	.53	.81	.92
Maltese	258	.07	.11	.16

Table 7.10: Cognate prediction results on test set.

encoder-decoder sequence-to-sequence model in Chapter 5 on this data, which was split into a 90-10 train-dev split, to predict a target language's cognate of a related language. Recall that the input is a sequence in the following format: `<src> <tgt> <c h a r a c t e r s>` and the output is the characters of the word in the target language. I evaluate the cognate model on our test set. Recall that in this model, any related language can be used to arrive at a gold translation. Hypotheses from all related languages are combined into a single n-best list, sorting by the decoder's score. A summary of accuracy results are shown in Table 7.10, along with 10-best accuracy across clustering thresholds in Figure 7.2.

The cognate models are the best performing models out of the three in this dissertation, and for good reason: there are very few language isolates, and thus there exist cognates in related languages, which the models can use to predict the correct word in the target language. Galician exemplifies this. While Galician is a low-resource language, as a member of the large Italic family, Galician can make use of its high-resource relatives. For example, for predicting the Galician word *sangue* 'blood', many related languages supply cognates:

Src Lang	Src Word	Model Predictions (Galician)
cos	sangui	sanga, **sangue**, sanguio, sangui, sango
ita	sangue	**sangue**, sanga, sanxa, sango, sang
lat	sanguis	sanguis, sanga, **sangue**, sangues, sanxa
pms	sangh	sang, sanga, sanghe, sange, san
por	sangue	**sangue,** sanga, sango, sang, sanxa
pov	sangui	sang, sanga, sangui, **sangue**, sango
ron	sanguină	sanguina, sanga, sanguino, **sangue**, sanxina
scn	sangu	sangu, sang, **sangue**, sanga, sango
vec	sangue	**sangue**, sanga, sango, sang, sanxa

This pattern is common for all of the cognate model's successes, even for lower resource languages and for concepts further down the core vocabulary list. Many concepts have multiple translations, which we consider correct if any source language will lead to a correct prediction. For example, for the concept 'redeem', Irish has three gold translations: *saor*, *slánaigh*, and *íoc*.

Src lang	Src Word	Model Predictions
gla	saor	**saor**, saor-, saorf, saír, saord
gla	ìoc	**íoc**, íoch, ioc, Ác, íoc-

In terms of errors, we noticed several categories of cases where the model could not predict a cognate. First, some words are clearly cognate but were not able to be generated, for example, the Irish word *tae* 'tea':

Src Lang	Src Word	Model Predictions (Irish)
bre	te	te, té, teo, tew, teu
cor	te	te, té, tí, teo, tes
cor	té	te, té, teu, teo, teD
cym	dysgled	dyscled, descled, dysclead, díscled, dascled
cym	paned	páinéad, pánadh, painéad, pánad, panéid
cym	te	te, té, tew, tes, é
cym	trwyth	troith, troyth, trosh, trwith, troíth
gla	tì	tí, tó, ó, té, Tí
glv	tey	te, teo, téa, té, tey

In these cases, though several source cognates exist, the model may have never seen transduction *e* $\rightarrow$ *ae* or *é* $\rightarrow$ *ae* to be able to generate the correct word *tae*. This phenomenon is more common for short words.

A second class of errors are words that are simply not cognate, and thus the cognate model is not amenable to these types of words. For example the Bulgarian *обществен имунитет* (obštestven imunitet) 'herd immunity' was not able to be generated from its related languages, because the first word обществен (obštestven) 'social, public, community' is not cognate with the other words in Slavic languages.

Src Lang	Src Word
ces	kolektivní imunita
hbs	imunost krda
mkd	колективен имунитет (kolektiven imunitet)
rus	популяционный иммунитет (populjacionnyj immunitet)
rus	коллекти́вный иммунитéт (kollektívnyj immunitét)
ukr	колективний імунітет (kolektyvnyj imunitet)

These types of errors were not handled by the cognate/sound-shift models and motivate the application of composition word formation models.

Language	Test Size	Acc1	Acc10	Acc100	AccN	Ed1	Ed10	Ed100
bul	740	.00	.00	.03	.24	6.60	5.12	3.59
gle	505	.00	.02	.08	.40	6.48	5.01	3.52
glg	619	.00	.03	.10	.37	6.14	4.50	3.00
mlt	235	.00	.01	.03	.26	6.02	4.62	3.47

Table 7.11: Compound prediction results on test set.

7.5 Compositional Models

I train compositional word formation models for generating foreign words as described in Chapter 4, holding out the test words. We use the best performing component joining method, which was the neural sequence-to-sequence model. Results are shown in Table 7.11. In-depth analysis on this test has already been presented in Chapter 4. To summarize, many of the test words are simply not compositional and thus not amenable to the compositional generation model. Overall, the compound recipes learned by the model are high quality, so the generation process is able to generate the correct word in the n-best list but often not in first rank, because the majority universal recipe of a concept does not always apply to a specific language.

7.6 Lexical Relation Model

Finally, I employ the lexical relation model described in Section 4.2 to produce translations of unknown concepts. Recall that this model does not generate unseen words, but rather uses a dictionary and WordNet to suggest existing words that may be valid trans-

Language	Test	Acc1	Acc10	AccN
Bulgarian	735	.12	.23	.38
Irish	602	.09	.21	.24
Galician	619	.10	.22	.31
Maltese	258	.12	.24	.24

Table 7.12: Compound prediction results on test set.

lations for a test concept. Evaluation of this model on the test set is shown in Table 7.12.

In depth analysis of this model on the test set has already been presented in Section 4.2. To summarize, this lexical relations model has practical utility, in that it does not require intensive training (compared to the cognate and compound models), and it reflects the actions that humans take when talking about unknown concepts(circumlocution). This model is especially useful for extremely low resource languages, such as Maltese, where there may not be enough cognate signal from related languages to train adequate cognate models.

7.7 Model Combination

Numerous studies have shown the efficacy of model combination in machine learning. I also perform model combination of the three above models for the task of unknown word generation. Hypotheses from each model are weighted as follows: let c be the compositionality score (Section 4.1.5.1) of a given concept. Then the weights are $w = [1 - c * 0.8, c * 0.8, 0.2]$ for the cognate, compositional, and lexical relation models, respectively. Then, model hypotheses are combined using rank-based voting, where each

Model	Acc1	Acc10	Acc100	AccN
Cognate	0.27	0.58	0.72	0.78
Compound	0.00	0.00	0.03	0.25
Lexical	0.12	0.30	0.38	0.38
Combined	0.24	0.60	0.73	0.85

Model	Acc1	Acc10	Acc100	AccN
Cognate	0.32	0.69	0.85	0.92
Compound	0.00	0.00	0.04	0.29
Lexical	0.14	0.35	0.44	0.45
Combined	0.28	0.71	0.86	1.00

Table 7.13: Model combination results on Bulgarian. The left table contains results on the 735 test concepts that exist in Wiktionary. The right table contains results on 626 test concepts where at least one model was able to generate the gold translation.

hypothesis gets a score of $(n - i) * w_m$, where n is the length of the n-best list, i is the rank of the hypothesis in the n-best list, and w_m is the weight given to model m.

In a real-world scenario, these models will have precomputed hypotheses, such that when a new text is first encountered, the user can look up new words the hypotheses lists. For each model (Tables 7.13 to 7.16), I report 1-best, 10-best, 100-best, and n-best accuracy, with the notion that any occurrence of a gold translation in the n-best list is considered a success. Why so? Due to the nature of this task, it is not terribly important that the models produce the gold unknown word as the 1-best or even 10-best translation. In a field linguistics scenario, a 100-best list is of a reasonable size for a native informant to quickly scan through and identify a valid translation. As more monolingual text is obtained in the target language, language models can be then built and used to filter these n-best lists.

For all the test languages, model combination gives a substantial improvement in accuracy, especially at the n-best accuracy metric. This result indicates that combining the three models allows one model to successfully compensate when other models cannot predict the answer. Naturally, each of the test concepts will not be amenable to all three

Model	Acc1	Acc10	Acc100	AccN
Cognate	0.14	0.27	0.31	0.32
Compound	0.00	0.02	0.07	0.34
Lexical	0.09	0.21	0.24	0.24
Combined	0.13	0.30	0.33	0.51

Model	Acc1	Acc10	Acc100	AccN
Cognate	0.27	0.53	0.61	0.62
Compound	0.01	0.04	0.14	0.66
Lexical	0.18	0.42	0.48	0.48
Combined	0.25	0.59	0.66	1.00

Table 7.14: Model combination results on Irish. The left table contains results on the 602 test concepts that exist in Wiktionary. The right table contains results on 306 test concepts where at least one model was able to generate the gold translation.

Model	Acc1	Acc10	Acc100	AccN
Cognate	0.53	0.81	0.90	0.92
Compound	0.00	0.03	0.10	0.37
Lexical	0.10	0.22	0.31	0.31
Combined	0.23	0.66	0.84	0.94

Model	Acc1	Acc10	Acc100	AccN
Cognate	0.56	0.86	0.96	0.98
Compound	0.01	0.03	0.10	0.40
Lexical	0.10	0.24	0.33	0.33
Combined	0.24	0.70	0.90	1.00

Table 7.15: Model combination results on Galician. The left table contains results on the 619 test concepts that exist in Wiktionary. The right table contains results on 581 test concepts where at least one model was able to generate the gold translation.

Model	Acc1	Acc10	Acc100	AccN
Cognate	0.07	0.11	0.15	0.16
Compound	0.00	0.01	0.03	0.24
Lexical	0.12	0.24	0.24	0.24
Combined	0.09	0.16	0.19	0.39

Model	Acc1	Acc10	Acc100	AccN
Cognate	0.19	0.28	0.39	0.40
Compound	0.00	0.02	0.07	0.61
Lexical	0.31	0.60	0.61	0.61
Combined	0.23	0.42	0.50	1.00

Table 7.16: Model combination results on Maltese. The left table contains results on the 258 test concepts that exist in Wiktionary. The right table contains results on 101 test concepts where at least one model was able to generate the gold translation.

of the cognate, compound, and lexical relation models, which have different but complementary strengths.

I also present system combination results on hypotheses for which at least one model produced an answer (Tables 7.13 to 7.16 right side). Overall, over a quarter of 1-best hypotheses were correct, and impressively, over 70% of 10-best hypotheses were correct. This shows that the models are able to perform well on amenable test concepts.

7.7.1 Analysis

In this section, I analyze the three models, looking at the specific strengths of each model. First, I examine the cognate model. As previously seen, the cognate model was the best performing of the three proposed translation generation models. Table 7.17 presents results on Galician where the cognate model was the only successful model to generate a hypothesis. There are quite a few proper nouns, which are more likely to be phonetically translated between languages. In addition, the cognate model is also performant on compositional words that are also phonetically translated rather than calqued. Examples of such successes include *New Zealand, central bank, flamethrower*, and *Old Testament.*

Looking specifically at successes from the compound model, they are fewer and often occur further down the n-best list. For Bulgarian, results where only the compound model could generate the correct translation are shown in Table 7.18. Most of these concepts are also compositional in English.

Finally, I present some successes from the lexical relation model on Maltese in Ta-

Concept	Gold	Idx	Top Model Hypotheses
tea	té	0	**té**, te, infusión, sopar, merenda, gostar, ditar, infuso, ceai, lonche
frog	gavacho,ra	7	rana, crapo, alamar, brogo, bivio, talón, xaronca, **ra**, granota, anura
Friday	venres,sexta feira	46	vender, venir, vener, venerde, venar, devender, verne, vendre, venerder, témpora
lung	pulmón,boche,livián,bofe	0	**pulmón**, polmón, palmón, resistencia, pumón, claro, lom, bofe, pulmo, lev
pine	madeira de piñeiro,piñeiro	18	pin, pino, pen, piño, pinu, pi, dor, firme, ansia, muga
thumb	polgar,matapiollos,escachapiollos	19	dedón, pulgar, policar, poso, púlgaro, deda, polegar, pouca, pólice, poce
dew	resío,orballo,rosada,relento	0	**rosada**, ros, rocio, rou, rizo, sereno, rucio, roua, relente, ruxiada
weapon	arma	0	**arma**, telo, arme, erma, armen, harma, aceiro, armas, telum, acero
ink	tinta,borra	0	**tinta**, escoria, tenta, negro, cerneal, encra, magma, lava, encre, intcha
Israel	Israel	0	**Israel**, Jsrael, israel, Israil, Esrael, Israiel, Trael, Yisrael, Ysrael, Israal
New Zealand	Nova Zelandia	1	neozelandés, **Nova Zelandia**, Nova Zelanda, Nueva Zelandia, Nova Selandia, neozelandesa, Nueva Zelanda
student	estudante,trancho	3	estudente, elevo, discente, **estudante**, escolar, estudiante, académico, educando, discípulo, alumno
ice cream	xeado,cornete	23	xelado, sorbete, neve, crema, glato, carapulla, mantecado, xelato, cremo, helado
bride	esposa,noiva,alarosa	0	**esposa**, esponsa, nuvia, novia, condición, noiva, bruto, nevasta, niveasta, nuta
adverb	adverbio	0	**adverbio**, adverbo, aberbio, averbio, alverbio, alberbio, adviebe, aberbo, advérbio, adverba
Romania	Romanía	0	**Romanía**, Rumanía, Romania, Romenia, Rumania, Romenía, Armania, Romaño, Rumenía, Remanía
Jordan	Xordania	0	**Xordania**, Xordán, xordán, xordano, Jordania, Iordania, Jordán, Xordaña, xordania, Iordán
easy	fácile,fácil,doado	0	**fácil**, simple, cómodo, levo, padre, mole, suave, lev, suelto, fácel
length	lonxitude	1	durada, **lonxitude**, largo, le, lonxitud, vasca, largura, duración, longor, lúnxime
Libya	Libia	0	**Libia**, Líbia, Libie, libia, Libía, Livia, Libio, Libye, Libea, Libya
example	exemplo	0	**exemplo**, exemplar, exemple, modelo, esemplo, esamplar, talco, espécime, calaña, exhibición
gender	sexo,xénero	0	**xénero**, sexo, xenro, sex, sexa, sexe, xen, xenero, sexus, sexu
shovel	pá,paa	12	pala, pica, negro, vanga, pela, espada, rutro, paleta, pique, pa
Chile	Chile	0	**Chile**, Chili, Cile, Xile, Chil, Chila, Cili, chile, chili, Chilo
turnip	nabo,cachola	3	nap, raba, rapa, **nabo**, napo, raf, rava, rave, rab, naveta
Brunei	Brunei	0	**Brunei**, brunei, Brunéi, brunéi, Bruneio, bruneio, Bruney, Brúnei, Bruneis, Bruneí
alms	esmola	4	elemosina, limosna, caridade, tuna, **esmola**, acato, almoina, almosno, pomano, milostenia
silicon	silicio	0	**silicio**, silicona, silicón, silisión, selicio, silício, xilicio, silico, silicia, silicone
organ	órgano,orgo	0	**órgano**, organo, orgue, orga, orgán, orgín, visco, argaño, ore, orgua
Prague	Praga	0	**Praga**, praga, Prague, Pragua, Praxa, Pragas, prague, Praca, Prágua, pragua
Bahamas	Bahamas	0	**Bahamas**, Bahama, Bahames, bahama, bahamas, Bahamás, Baamas, bahamás, bahames, Bahame
scrotum	escroto	0	**escroto**, paquete, coleo, cúleo, folículo, colia, croto, colla, escrota, escloto
mammal	mamífero	0	**mamífero**, mamalia, mamífaro, mamal, mamifero, mamálico, mamalía, mamífera, mamífico, mamífer
strike	folga,paro	7	golpe, bot, cop, vaga, greve, palo, ataque, **paro**, pic, bamba
Naples	Nápoles	0	**Nápoles**, Nápols, Nápoli, Nápole, Nápolis, Nápol, Neapolis, Nápola, Neápolis, Napol
sushi	sushi	0	**sushi**, sushí, subshi, sush, aperisushi, aperisus, suchi, sushín, sohi, sushin
toilet paper	papel hixiénico	15	papel hixénico, aniterxio, confort, conforte, papel hixenico, carta ixenica, papel higiénico
gas station	gasolineira	0	**gasolineira**, bomba, grifo, gasolineiro, distributor, servicentro, bencineiro, bencineira, benzineiro, filing
resin	pez,resina,recina	0	**resina**, rasa, resiña, moco, mugo, verniz, pece, reina, rosina, recina
clever	avisado	440	hábil, áxil, bravo, astuto, intelixente, listo, destro, inxenioso, cuca, teso
Sahara	Sáhara	2	Sara, Sahara, **Sáhara**, Saara, sahara, sara, sáhara, saara, Sàhara, Sará
etc.	etcétera	1	etc., **etcétera**, etcetera, etc, ecc., ..., ecetera, et cetera, etceteira, etetera
Cold War	Guerra fría	8	Guerra Fría, Guerra Fria, Guerra freda, Guerra Frida, Guera Freda, Guerra freia, Guerra froide, Guerra Freja
mechanics	mecánica	0	**mecánica**, mecánico, mequánico, mechánico, mecania, mecànico, mecànica, mehánico, mecanica, megánico
Gabon	Gabón	0	**Gabón**, xabón, Xabón, Gabon, Gabonia, Xabon, Jabón, xabon, gabón, Gabán
resistance	pulmón,resistencia,treina	0	**resistencia**, pulmón, polmón, oposición, palmón, aguante, rexistencia, ocursación, renitencia, repugnancia
werewolf	lobishome,licántropo	1	licantropo, **licántropo**, garú, lobisome, lupinoto, lobisona, outo, pricólico, lulo, bzou
Latin	latín	0	**latín**, latino, latina, Latín, Latino, latén, látimo, laten, lateno, limba latina
diameter	diámetro	0	**diámetro**, diametro, diameta, diamete, dimetiente, diàmetro, diameto, diámetros, diametros, diamét
regiment	rexemento,bandeira	8	reximento, reximiento, cohorso, cohors, cohor, rexemente, reximente, regimento, **rexemento**, reximentos
thrush	chalra,malvís,arnelo,tordo	0	**tordo**, turdo, torde, mirlo, muguete, griva, merlo, sapito, candidose, mugueto
USSR	URSS	0	**URSS**, URS, ORSS, RSU, uRSS, UrSS, ERSS, WRSS, URSE, URSA
policy	póliza,política	0	**política**, político, poliza, polisa, actitude, apólice, policia, reglamento, policio, policía
snooker	sinuca	3	billar, bilar, biliardo, **sinuca**, ventana, restornar, esteca, billardo, bisar, bilardo
Samarkand	Samarcanda	0	**Samarcanda**, Samarcande, Samarcand, Samarkanda, Sarmagante, Maracanda, samarcanda, Samarcando
Vishnu	Vishnu	5	Visnú, Vixnú, Vixnu, Vishnú, Visnu, **Vishnu**, Vixno, Visno, Vijnú, Vixhnú
decade	década,decenio	0	**decenio**, década, decada, deca, decina, dezena, décade, decas, deceno, decade
microbe	microbio	0	**microbio**, microbo, xerme, microbe, microba, mícrobo, mícrobe, mecrobio, microb, microbia
berkelium	berkelio,berquelio	0	**berkelio**, berquelio, bercelio, berchelio, berkeli, berkélio, berkelo, berKelio, berkelí, berkelío
thulium	tulio	0	**tulio**, tolio, túlio, tulo, tulío, tullo, thulio, tolo, tulho, tolir
adultery	adulterio	0	**adulterio**, adúltero, tradimento, adulteiro, crime, crimen, adúltera, adiltar, adultero, adultria
central bank	banco central	0	**banco central**, banca central, banque central, banc central, bance central, banco cintral, banqua central
fax	fax	0	**fax**, telecopia, telecopía, facsímil, teléfaxo, telecopior, teléfax, faz, telefaxe, telecopio
Mount Everest	Monte Everest	1	Everest, **Monte Everest**, monte Everest, Evereste, monso Evereste, monte Evereste, Everesto, Euereste
harem	harén	2	harem, harém, **harén**, serrallo, harema, serralio, farén, haremo, serral, haré
ace	ás	6	as, ace, iota, es, craque, dio, **ás**, campión, crack, dío
nebula	nebulosa	0	**nebulosa**, nebla, nebuloso, nébua, nebuleo, nevulosa, néboa, nebra, névoa, nebolosa
surprise	sorpresa	0	**sorpresa**, golpe, comoción, surpresa, inopinato, suspresa, surpriso, meravilla, surprisa, surprise
-ism	-ismo	0	**-ismo**, -isa, -asmo, -ísmo, -ista, -esmo, -izmo, -esa, -isma, -asma
Latvian	letón	0	**letón**, letona, Letón, lituano, leto, Letona, letone, leton, letão, letán
necrosis	necrose	0	**necrose**, necrosis, necrosa, necrozar, necrosar, necroso, necrosio, necrote, negrose, necrosie
Cancer	Cáncer	0	**Cáncer**, cáncer, Cranco, Cancro, Cámbaro, cranco, cancro, Raculuir, Cancer, Cáncro
dynamite	dinamita	0	**dinamita**, dinamite, diñamita, dinamito, diñamite, dinamida, dinamista, dinamitis, diamita, dinamitá
goldsmith	ourive	22	orfebre, aurario, orafa, orive, orafo, oribe, orífice, aurar, ourives, aurífice
Chicago	Chicago	0	**Chicago**, chicago, Khicago, Cicago, Xicago, Kicago, Chícago, Jicago, quicago, Cxicago
flamethrower	lanzachamas	17	lanciafuoco, lanciafiame, lanceflama, lanzaflames, bitaflomás, lanza chamas, xirlafiar, lanzaflamas, lancaflamas
kiosk	quiosco	0	**quiosco**, quiosque, estanque, chiosco, glorieta, pavellón, kiosque, chosco, ciosque, kiosco
Old Testament	Antigo Testamento	0	**Antigo Testamento**, veterotestamentario, antigo Testamento, Antico Testamento, Vetus Testamento

Table 7.17: Results on Galician, where the cognate model was the only successful model.

Concept	Gold	Gold Idx	Top Model Hypotheses
necktie	вратовръ́зка,вратовръзка	3165	гушо, сламо, гърля, нешия, нея, нося, небия, шияв, гавко, нев
gas station	бензиноста́нция	1308	газгара, бензинсърця, бензинставя, газолинсърця, газолинставя, бензинточка
supernatural	свръхестествен	66	огол, обекар, сбекар, нагол, оами, забекар, отбекар, вбекар, набекар, избекар
fishing cat	котка рибар	1022	мацко, банко, рибо, птичо, птицо, страно, рибис, рибас, коткоте, рибав
continuity	непрекъснатост	82	отия, траия, вамия, отие, ипол, отория, иост, морие, икаца, траие
covet	жадувам,пожелавам	10778	пос, сглася, наче, нас, ходе, схваля, оте, полус, схвалба, сдобре
opposite	срещу	10741	напо, опо, пос, нав, скрай, нада, спак, нас, нао, плюсс
patronymic	бащино име	8631	бияза, бащо, шефо, бащоколо, отцо, отциме, бащиме, бащавред, избягвамо, бащаза
scavenger	лешояд	886	лещдо, лешза, лешс, пътвек, мършас, лещда, лешкаца, мършав, лешвек, мършера
ptomaine[4]	трупна отрова	5888	щаяд, трупа, дана, поемо, тяла, съща, дас, щана, щатровя, щас
blacklist	че́рен спи́сък	1002	черчер, черчерен, черива, чержелая, лошсъвет, черискам, мракчер, черкенар

Table 7.18: Results on Bulgarian, where the compositional model was the only successful model.

Concept	Gold	Gold Idx	Top Model Hypotheses
seed	żerriegħa	2	sperma, liba, **żerriegħa**, ħabba, ħawwel, xitla
mercury	merkurju	0	**merkurju**
happen	ħabat,ġara,seħħ	0	**ġara**, ġara, laqat, ħabat
hang	għallaq,dendel	0	**dendel**, għereq, għarraq
liberty	ħelsien,libertà	0	**ħelsien**, libertà
catch	sab,qabad	1	jassar, **qabad**, jassar, qabad, ħa, dam, jtul, xeħet, ħasad, ħa
clever	bravu	0	**bravu**
instrument	għodda	0	**għodda**, istrumenti mużikali, magna, qies, kejl
adultery	żina	1	żinja, **żina**
stair	taraġ	0	**taraġ**
occur	ħabat,seħħ	2	ġara, laqat, **ħabat**, ġera
glad	ferrieħ,ferrieħi,ferħan	0	**ferrieħ**, ferħan, kuntenti, ferrieħi, hieni, kuntent
ascend	għola,tela’	0	**tela’**, għola, qam, tela’, għola, qam, tela’, għola, qam, tela’
itch	qaras	2	gidem, igdem, **qaras**
remedy	duwa	1	dewwa, **duwa**, tazza, kikkra
disperse	xerred	0	**xerred**
follower	sieħeb	0	**sieħeb**, għarus, sieħeb, ħabib, xxierek, sieħeb, soċju
suckle	reda’,redda’	0	**redda’**, ners, infermier, infermiera, reda’
pierce	nifed	1	ppenetra, **nifed**, ppenetra, nifed, nifed
accomplish	wettaq	4	lesta, lesta, laħaq, laħaq, **wettaq**, għamel, wettaq, rċieva, kiseb
revive	ħeja,ġedded	0	**ġedded**
screw	niek	0	**niek**, nejka, batta, sawwat, taħan, laqat, daqq, mellaħ, ħerba
redeem	feda	1	rahan, **feda**, welled, feda, ħeles, wieled, wiled, biegħ

Table 7.19: Results on Maltese, where the lexical relation model was the only successful model.

ble 7.19.

In terms of model combination, the three models generate vastly different sized n-best lists: the cognate model's n-best list length is the order of 1,000 hypotheses, the compositional model generates on the order of 10,000 hypotheses, while the lexical relation model generates on the order of 100 hypotheses. Combining the results using the rank-based voting strategy is effective when not all models have generated the correct hypothesis. Table 7.20 presents results on Irish test words, showing the index of the gold translation in the n-best lists of each model, as well as the index in the hypothesis list combined using rank-based voting. When more than one model has correctly predicted the translation, combining the hypothesis lists and reranking occasionally pushes the gold translations further down the list. However, this is not a problem, as discussed above.

In summary, I have shown successes of the three models of cognates, compositional word formation, and lexical relations at generating translations of unknown concepts in low-resource target languages. While on their own effective at certain classes of words, these models can be combined using a simple but effective model combination approach to realize drastic improvements in prediction accuracy, thus leveraging multiple model's experience. Future work may explore more sophisticated model combination strategies.

7.8 A Dense Induced Bible Language Core Vocabulary Translation Dictionary

The culmination of the multiple efforts included in this dissertation naturally lead to the construction of an artifact: a massive induced core vocabulary dictionary. I successively build up this artifact of a dense core vocabulary translation dictionary, starting with Wiktionary, followed by the addition of Bible word alignments, and the contributions of the various models of word formation. To start, I focus my efforts the 1,106 languages for which we have a Bible (McCarthy, Wicks, et al., 2020), and ensure coverage over the top 1,000 core vocabulary concepts from the core vocabulary described in Section 3.2.

Wiktionary. I start with Wiktionary as a source of ground truth translations. The

Concept	Gold	Cog Idx	Comp Idx	Rel Idx	Combined Idx
blood	gaol,flann,sampla fola,cró,fuil	0	10102	0	0
white	geal,bán	0	11598	5	0
light	léas,spéir,sorcha,geantraí,coinneal,solas,soilse	0	23835	4	0
tea	tae	.	11379	.	12207
frog	frog,loscann,loscán,froga	4	10143	.	2
seed	síol,pór	0	10531	5	0
Friday	Aoine	.	10546	.	11126
die	faigh bás,éag,básaigh,caill	1	13253	.	1
deer	fiara,fia,os	0	60	.	0
thousand	míle	0	.	.	5
go	gabh,téigh,imigh	1	11560	1	0
lung	scamhóg	.	10855	.	11259
whale	míol mór	.	.	.	.
now	adrásta,anois,anuas	.	.	.	.
pine	ailm,giúis,péine	20	.	0	22
give	tabhair	8	12060	67	2
fork	adhal,glac,gabhlóg,gabhal,forc,píce	0	11483	1	0
south	deisceart	.	.	.	.
laugh	déan gáire,gáir	8	10554	.	3
nineteen	naoi déag	.	9609	.	10504
thumb	ordóg,ladhar	3	.	.	4
dew	drúcht	.	.	.	.
weapon	arm	0	10360	2	1
well	tiobraid,tobar	1	10728	.	4
want	teastaigh ó,is mian le	.	.	27	809
box	crann bosca,bucas,bosca	56	8	.	62
sickle	corrán	1	.	.	1
vulva	pit	1	.	0	0
ink	dúch	.	12709	.	13091
bird	éan	0	10228	.	3
Israel	Stát Iosrael,Iosrael	0	.	.	0
knowledge	aithne,eol,eolas,fios	0	21	0	0
stick	bata,camán,craobh,maide,maide haca	0	13720	.	7
New Zealand	Nua-Shéalainn	.	10114	.	10615
student	scoláire,dalta,mac léinn	.	.	1	7218
belt	buille,crios,beilt	1	11994	25	0
ice cream	reoiteog,uachtar reoite	14	.	.	14
enter	iontráil	0	.	.	1
bride	brídeach	.	.	.	.
saliva	seile	0	.	1	1

Table 7.20: Indices of the correct translation in the hypothesis lists for Irish test words. A dot (.) indicates that the gold translation was not in the n-best list, not that the model did not produce any hypotheses.

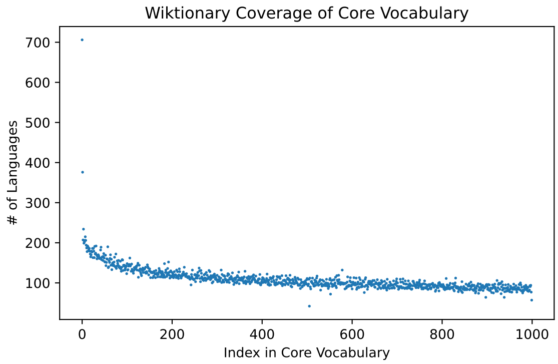


Figure 7.3: Wiktionary coverage of core vocabulary.

coverage of Wiktionary over core vocabulary words is shown in Figure 7.3, where the x-axis is the index of the concept in the sorted core vocabulary list, and the y-axis is the number of languages containing a translation of that concept. The shape of this graph follows a typical power law distribution, which I have also found for the relationship between languages in Wiktionary and the number of entries for each language. Note that the plot is almost monotonically decreasing, because existence in multiple dictionaries is the criterion that Wu, Nicolai, and Yarowsky (2020) used for ordering their core vocabulary list.

Bible. While the Bible is the most translated document in the world, we do not have translations into all 7,000 languages in the world. Nevertheless, the Bible is a useful source of translations in low-resource languages. In fact, there are 256 languages for which we have Bibles but do not have entries in Wiktionary (McCarthy, Wicks, et al., 2020). To obtain lexical translations from the Bible, I compute word alignments using fast_align (Dyer, Chahuneau, and N. A. Smith, 2013), from every language to English. Because the alignment process is noisy, for each source word, I keep the top 20 aligned target words,

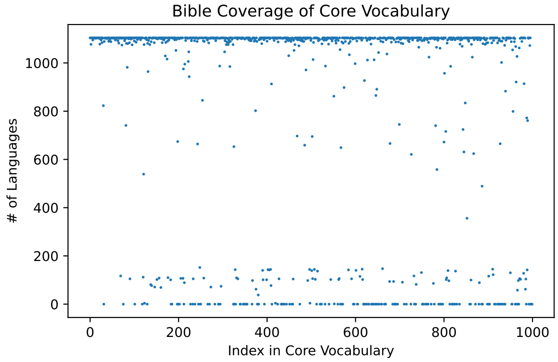


Figure 7.4: Bible coverage of core vocabulary.

along with its associated alignment probability.

In terms of coverage over core vocabulary, the Bible contains the majority of words in the top 1,000 words of the core vocabulary list. Figure 7.4 shows coverage of translations of the Bible over the sorted core list. Note that since coverage is calculated over 1,100 translations of the Bible, rather than on a single English edition, some languages may cover a certain concept while others do not, either due to variations in translations or because the Bible translation for some languages is incomplete.

There are 152 core concepts that do not exist in the Bible. They are listed alphabetically as follows:

> Afghanistan, Albania, Antarctica, April, August, Australia, Austria, Belgium, Canada, Christmas, December, Denmark, Estonia, Europe, February, Finland, France, French, German, Germany, Greenland, I love you, Iceland, January, July, June, Mexico, Monday, November, Russia, Russian, September, Sweden, Thursday, Tuesday, Turkey, United States of America, Wednesday, Wikipedia, airplane, airport, alphabet, anus, armpit, bamboo, banana, be able to, beaver, bicycle, brain, bus, butterfly, button, cabbage, capital city, carrot, cat, century, chicken, chocolate, cigarette, claw, cockroach, coconut, coffee, computer, cough, crab, crocodile, dandruff, dictionary, dolphin, duck, eel, eggplant, electricity, eyebrow, eyelash, feather, fingernail, ginger, glove, good morning, goose, gun, hospital, human being, hydrogen, kidney, kitchen, lemon, louse, maize, mango, mathematics, monkey, mosque, mosquito, moss, moustache, mushroom, newspaper, noun, old man, onion, orange, otter, oxygen, page, parrot, passport, peach, pear, pencil, pepper, pineapple, planet, potato, pumpkin, puppy, rat, rezpublic, rhinoceros, shark, skunk, sleeve, spleen, squirrel, steam, strawberry, sweet potato, tea,

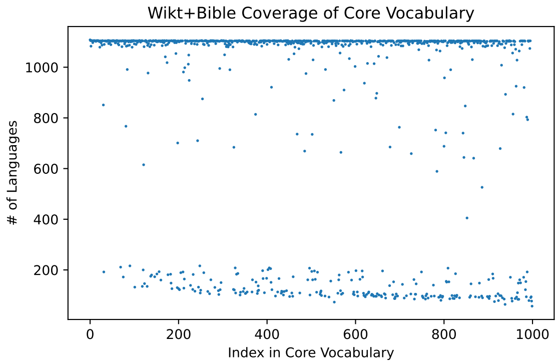


Figure 7.5: Wiktionary+Bible coverage of core vocabulary.

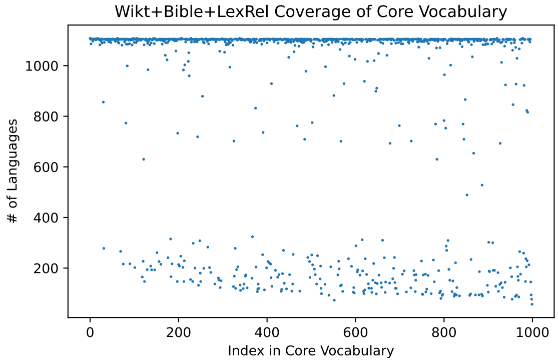


Figure 7.6: Wiktionary+Bible+Lexical Relation coverage of core vocabulary.

telephone, television, testicle, thank you, tick, tiger, toad, tobacco, tomato, toucan, turkey, umbrella, vagina, verb, volcano, vulva, wake up, wasp, watermelon, zero

Many of these concepts, including country names, month names, and modern terminology (e.g. computer, newspaper, telephone) are essential for daily life but are conspicuously missing from the Bible. This shows the deficiencies of relying solely on text in a specialized domain for translations. Also see discussion in Section 3.2.

Lexical Relation Extensional Model. I apply the extensional translation method to all the core vocabulary concepts. For words that do not yet exist in the Bible or Wiktionary, the lexical translation method generated a total of 12,032 new (concept, language) pairs.

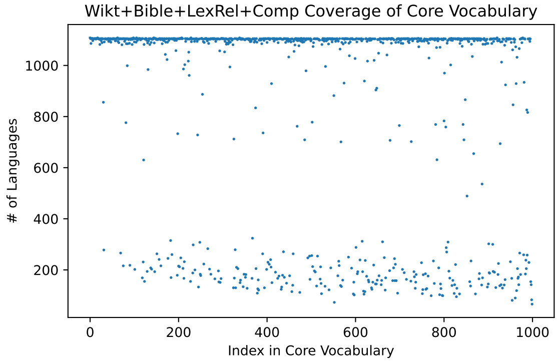


Figure 7.7: Wiktionary+Bible+Lexical Relation+Compositional coverage of core vocabulary.

A sample of induced translations appears in Table 7.21.

Many of these are related words which, while not exact translations, are close enough to the target concept for communication about topics related to the concept. For example, *кумдус* ‘beaver’ for ‘otter’, *lac* ‘plate’ for ‘spoon’, and *letswai* ‘salt’ for ‘pepper’. Figure 7.6 shows the coverage over the core vocabulary using the combined translations from Wiktionary, the Bible word alignments, and the lexical translation (extensional) model.

Compositional Model. While I have shown that many core vocabulary words are not likely to be compositional, I apply the model of compositional word formation (Wu and Yarowsky, 2018c) to generate compositional words for core vocabulary, so that end users of the resulting dictionary have the option of using these hypothesized translations if they wish. The compositional word formation model contributes 7.4 million induced translations for 115K (concept, language) pairs. Combined with translations from Wiktionary, the Bible word alignments, and the lexical translation model, coverage is shown in Figure 7.6. However, the compositional model does not contribute many new transla-

Concept	Lang	Induced Translations
urine	anv	mana (0.138)
butter	mww	mis (0.012), ntxuav (0.012)
cook	nlc	soko (0.003)
goose	fij	ga (0.308)
son-in-law	kal	ningaaq (0.783), sakeq (0.087)
berry	jiv	jinkiai (0.177)
otter	alt	кумдус (0.583)
mouse	krc	къаплан (0.007), агъаз (0.007)
orphan	kjh	хул (0.091)
tin	amm	ono (0.010)
cotton	gsw	Lätsch (0.015), Härre (0.015)
thumb	tcs	pingga (0.600)
liver	cgc	tagipusuon (0.471), arey (0.029)
sleeve	itv	abaha (0.034)
pear	tbl	lanas (0.022)
spoon	quc	lac (0.017)
star	mwf	njeyrt (0.007)
puppy	hmo	sisiu (0.120)
ash	tsn	setlhare (0.019), leru (0.019)
tiger	hil	balabaw (0.018), kuring (0.018)
pepper	nso	letswai (0.022)

Table 7.21: Translations induced from the extensional model

tions, because this model composes existing known words. See (Wu and Yarowsky, 2018c) for further analysis.

Cognate Models. I employ a multilingual cognate generation model (Wu and Yarowsky, 2018b) for the task of dictionary induction. In contrast to the existing models described above, cognate models can generate completely new word forms as long as a single cognate pair exists for a target language. This allows the cognate models to bring coverage over the core vocabulary to 100%. I have previously shown the success of these models in successfully inducing missing dictionary entries in several works (Wu and Yarowsky, 2020a; Lewis et al., 2020; Wu and Yarowsky, 2021). I direct the reader to these publications as well as Chapter 5 for more in-depth analysis.

Direct Neural Models. Finally, I include in the model combination the results of the character-basd direct neural model, which generates hypothesized translations of concepts across languages. Recall that this is essentially a transliteration method from English.

The models were applied on all concepts in the core vocabulary list, including those that already exist in Wiktionary. The resulting dictionary is distributed as a collection of tab-separated files totaling 5.7 GB (uncompressed) and contains over 200M new induced translations. Each entry in this dictionary contains both the provenance of translation as well as the probability given by each of the six sources described above (the probability for entries in Wiktionary is 1). A sample of this artifact is shown in Table 7.22. I envision this artifact to be a tremendous resource for low-resource machine translation, where this dictionary can be used as additional training bitext or serve as a precomputed unigram language model to identify unknown words at runtime.

7.9 Retraining with Induced Data

Here, I briefly examine an iterative approach, where I utilize this new expanded dictionary to retrain an existing translation generation model. I experimented with the compositional model from Chapter 4, using the existing learned compositional recipes but generating with a new dictionary of induced translations of top 1,000 core vocabulary concepts. Testing on the test set described in Section 7.1, I find no improvement in compositional generation performance. This may be due to the fact that many of the test concepts are not compositional, and for the compositional concepts, the main issue with this model was not that the component translations do not exist, but rather that the word composition process was not generalizable (Section 4.1.5). In addition, many compositional concepts in the test set are formed from components outside of the top 1,000 core concepts that were induced across thousands of languages, e.g. Buckingham^{-1} Palace^{1100}, mobile^{-1} phone^{6114}, neck^{77}tie^{1027}, and olive^{1265} tree^{28}, where the superscript numbers indicate the index of the word in the core vocabulary concept list. Nevertheless, I believe this loop of generating and retraining is an important process for refining my models' predictions, and I propose avenues of future research along this line in the next chapter.

Source	Word	Probability
bible	cão	-0.946008
bible	lambendo	-1.45534
bible	ditados	-1.60102
bible	abrange	-1.60593
bible	ganidos	-1.6094
cog	can	-5.604016
cog	cán	-5.953931
cog	cacan	-6.026464
cog	cān	-6.200143
cog	cana	-6.456468
comp	cãoneto	-3.428665
comp	cãohomem	-3.428679
comp	cãoavô	-3.428763
comp	cachorroneto	-3.429380
comp	cachorrohomem	-3.429394
direct	carro	-4.643856
direct	cacho	-4.673592
direct	colo	-4.703990
direct	capa	-4.735005
direct	canto	-4.766701
lr	mulherengo	-1.172038
lr	canino	-2.424798
lr	rafeiro	-2.587325
lr	cachorrinho	-2.855588
lr	totó	-2.855588
wikt	cachorro	0.0
wikt	perro	0.0
wikt	cachorrinho	0.0
wikt	cão	0.0
wikt	kasor	0.0

Table 7.22: Translation dictionary contents for the Portuguese word for DOG. Note that these probabilies are log probabilities.

Chapter 8

Conclusion

While there exist over seven thousand languages in the world, language technologies exist only for a tiny percentage of these languages, which we may call high-resource languages. The large majority of the 6,000+ remaining languages simply do not yet have enough data for developing data-intensive high-accuracy language technologies such as machine translation. Certain modern techniques including multilingual embeddings have been developed to solve the issue of lack of training data, but these methods require at least a substantial monolingual corpus on which to train the embeddings.

This dissertation pioneers the relatively new and promising field of computational etymology, which spans word formation, word origins, and the relationships between words across languages. The computational study of word etymology is a key step in developing comprehensive dictionaries for low-resource languages, which will enable better communication with and language-technology support for underserved language communities. To tackle the challenges of computational modeling words' formation processes and origins, this dissertation presents novel algorithms, methods, and tools, detailed in the preceding chapters.

In Chapter 3, I presented *Yawipa*, a novel high-performance Wiktionary parsing, extraction, and normalization system, which I developed to directly support the entirety of the work in this dissertation, providing very broad-coverage training and evaluation ground-truth data sets. Yawipa is a comprehensive and extensible framework for parsing all the types of information stored as structured or semi-structured data in Wiktionary. It contains a comprehensive parser for the diverse classes of linguistic information stored in the English edition of Wiktionary and also parsers for several other editions. Compared to existing work, Yawipa extracts and normalizes Wiktionary data in much greater detail and breadth, especially with regard to etymology, pronunciations, morphology, and translations.

In Section 3.2, I presented a novel practical and formal criterion for the construction of core vocabulary sets based on the number of foreign language dictionaries containing a specific concept. This criterion enables a ranking of concepts by essentially aggregating votes from thousands of lexicographers. Compared to existing core vocabulary lists, which are often small or language specific, this new core list constructed using this criterion is better suited for the task of dictionary induction and is used to prioritize concepts for inclusion in the massively multilingual dictionary instantiated in the dissertation.

I approach the task of massively multilingual dictionary induction through word formation, which comprises the bulk of this dissertation, and is an integral part of computational etymology. The techniques I developed for computational word formation are based on principles in linguistics and are directly applicable for low-resource languages. My multilingual models learn from the thousands of languages in Wiktionary, a substantially larger set of training languages than in prior work. The compositional model described in Chapter 4 learns cross-lingual probabilistic recipes of compound word formation using a variety of compound splitting mechanisms. These universal compound analysis and generation models can translate both into and out of English using probabilistic models of different parts of the compounding process. These models account for a large variety of linguistic compounding processes including concatenation, epenthesis, and elision. While much existing work focuses on a single language or a handful of languages, these models, trained on hundreds of languages, are also applicable to hundreds of new languages and can accurately generate unseen words into target languages.

The cognate models described in Chapter 5 exploit the relationships between languages around the world to generate potential cognate translations. These models are trained on cognate pairs, which are not readily available for most languages. To solve this issue, I developed a novel clustering procedure with weighted edit distance to automatically acquire sets of cognates in related languages from only a readily available multilingual dictionary. Using these cognate sets, multilingual models of cognates and sounds shifts are trained to learn sound-shift processes between related languages and can accurately recover held-out cognates.

As a straightforward model that does not require substantial training, the lexical relation model in Section 4.2 models the probability of existing words as acceptable translations for unknown concepts. This model learns translational equivalence between synonyms, hypernyms, hyponyms, and co-hyponyms from WordNet, which have not all been studied in prior work. This model is especially applicable for languages with little training data.

In addition to modeling the processes of word formation, In Chapter 6 I also realize additional novel components in the modeling of computational etymology, including novel experiments with neural classification models to determine the language from which a word originates and the etymological relation between a word and its donor. I also developed regression approaches to identify the year in which a word enters a language. Together, the components of this and preceding chapters form the basis of novel, broad-coverage, massively-multilingual framework of computational etymology may serve as a foundational framework for additional computational work and shared tasks in this previously understudied field, as well as providing potential insights to benefit the work of lexicographers and linguists of low-resource languages.

Chapter Chapter 7 presents the culmination of the dissertation: effective system combination of the multiple cognate, compositional, and lexical relation models applied to the task of unknown word generation. It also presents an induced translation dictionary further incorporating Bible-multitext-learned and dictionary-extracted translations of core vocabulary. The combined framework is instantiated and evaluated on the extremely challenging task of unknown word generation in the absence of a monolingual corpus in the target language, thus without a language model for verification, ranking, or context-based embedding models, which is the *de facto* situation with the most of the 6,000 languages of the world currently lacking nontrivial and practically identifiable monolingual corpora. The chapter shows that each of its three models for component combination have complimentary strengths, and together with the all of the previous chapters of the dissertation, they realize an instantiated induced dictionary as a lasting and constantly growing artifact that will facilitate both further practical applications and research in linguistics, machine translation, and other NLP technologies for the low-resource and very-low-resource languages that form the large majority of the world's languages.

8.1 Future Work and Final Remarks

Much of human knowledge is encoded in language, and every language has a unique body of knowledge that is inaccessible for those who do not know the language. The overarching goal of my research to break down language barriers, so that for anyone in the world, no matter what language they speak, they should be able to read anything, communicate with anyone, and have universal access to knowledge. Throughout my PhD, I have worked on technologies for low-resource languages, focused on solving the task of unknown word translation. The approaches and models presented in this dissertation are applicable to the very diverse and massively-multilingual scope of low resource languages around the world. But in the real world, when predicting unknown words in a language, these models often face the particular challenge of generating translations which are not yet attested in monolingual wordlists and have no monolingual corpora for exploiting contextual similarity via embeddings or other techniques, and for which there is yet no ground truth for evaluation. For maximum applicability, we need real humans to validate, edit, and augment these model predictions.

For future work, I plan to build an online crowdsourced research platform for native speakers in the world to easily contribute knowledge of their own language. This platform would support, as well as learn from, thousands of underserved language communities around the world. In terms of this kind of platform, existing solutions like Wiktionary, though also crowdsourced, are not ideal, because users must be tech-savvy to contribute. Instead, we need something that is easy to use and accessible by anyone. This platform could exist as a web app and/or a mobile app that anyone can download on their phone. It would display a translation matrix, where every cell is editable by users who would log in and make contributions. Other users can vote on the contributions and indicate their confidence in proposed translations.

This proposed app will be a research platform in which we can run studies to see what are the best ways to elicit concept translations from native speakers. Developing this will be a multi-year collaborative effort between people in computer science, linguistics, psychology, and other interdisciplinary fields. Contributions from human users can be used to validate my models predictions about new words, but will also serve as new data which can be used to retrain my models, forming a continuous feedback loop (described in Chapter 7) where machines help humans and humans helps machines.

Humans are an integral part of machine learning. After all, where did all our data come from? I strongly believe that machine learning should ultimately help and benefit humans. The combination of the models and techniques proposed in this dissertation, along with reinforcement and contributions from human speakers, will bring us closer to solving the grand challenge of universal translation between all the world's languages, leading our society into a globally connected world where everyone has universal access to knowledge.

References

Abrahamsson, Emil, Timothy Forni, Maria Skeppstedt, and Maria Kvist (Apr. 2014). “Medical text simplification using synonym replacement: Adapting assessment of word difficulty to a compounding language”. In: *Proceedings of the 3rd Workshop on Predicting and Improving Text Readability for Target Reader Populations (PITR)*. Gothenburg, Sweden: Association for Computational Linguistics, pp. 57–65. URL: https://aclanthology.org/W14-1207.

Ács, Judit (May 2014). “Pivot-based multilingual dictionary building using Wiktionary”. In: *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC’14)*. Reykjavik, Iceland: European Language Resources Association (ELRA), pp. 1938–1942. URL: http://www.lrec-conf.org/proceedings/lrec2014/pdf/864_Paper.pdf.

Ács, Judit, Katalin Pajkossy, and András Kornai (Aug. 2013). “Building basic vocabulary across 40 languages”. In: *Proceedings of the Sixth Workshop on Building and Using Comparable Corpora*. Sofia, Bulgaria: Association for Computational Linguistics, pp. 52–58. URL: https://aclanthology.org/W13-2507.

REFERENCES

Ahmad, Khurshid (2000). “Neologisms, nonces and word formation”. In: *Proceedings of the Ninth EURALEX Intern ational Congress*, p. 71.

Algeo, John and Adele S Algeo (1993). *Fifty years among the new words: A dictionary of neologisms 1941-1991*. Cambridge University Press.

Andrade, Daniel, Masaaki Tsuchida, Takashi Onishi, and Kai Ishikawa (June 2013). “Translation Acquisition Using Synonym Sets”. In: *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Atlanta, Georgia: Association for Computational Linguistics, pp. 655–660. URL: https://aclanthology.org/N13-1075.

Artetxe, Mikel, Gorka Labaka, and Eneko Agirre (July 2019). “An Effective Approach to Unsupervised Machine Translation”. In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Florence, Italy: Association for Computational Linguistics, pp. 194–203. URL: https://aclanthology.org/P19-1019.

Baayen, R Harald, Richard Piepenbrock, and Leon Gulikers (1996). “The CELEX lexical database (cd-rom)”. In.

Baldwin, Timothy, Jonathan Pool, and Susan Colowick (Aug. 2010). “PanLex and LEXTRACT: Translating all Words of all Languages of the World”. In: *Coling 2010: Demonstrations*. Beijing, China: Coling 2010 Organizing Committee, pp. 37–40. URL: https://aclanthology.org/C10-3010.

Banerjee, Satanjeev and Alon Lavie (June 2005). “METEOR: An Automatic Metric for MT Evaluation with Improved Correlation with Human Judgments”. In: *Proceedings of the*

ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization. Ann Arbor, Michigan: Association for Computational Linguistics, pp. 65–72. URL: https://aclanthology.org/W05-0909.

Bañón, Marta, Pinzhen Chen, Barry Haddow, Kenneth Heafield, Hieu Hoang, Miquel Esplà-Gomis, Mikel L. Forcada, Amir Kamran, Faheem Kirefu, Philipp Koehn, Sergio Ortiz Rojas, Leopoldo Pla Sempere, Gema Ramírez-Sánchez, Elsa Sarrías, Marek Strelec, Brian Thompson, William Waites, Dion Wiggins, and Jaume Zaragoza (July 2020). "ParaCrawl: Web-Scale Acquisition of Parallel Corpora". In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Online: Association for Computational Linguistics, pp. 4555–4567. URL: https://aclanthology.org/2020.acl-main.417.

Bartlett, Susan, Grzegorz Kondrak, and Colin Cherry (June 2008). "Automatic Syllabification with Structured SVMs for Letter-to-Phoneme Conversion". In: *Proceedings of ACL-08: HLT*. Columbus, Ohio: Association for Computational Linguistics, pp. 568–576. URL: https://aclanthology.org/P08-1065.

Batsuren, Khuyagbaatar, Gabor Bella, and Fausto Giunchiglia (July 2019). "CogNet: A Large-Scale Cognate Database". In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Florence, Italy: Association for Computational Linguistics, pp. 3136–3145. URL: https://aclanthology.org/P19-1302.

Bauer, Laurie (2009). "Typology of compounds". In: *The Oxford handbook of compounding*.

Beinborn, Lisa, Torsten Zesch, and Iryna Gurevych (Oct. 2013). “Cognate Production using Character-based Machine Translation”. In: *Proceedings of the Sixth International Joint Conference on Natural Language Processing*. Nagoya, Japan: Asian Federation of Natural Language Processing, pp. 883–891. URL: https://aclanthology.org/I13-1112.

Bergsma, Shane and Grzegorz Kondrak (June 2007). “Alignment-Based Discriminative String Similarity”. In: *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*. Prague, Czech Republic: Association for Computational Linguistics, pp. 656–663. URL: https://aclanthology.org/P07-1083.

Bond, Francis and Ryan Foster (Aug. 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)*. Sofia, Bulgaria: Association for Computational Linguistics, pp. 1352–1362. URL: https://aclanthology.org/P13-1133.

Booij, Geert (2009). *Compounding and construction morphology*. na.

Brew, Chris, David McKelvie, et al. (1996). “Word-pair extraction for lexicography”. In: *Proceedings of the 2nd international conference on new methods in language processing*. Citeseer, pp. 45–55.

Browne, Charles (2014). “A new general service list: The better mousetrap we’ve been looking for”. In: *Vocabulary Learning and Instruction* 3.2, pp. 1–10.

Bungum, Lars and Stephan Oepen (May 2009). “Automatic Translation of Norwegian Noun Compounds”. In: *Proceedings of the 13th Annual conference of the European Asso-*

ciation for Machine Translation. Barcelona, Spain: European Association for Machine Translation. URL: https://aclanthology.org/2009.eamt-1.19.

Ciobanu, Alina Maria (2016). "Sequence labeling for cognate production". In: *Procedia Computer Science* 96, pp. 1391–1399.

Ciobanu, Alina Maria and Liviu P. Dinu (June 2014). "Automatic Detection of Cognates Using Orthographic Alignment". In: *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Baltimore, Maryland: Association for Computational Linguistics, pp. 99–105. URL: https://aclanthology.org/P14-2017.

Collier, Nigel, Hideki Hirakawa, and Akira Kumano (Aug. 1998). "Machine Translation vs. Dictionary Term Translation - a Comparison for English-Japanese News Article Alignment". In: *36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics, Volume 1*. Montreal, Quebec, Canada: Association for Computational Linguistics, pp. 263–267. URL: https://aclanthology.org/P98-1041.

Cook, Paul, Jelena Mitrović, Carla Parra Escartín, Ashwini Vaidya, Petya Osenova, Shiva Taslimipoor, and Carlos Ramisch, eds. (Aug. 2021). *Proceedings of the 17th Workshop on Multiword Expressions (MWE 2021)*. Online: Association for Computational Linguistics. URL: https://aclanthology.org/2021.mwe-1.0.

Dale, Edgar and Jeanne S Chall (1948). "A formula for predicting readability: Instructions". In: *Educational research bulletin*, pp. 37–54.

De Vaan, Michiel (2018). *Etymological dictionary of Latin and the other Italic languages.* Vol. 7. LEIDEN· BOSTON, 2008.

Denning, Keith, Brett Kessler, and William R Leben (2007). *English vocabulary elements.* Oxford University Press.

Deri, Aliya and Kevin Knight (Aug. 2016). "Grapheme-to-Phoneme Models for (Almost) Any Language". In: *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers).* Berlin, Germany: Association for Computational Linguistics, pp. 399–408. URL: https://aclanthology.org/P16-1038.

Derksen, Rick (2007). "Etymological Dictionary of the Slavic Inherited Lexicon (Leiden Indo-European Etymological Dictionary)". In.

Dictionary, Marriam-Webster (2006). *The Merriam-Webster Dictionary.* Merriam-Webster, Incorporated.

Dunn, Michael (2015). "Language phylogenies". In: *The Routledge handbook of historical linguistics*, pp. 190–211.

Dyen, Isidore, Joseph B Kruskal, and Paul Black (1992). "An Indoeuropean classification: A lexicostatistical experiment". In: *Transactions of the American Philosophical society* 82.5, pp. iii–132.

Dyer, Chris, Victor Chahuneau, and Noah A. Smith (June 2013). "A Simple, Fast, and Effective Reparameterization of IBM Model 2". In: *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human*

Language Technologies. Atlanta, Georgia: Association for Computational Linguistics, pp. 644–648. URL: https://aclanthology.org/N13-1073.

Fang, Alex Chengyu, Wanyin Li, and Nancy Ide (Dec. 2009). "Latin Etymologies as Features on BNC Text Categorization". In: *Proceedings of the 23rd Pacific Asia Conference on Language, Information and Computation, Volume 2*. Hong Kong: City University of Hong Kong, pp. 662–669. URL: https://aclanthology.org/Y09-2026.

Fellbaum, Christiane (2010). "WordNet". In: *Theory and applications of ontology: computer applications*. Springer, pp. 231–243.

Fillmore, Charles J (1988). "The mechanisms of construction grammar". In: *Annual Meeting of the Berkeley Linguistics Society*. Vol. 14, pp. 35–55.

Fischer, Roswitha (1998). *Lexical change in present-day English: A corpus-based study of the motivation, institutionalization, and productivity of creative neologisms*. Vol. 17. Gunter Narr Verlag.

Fourrier, Clémentine and Benoît Sagot (May 2020). "Methodological Aspects of Developing and Managing an Etymological Lexical Resource: Introducing EtymDB-2.0". English. In: *Proceedings of the 12th Language Resources and Evaluation Conference*. Marseille, France: European Language Resources Association, pp. 3207–3216. URL: https://aclanthology.org/2020.lrec-1.392.

Fromkin, Victoria, Robert Rodman, and Nina Hyams (2018). *An introduction to language*. Cengage Learning.

Frunza, Oana Magdalena (2006). "Automatic identification of cognates, false friends, and partial cognates". PhD thesis. University of Ottawa (Canada).

Gagné, Christina L, Kristan A Marchak, and Thomas L Spalding (2010). "Meaning predictability and compound interpretation: A psycholinguistic investigation". In: *Word Structure* 3.2, pp. 234–251.

Garera, Nikesh and David Yarowsky (2008). "Translating Compounds by Learning Component Gloss Translation Models via Multiple Languages". In: *Proceedings of the Third International Joint Conference on Natural Language Processing: Volume-I*. URL: https://aclanthology.org/I08-1053.

Goldberg, Adele E (2006). *Constructions at work: The nature of generalization in language*. Oxford University Press on Demand.

Gorman, Kyle, Lucas F.E. Ashby, Aaron Goyzueta, Arya McCarthy, Shijie Wu, and Daniel You (July 2020). "The SIGMORPHON 2020 Shared Task on Multilingual Grapheme-to-Phoneme Conversion". In: *Proceedings of the 17th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology*. Online: Association for Computational Linguistics, pp. 40–50. URL: https://aclanthology.org/2020.sigmorphon-1.2.

Grefenstette, Gregory (Nov. 1999). "The World Wide Web as a Resource for Example-Based Machine Translation Tasks". In: *Proceedings of Translating and the Computer 21*. London, UK: Aslib. URL: https://aclanthology.org/1999.tc-1.8.

Guevara, Emiliano, Sergio Scalise, Antonietta Bisetto, and Chiara Melloni (May 2006). “MORBO/COMP: a multilingual database of compound words”. In: *Proceedings of the Fifth International Conference on Language Resources and Evaluation (LREC’06)*. Genoa, Italy: European Language Resources Association (ELRA). URL: http://www.lrec-conf.org/proceedings/lrec2006/pdf/286_pdf.pdf.

Guldenoglu, Birkan (2016). “The effects of syllable-awareness skills on the word-reading performances of students reading in a transparent orthography”. In: *International Electronic Journal of Elementary Education* 8, pp. 425–442.

Gyanendro Singh, Loitongbam, Lenin Laitonjam, and Sanasam Ranbir Singh (Dec. 2016). “Automatic Syllabification for Manipuri language”. In: *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*. Osaka, Japan: The COLING 2016 Organizing Committee, pp. 349–357. URL: https://aclanthology.org/C16-1034.

Haspelmath, Martin and Uri Tadmor (2009). *Loanwords in the world’s languages: a comparative handbook*. Walter de Gruyter.

Hauer, Bradley and Grzegorz Kondrak (Nov. 2011). “Clustering Semantically Equivalent Words into Cognate Sets in Multilingual Lists”. In: *Proceedings of 5th International Joint Conference on Natural Language Processing*. Chiang Mai, Thailand: Asian Federation of Natural Language Processing, pp. 865–873. URL: https://aclanthology.org/I11-1097.

He, Yifan, Jinhua Du, Andy Way, and Josef van Genabith (July 2010). “The DCU Dependency-Based Metric in WMT-MetricsMATR 2010”. In: *Proceedings of the Joint Fifth Workshop on Statistical Machine Translation and MetricsMATR*. Uppsala, Sweden: Association for Computational Linguistics, pp. 349–353. URL: https://aclanthology.org/W10-1753.

Hendrickx, Iris, Zornitsa Kozareva, Preslav Nakov, Diarmuid Ó Séaghdha, Stan Szpakowicz, and Tony Veale (June 2013). “SemEval-2013 Task 4: Free Paraphrases of Noun Compounds”. In: *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*. Atlanta, Georgia, USA: Association for Computational Linguistics, pp. 138–143. URL: https://aclanthology.org/S13-2025.

Huang, Chu-Ren, I-Li Su, Jia-Fei Hong, and Xiang-Bing Li (2005). “Cross-lingual Conversion of Lexical Semantic Relations: Building Parallel Wordnets”. In: *Proceedings of the Fifth Workshop on Asian Language Resources (ALR-05) and First Symposium on Asian Language Resources Network (ALRN)*. URL: https://aclanthology.org/I05-4007.

Huang, Chu-Ren, I-Ju E. Tseng, and Dylan B.S. Tsai (2002). “Translating Lexical Semantic Relations: The First Step towards Multilingual Wordnets”. In: *COLING-02: SEMANET: Building and Using Semantic Networks*. URL: https://aclanthology.org/W02-1106.

Ide, Nancy and Catherine Macleod (2001). “The american national corpus: A standardized resource of american english”. In: *Proceedings of corpus linguistics*. Vol. 3. Citeseer, pp. 1–7.

Inkpen, Diana, Oana Frunza, and Grzegorz Kondrak (2005). “Automatic identification of cognates and false friends in French and English”. In: *Proceedings of the International Conference Recent Advances in Natural Language Processing*. Vol. 9, pp. 251–257.

Kamholz, David, Jonathan Pool, and Susan Colowick (May 2014). “PanLex: Building a Resource for Panlingual Lexical Translation”. In: *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC’14)*. Reykjavik, Iceland: European Language Resources Association (ELRA), pp. 3145–3150. URL: http://www.lrec-conf.org/proceedings/lrec2014/pdf/1029_Paper.pdf.

Kavka, Stanislav (2009). “Compounding and idiomatology”. In: *The Oxford handbook of compounding*.

Kazakov, Dimitar and Ahmad R. Shahid (Sept. 2009). “Unsupervised Construction of a Multilingual WordNet from Parallel Corpora”. In: *Proceedings of the Workshop on Natural Language Processing Methods and Corpora in Translation, Lexicography, and Language Learning*. Borovets, Bulgaria: Association for Computational Linguistics, pp. 9–12. URL: https://aclanthology.org/W09-4202.

Kerremans, Daphné, Susanne Stegmayr, and Hans-Jörg Schmid (2011). “The NeoCrawler: Identifying and retrieving neologisms from the internet and monitoring ongoing change”. In: *Current methods in historical semantics*. De Gruyter Mouton, pp. 59–96.

Kirov, Christo, Ryan Cotterell, John Sylak-Glassman, Géraldine Walther, Ekaterina Vylomova, Patrick Xia, Manaal Faruqui, Sabrina J. Mielke, Arya McCarthy, Sandra Kübler, David Yarowsky, Jason Eisner, and Mans Hulden (May 2018). “UniMorph 2.0: Universal Morphology”. In: *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*. Miyazaki, Japan: European Language Resources Association (ELRA). URL: https://aclanthology.org/L18-1293.

Kirov, Christo, John Sylak-Glassman, Roger Que, and David Yarowsky (May 2016). “Very-large Scale Parsing and Normalization of Wiktionary Morphological Paradigms”. In: *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC’16)*. Portorož, Slovenia: European Language Resources Association (ELRA), pp. 3121–3126. URL: https://aclanthology.org/L16-1498.

Klein, Guillaume, François Hernandez, Vincent Nguyen, and Jean Senellart (Oct. 2020). “The OpenNMT Neural Machine Translation Toolkit: 2020 Edition”. In: *Proceedings of the 14th Conference of the Association for Machine Translation in the Americas (Volume 1: Research Track)*. Virtual: Association for Machine Translation in the Americas, pp. 102–109. URL: https://aclanthology.org/2020.amta-research.9.

Klein, Guillaume, Yoon Kim, Yuntian Deng, Vincent Nguyen, Jean Senellart, and Alexander Rush (Mar. 2018). “OpenNMT: Neural Machine Translation Toolkit”. In: *Proceedings of the 13th Conference of the Association for Machine Translation in the Americas (Volume 1: Research Track)*. Boston, MA: Association for Machine Translation in the Americas, pp. 177–184. URL: https://aclanthology.org/W18-1817.

Klein, Guillaume, Yoon Kim, Yuntian Deng, Jean Senellart, and Alexander Rush (July 2017). “OpenNMT: Open-Source Toolkit for Neural Machine Translation”. In: *Proceedings of*

ACL 2017, System Demonstrations. Vancouver, Canada: Association for Computational Linguistics, pp. 67–72. URL: https://aclanthology.org/P17-4012.

Koehn, Philipp and Kevin Knight (Apr. 2003). "Empirical Methods for Compound Splitting". In: *10th Conference of the European Chapter of the Association for Computational Linguistics*. Budapest, Hungary: Association for Computational Linguistics. URL: https://aclanthology.org/E03-1076.

Kondrak, Grzegorz (2000). "A New Algorithm for the Alignment of Phonetic Sequences". In: *1st Meeting of the North American Chapter of the Association for Computational Linguistics*. URL: https://aclanthology.org/A00-2038.

Kondrak, Grzegorz (2001). "Identifying Cognates by Phonetic and Semantic Similarity". In: *Second Meeting of the North American Chapter of the Association for Computational Linguistics*. URL: https://aclanthology.org/N01-1014.

Kondrak, Grzegorz (2005). "N-gram similarity and distance". In: *International symposium on string processing and information retrieval*. Springer, pp. 115–126.

Kondrak, Grzegorz and Bonnie Dorr (Aug. 2004). "Identification of Confusable Drug Names: A New Approach and Evaluation Methodology". In: *COLING 2004: Proceedings of the 20th International Conference on Computational Linguistics*. Geneva, Switzerland: COLING, pp. 952–958. URL: https://aclanthology.org/C04-1137.

Kondrak, Grzegorz and Tarek Sherif (July 2006). "Evaluation of Several Phonetic Similarity Algorithms on the Task of Cognate Identification". In: *Proceedings of the Workshop* *on Linguistic Distances*. Sydney, Australia: Association for Computational Linguistics, pp. 43–50. URL: https://aclanthology.org/W06-1107.

Krotova, Irina, Sergey Aksenov, and Ekaterina Artemova (May 2020). “A Joint Approach to Compound Splitting and Idiomatic Compound Detection”. English. In: *Proceedings of the 12th Language Resources and Evaluation Conference*. Marseille, France: European Language Resources Association, pp. 4410–4417. URL: https://aclanthology.org/2020.lrec-1.543.

Kudo, Taku and John Richardson (Nov. 2018). “SentencePiece: A simple and language independent subword tokenizer and detokenizer for Neural Text Processing”. In: *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*. Brussels, Belgium: Association for Computational Linguistics, pp. 66–71. URL: https://aclanthology.org/D18-2012.

Lee, Jackson L., Lucas F.E. Ashby, M. Elizabeth Garza, Yeonju Lee-Sikka, Sean Miller, Alan Wong, Arya D. McCarthy, and Kyle Gorman (May 2020). “Massively Multilingual Pronunciation Modeling with WikiPron”. English. In: *Proceedings of the 12th Language Resources and Evaluation Conference*. Marseille, France: European Language Resources Association, pp. 4223–4228. URL: https://aclanthology.org/2020.lrec-1.521.

Leech, Geoffrey, Paul Rayson, et al. (2014). *Word frequencies in written and spoken English: Based on the British National Corpus*. Routledge.

Levenshtein, Vladimir I et al. (1966). “Binary codes capable of correcting deletions, insertions, and reversals”. In: *Soviet physics doklady*. Vol. 10. 8. Soviet Union, pp. 707–710.

REFERENCES

Lewis, Dylan, Winston Wu, Arya D. McCarthy, and David Yarowsky (Dec. 2020). “Neural Transduction for Multilingual Lexical Translation”. In: *Proceedings of the 28th International Conference on Computational Linguistics*. Barcelona, Spain (Online): International Committee on Computational Linguistics, pp. 4373–4384. URL: https://aclanthology.org/2020.coling-main.387.

Liebeck, Matthias and Stefan Conrad (July 2015). “IWNLP: Inverse Wiktionary for Natural Language Processing”. In: *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*. Beijing, China: Association for Computational Linguistics, pp. 414–418. URL: https://aclanthology.org/P15-2068.

Lieber, Rochelle and Pavol Stekauer (2011). “The Oxford handbook of compounding”. In.

List, Johann-Mattis, Michael Cysouw, and Robert Forkel (May 2016). “Concepticon: A Resource for the Linking of Concept Lists”. In: *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC’16)*. Portorož, Slovenia: European Language Resources Association (ELRA), pp. 2393–2400. URL: https://aclanthology.org/L16-1379.

Liu, Chang, Daniel Dahlmeier, and Hwee Tou Ng (July 2010). “TESLA: Translation Evaluation of Sentences with Linear-Programming-Based Analysis”. In: *Proceedings of the Joint Fifth Workshop on Statistical Machine Translation and MetricsMATR*. Uppsala, Sweden: Association for Computational Linguistics, pp. 354–359. URL: https://aclanthology.org/W10-1754.

Mackay, Wesley and Grzegorz Kondrak (June 2005). "Computing Word Similarity and Identifying Cognates with Pair Hidden Markov Models". In: *Proceedings of the Ninth Conference on Computational Natural Language Learning (CoNLL-2005)*. Ann Arbor, Michigan: Association for Computational Linguistics, pp. 40–47. URL: https://aclanthology.org/W05-0606.

Mann, Gideon S. and David Yarowsky (2001). "Multipath Translation Lexicon Induction via Bridge Languages". In: *Second Meeting of the North American Chapter of the Association for Computational Linguistics*. URL: https://aclanthology.org/N01-1020.

Marchisio, Kelly, Kevin Duh, and Philipp Koehn (Nov. 2020). "When Does Unsupervised Machine Translation Work?" In: *Proceedings of the Fifth Conference on Machine Translation*. Online: Association for Computational Linguistics, pp. 571–583. URL: https://aclanthology.org/2020.wmt-1.68.

Marchisio, Kelly, Philipp Koehn, and Conghao Xiong (Aug. 2021). "An Alignment-Based Approach to Semi-Supervised Bilingual Lexicon Induction with Small Parallel Corpora". In: *Proceedings of the 18th Biennial Machine Translation Summit (Volume 1: Research Track)*. Virtual: Association for Machine Translation in the Americas, pp. 293–304. URL: https://aclanthology.org/2021.mtsummit-research.24.

Matthews, Austin, Eva Schlinger, Alon Lavie, and Chris Dyer (Aug. 2016). "Synthesizing Compound Words for Machine Translation". In: *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Berlin, Germany: Association for Computational Linguistics, pp. 1085–1094. URL: https://aclanthology.org/P16-1103.

McBride-Chang, Catherine, Ellen Bialystok, Karen KY Chong, and Yanping Li (2004). "Levels of phonological awareness in three cultures". In: *Journal of experimental child psychology* 89.2, pp. 93–111.

McCarthy, Arya D., Christo Kirov, Matteo Grella, Amrit Nidhi, Patrick Xia, Kyle Gorman, Ekaterina Vylomova, Sabrina J. Mielke, Garrett Nicolai, Miikka Silfverberg, Timofey Arkhangelskiy, Nataly Krizhanovsky, Andrew Krizhanovsky, Elena Klyachko, Alexey Sorokin, John Mansfield, Valts Ernštreits, Yuval Pinter, Cassandra L. Jacobs, Ryan Cotterell, Mans Hulden, and David Yarowsky (May 2020). "UniMorph 3.0: Universal Morphology". English. In: *Proceedings of the 12th Language Resources and Evaluation Conference*. Marseille, France: European Language Resources Association, pp. 3922–3931. URL: https://aclanthology.org/2020.lrec-1.483.

McCarthy, Arya D., Rachel Wicks, Dylan Lewis, Aaron Mueller, Winston Wu, Oliver Adams, Garrett Nicolai, Matt Post, and David Yarowsky (May 2020). "The Johns Hopkins University Bible Corpus: 1600+ Tongues for Typological Exploration". English. In: *Proceedings of the 12th Language Resources and Evaluation Conference*. Marseille, France: European Language Resources Association, pp. 2884–2892. URL: https://aclanthology.org/2020.lrec-1.352.

McFate, Clifton and Kenneth Forbus (June 2011). "NULEX: An Open-License Broad Coverage Lexicon". In: *Proceedings of the 49th Annual Meeting of the Association for Computa-*

tional Linguistics: Human Language Technologies. Portland, Oregon, USA: Association for Computational Linguistics, pp. 363–367. URL: https://aclanthology.org/P11-2063.

Melamed, I. Dan (1999). "Bitext Maps and Alignment via Pattern Recognition". In: *Computational Linguistics* 25.1, pp. 107–130. URL: https://aclanthology.org/J99-1003.

Melo, Gerard de (May 2014). "Etymological Wordnet: Tracing The History of Words". In: *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*. Reykjavik, Iceland: European Language Resources Association (ELRA), pp. 1148–1154. URL: http://www.lrec-conf.org/proceedings/lrec2014/pdf/1083_Paper.pdf.

Meloni, Carlo, Shauli Ravfogel, and Yoav Goldberg (June 2021). "Ab Antiquo: Neural Proto-language Reconstruction". In: *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Online: Association for Computational Linguistics, pp. 4460–4473. URL: https://aclanthology.org/2021.naacl-main.353.

Michel, Jean-Baptiste, Yuan Kui Shen, Aviva Presser Aiden, Adrian Veres, Matthew K Gray, Google Books Team, Joseph P Pickett, Dale Hoiberg, Dan Clancy, Peter Norvig, et al. (2011). "Quantitative analysis of culture using millions of digitized books". In: *science* 331.6014, pp. 176–182.

Müller, Karin (June 2006). "Improving Syllabification Models with Phonotactic Knowledge". In: *Proceedings of the Eighth Meeting of the ACL Special Interest Group on Com-*

putational Phonology and Morphology at HLT-NAACL 2006. New York City, USA: Association for Computational Linguistics, pp. 11–20. URL: https://aclanthology.org/W06-3202.

Mulloni, Andrea (June 2007). “Automatic Prediction of Cognate Orthography Using Support Vector Machines”. In: *Proceedings of the ACL 2007 Student Research Workshop*. Prague, Czech Republic: Association for Computational Linguistics, pp. 25–30. URL: https://aclanthology.org/P07-3005.

Nastase, Vivi and Carlo Strapparava (Aug. 2013). “Bridging Languages through Etymology: The case of cross language text categorization”. In: *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Sofia, Bulgaria: Association for Computational Linguistics, pp. 651–659. URL: https://aclanthology.org/P13-1064.

Nastase, Vivi and Carlo Strapparava (2015). “knoWitiary: A Machine Readable Incarnation of Wiktionary.” In: *Int. J. Comput. Linguistics Appl.* 6.2, pp. 61–82.

Nastase, Vivi, Michael Strube, Benjamin Boerschinger, Caecilia Zirn, and Anas Elghafari (May 2010). “WikiNet: A Very Large Scale Multi-Lingual Concept Network”. In: *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC’10)*. Valletta, Malta: European Language Resources Association (ELRA). URL: http://www.lrec-conf.org/proceedings/lrec2010/pdf/615_Paper.pdf.

Navarro, Emmanuel, Franck Sajous, Bruno Gaume, Laurent Prévot, ShuKai Hsieh, Ivy Kuo, Pierre Magistry, and Chu-Ren Huang (Aug. 2009). “Wiktionary for Natural Language

Processing: Methodology and Limitations". In: *Proceedings of the 2009 Workshop on The People's Web Meets NLP: Collaboratively Constructed Semantic Resources (People's Web)*. Suntec, Singapore: Association for Computational Linguistics, pp. 19–27. URL: https://aclanthology.org/W09-3303.

Ngo, Thi-Vinh, Thanh-Le Ha, Phuong-Thai Nguyen, and Le-Minh Nguyen (Nov. 2019). "Overcoming the Rare Word Problem for low-resource language pairs in Neural Machine Translation". In: *Proceedings of the 6th Workshop on Asian Translation*. Hong Kong, China: Association for Computational Linguistics, pp. 207–214. URL: https://aclanthology.org/D19-5228.

Nichols, Johanna and Tandy Warnow (2008). "Tutorial on computational linguistic phylogeny". In: *Language and Linguistics Compass* 2.5, pp. 760–820.

Nicolai, Garrett, Dylan Lewis, Arya D. McCarthy, Aaron Mueller, Winston Wu, and David Yarowsky (May 2020). "Fine-grained Morphosyntactic Analysis and Generation Tools for More Than One Thousand Languages". English. In: *Proceedings of the 12th Language Resources and Evaluation Conference*. Marseille, France: European Language Resources Association, pp. 3963–3972. URL: https://aclanthology.org/2020.lrec-1.488.

Nicolai, Garrett, Lei Yao, and Grzegorz Kondrak (Aug. 2016). "Morphological Segmentation Can Improve Syllabification". In: *Proceedings of the 14th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology*. Berlin, Germany: Association for Computational Linguistics, pp. 99–103. URL: https://aclanthology.org/W16-2016.

Nien, Tzu-yi, Tsun Ku, Chung-chi Huang, Mei-hua Chen, and Jason S. Chang (Dec. 2009). "Extending Bilingual WordNet via Hierarchical Word Translation Classification". In: *Proceedings of the 23rd Pacific Asia Conference on Language, Information and Computation, Volume 1*. Hong Kong: City University of Hong Kong, pp. 375–384. URL: https://aclanthology.org/Y09-1040.

Nivre, Joakim, Marie-Catherine de Marneffe, Filip Ginter, Yoav Goldberg, Jan Hajič, Christopher D. Manning, Ryan McDonald, Slav Petrov, Sampo Pyysalo, Natalia Silveira, Reut Tsarfaty, and Daniel Zeman (May 2016). "Universal Dependencies v1: A Multilingual Treebank Collection". In: *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*. Portorož, Slovenia: European Language Resources Association (ELRA), pp. 1659–1666. URL: https://aclanthology.org/L16-1262.

Nivre, Joakim, Marie-Catherine de Marneffe, Filip Ginter, Jan Hajič, Christopher D. Manning, Sampo Pyysalo, Sebastian Schuster, Francis Tyers, and Daniel Zeman (May 2020). "Universal Dependencies v2: An Evergrowing Multilingual Treebank Collection". English. In: *Proceedings of the 12th Language Resources and Evaluation Conference*. Marseille, France: European Language Resources Association, pp. 4034–4043. URL: https://aclanthology.org/2020.lrec-1.497.

Nordhoff, Sebastian and Harald Hammarström (2011). "Glottolog/Langdoc: Defining dialects, languages, and language families as collections of resources". In: *First International Workshop on Linked Science 2011-In conjunction with the International Semantic Web Conference (ISWC 2011)*.

Nouri, Javad and Roman Yangarber (Aug. 2016). “From alignment of etymological data to phylogenetic inference via population genetics”. In: *Proceedings of the 7th Workshop on Cognitive Aspects of Computational Language Learning*. Berlin: Association for Computational Linguistics, pp. 27–37. URL: https://aclanthology.org/W16-1905.

Och, Franz Josef and Hermann Ney (Oct. 2000). “Improved Statistical Alignment Models”. In: *Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics*. Hong Kong: Association for Computational Linguistics, pp. 440–447. URL: https://aclanthology.org/P00-1056.

Ogden, Charles Kay (1932). *The ABC of basic English (in Basic)*. Vol. 43. K. Paul, Trench, Trubner & Company Limited.

Orel, Vladimir (1998). *Albanian etymological dictionary*. Brill.

Partridge, Eric (2006). *Origins: A short etymological dictionary of modern English*. Routledge.

Petersen, Alexander M, Joel Tenenbaum, Shlomo Havlin, and H Eugene Stanley (2012). “Statistical laws governing fluctuations in word use from word birth to word death”. In: *Scientific reports* 2.1, pp. 1–9.

Plaza, Monique and Henri Cohen (2007). “The contribution of phonological awareness and visual attention in early reading and spelling”. In: *Dyslexia* 13.1, pp. 67–76.

Post, Matt (Oct. 2018). “A Call for Clarity in Reporting BLEU Scores”. In: *Proceedings of the Third Conference on Machine Translation: Research Papers*. Brussels, Belgium: Association for Computational Linguistics, pp. 186–191. URL: https://aclanthology.org/W18-6319.

Pyysalo, Jouna (May 2017). “Proto-Indo-European Lexicon: The Generative Etymological Dictionary of Indo-European Languages”. In: *Proceedings of the 21st Nordic Conference on Computational Linguistics*. Gothenburg, Sweden: Association for Computational Linguistics, pp. 259–262. URL: https://aclanthology.org/W17-0234.

Rackow, Ulrike, Ido Dagan, and Ulrike Schwall (1992). “Automatic Translation of Noun Compounds”. In: *COLING 1992 Volume 4: The 14th International Conference on Computational Linguistics*. URL: https://aclanthology.org/C92-4201.

Rama, Taraka (May 2015). “Automatic cognate identification with gap-weighted string subsequences.” In: *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Denver, Colorado: Association for Computational Linguistics, pp. 1227–1231. URL: https://aclanthology.org/N15-1130.

Ravi, Sujith and Kevin Knight (June 2011). “Deciphering Foreign Language”. In: *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*. Portland, Oregon, USA: Association for Computational Linguistics, pp. 12–21. URL: https://aclanthology.org/P11-1002.

Ryskina, Maria, Ella Rabinovich, Taylor Berg-Kirkpatrick, David Mortensen, and Yulia Tsvetkov (Jan. 2020). “Where New Words Are Born: Distributional Semantic Analysis of Neologisms and Their Semantic Neighborhoods”. In: *Proceedings of the Society for*

Computation in Linguistics 2020. New York, New York: Association for Computational Linguistics, pp. 367–376. URL: https://aclanthology.org/2020.scil-1.43.

Sagot, Benoît (2017). "Extracting an etymological database from wiktionary". In: *Electronic Lexicography in the 21st century (eLex 2017)*, pp. 716–728.

Sajous, Franck, Basilio Calderone, and Nabil Hathout (May 2020). "ENGLAWI: From Human- to Machine-Readable Wiktionary". English. In: *Proceedings of the 12th Language Resources and Evaluation Conference*. Marseille, France: European Language Resources Association, pp. 3016–3026. URL: https://aclanthology.org/2020.lrec-1.369.

Schafer, Charles and David Yarowsky (2002). "Inducing Translation Lexicons via Diverse Similarity Measures and Bridge Languages". In: *COLING-02: The 6th Conference on Natural Language Learning 2002 (CoNLL-2002)*. URL: https://aclanthology.org/W02-2026.

Scherrer, Yves and Benoît Sagot (May 2014). "A language-independent and fully unsupervised approach to lexicon induction and part-of-speech tagging for closely related languages". In: *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*. Reykjavik, Iceland: European Language Resources Association (ELRA), pp. 502–508. URL: http://www.lrec-conf.org/proceedings/lrec2014/pdf/797_Paper.pdf.

Schlippe, Tim, Sebastian Ochs, and Tanja Schultz (2010). "Wiktionary as a source for automatic pronunciation extraction". In: *Eleventh Annual Conference of the International Speech Communication Association*.

Schuessler, A (2007). “ABC dictionary of Old Chinese”. In: *University of Hawai’i Press: Honolulu*.

See, Abigail, Peter J. Liu, and Christopher D. Manning (July 2017). “Get To The Point: Summarization with Pointer-Generator Networks”. In: *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Vancouver, Canada: Association for Computational Linguistics, pp. 1073–1083. URL: https://aclanthology.org/P17-1099.

Sennrich, Rico, Barry Haddow, and Alexandra Birch (Aug. 2016). “Neural Machine Translation of Rare Words with Subword Units”. In: *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Berlin, Germany: Association for Computational Linguistics, pp. 1715–1725. URL: https://aclanthology.org/P16-1162.

Sérasset, Gilles (2015). “DBnary: Wiktionary as a Lemon-based multilingual lexical resource in RDF”. In: *Semantic Web* 6.4, pp. 355–361.

Shqerra, Nereida and Endri Shqerra (2014). “The role of derivation and compounding in the process of English language acquisition”. In: *Journal of Educational and Social Research* 4.2, p. 117.

Simard, Michel, George F Foster, and Pierre Isabelle (1992). “Using cognates to align sentences in bilingual corpora”. In: *Proceedings of the Fourth Conference on Theoretical and Methodological Issues in Machine Translation of Natural Languages*.

Smith, Jason R., Herve Saint-Amand, Magdalena Plamada, Philipp Koehn, Chris Callison-Burch, and Adam Lopez (Aug. 2013). "Dirt Cheap Web-Scale Parallel Text from the Common Crawl". In: *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Sofia, Bulgaria: Association for Computational Linguistics, pp. 1374–1383. URL: https://aclanthology.org/P13-1135.

Starostin, Sergei A, Anna Dybo, Oleg Mudrak, and Ilya Gruntov (2003). *Etymological dictionary of the Altaic languages*. Vol. 3. Brill Leiden.

Štekauer, Pavol (2009). "Meaning predictability of novel context-free compounds". In: *The Oxford handbook of compounding*.

Stymne, Sara and Nicola Cancedda (July 2011). "Productive Generation of Compound Words in Statistical Machine Translation". In: *Proceedings of the Sixth Workshop on Statistical Machine Translation*. Edinburgh, Scotland: Association for Computational Linguistics, pp. 250–260. URL: https://aclanthology.org/W11-2129.

Stymne, Sara, Nicola Cancedda, and Lars Ahrenberg (2013). "Generation of Compound Words in Statistical Machine Translation into Compounding Languages". In: *Computational Linguistics* 39.4, pp. 1067–1108. URL: https://aclanthology.org/J13-4009.

Swadesh, Morris (1952). "Lexico-statistic dating of prehistoric ethnic contacts: with special reference to North American Indians and Eskimos". In: *Proceedings of the American philosophical society* 96.4, pp. 452–463.

Swadesh, Morris (1955). "Towards greater accuracy in lexicostatistic dating". In: *International journal of American linguistics* 21.2, pp. 121–137.

Swadesh, Morris (2017). *The origin and diversification of language*. Routledge.

Tanaka, Takaaki and Timothy Baldwin (July 2003). "Noun-Noun Compound Machine Translation A Feasibility Study on Shallow Processing". In: *Proceedings of the ACL 2003 Workshop on Multiword Expressions: Analysis, Acquisition and Treatment*. Sapporo, Japan: Association for Computational Linguistics, pp. 17–24. URL: https://aclanthology.org/W03-1803.

Trask, Robert Lawrence (2000). *The dictionary of historical and comparative linguistics*. Psychology Press.

Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin (2017). "Attention is all you need". In: *Advances in neural information processing systems*, pp. 5998–6008.

Verhoeven, Ben, Walter Daelemans, Menno van Zaanen, and Gerhard van Huyssteen, eds. (Aug. 2014). *Proceedings of the First Workshop on Computational Approaches to Compound Analysis (ComAComA 2014)*. Dublin, Ireland: Association for Computational Linguistics and Dublin City University. URL: https://aclanthology.org/W14-5700.

Waxmonsky, Sonjia and Sravana Reddy (June 2012). "G2P Conversion of Proper Names Using Word Origin Information". In: *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Montréal, Canada: Association for Computational Linguistics, pp. 367–371. URL: https://aclanthology.org/N12-1039.

Weerasinghe, Ruvan, Asanka Wasala, and Kumudu Gamage (2005). “A Rule Based Syllabification Algorithm for Sinhala”. In: *Second International Joint Conference on Natural Language Processing: Full Papers*. URL: https://aclanthology.org/I05-1039.

West, Michael (1953). “A General Service List of English Words1953”. In: *West A General Service List of English Words1953*.

Wu, Winston, Kevin Duh, and David Yarowsky (Aug. 2021). “Sequence Models for Computational Etymology of Borrowings”. In: *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*. Online: Association for Computational Linguistics, pp. 4032–4037. URL: https://aclanthology.org/2021.findings-acl.353.

Wu, Winston, Garrett Nicolai, and David Yarowsky (May 2020). “Multilingual Dictionary Based Construction of Core Vocabulary”. English. In: *Proceedings of the 12th Language Resources and Evaluation Conference*. Marseille, France: European Language Resources Association, pp. 4211–4217. URL: https://aclanthology.org/2020.lrec-1.519.

Wu, Winston, Nidhi Vyas, and David Yarowsky (May 2018). “Creating a Translation Matrix of the Bible’s Names Across 591 Languages”. In: *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*. Miyazaki, Japan: European Language Resources Association (ELRA). URL: https://aclanthology.org/L18-1263.

Wu, Winston and David Yarowsky (May 2018a). “A Comparative Study of Extremely Low-Resource Transliteration of the World’s Languages”. In: *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*. Miyazaki, Japan: European Language Resources Association (ELRA). URL: https://aclanthology.org/L18-1150.

Wu, Winston and David Yarowsky (May 2018b). “Creating Large-Scale Multilingual Cognate Tables”. In: *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*. Miyazaki, Japan: European Language Resources Association (ELRA). URL: https://aclanthology.org/L18-1538.

Wu, Winston and David Yarowsky (May 2018c). “Massively Translingual Compound Analysis and Translation Discovery”. In: *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*. Miyazaki, Japan: European Language Resources Association (ELRA). URL: https://aclanthology.org/L18-1612.

Wu, Winston and David Yarowsky (May 2020a). “Computational Etymology and Word Emergence”. English. In: *Proceedings of the 12th Language Resources and Evaluation Conference*. Marseille, France: European Language Resources Association, pp. 3252–3259. URL: https://aclanthology.org/2020.lrec-1.397.

Wu, Winston and David Yarowsky (Dec. 2020b). “Wiktionary Normalization of Translations and Morphological Information”. In: *Proceedings of the 28th International Conference on Computational Linguistics*. Barcelona, Spain (Online): International Committee on Computational Linguistics, pp. 4683–4692. URL: https://aclanthology.org/2020.coling-main.413.

Wu, Winston and David Yarowsky (Sept. 2021). “On Pronunciations in Wiktionary: Extraction and Experiments on Multilingual Syllabification and Stress Prediction”. In: *Proceedings of the 14th Workshop on Building and Using Comparable Corpora (BUCC 2021)*. Online (Virtual Mode): INCOMA Ltd., pp. 68–74. URL: https://aclanthology.org/2021.bucc-1.9.

Yadav, Satya P (2007). “The wholeness in suffix-omics,-omes, and the word om”. In: *Journal of biomolecular techniques: JBT* 18.5, p. 277.

Yang, Guang-rong (2004). “The Prospect of the Etymological Study in the 21st Century: Computational Etymology”. In: *Journal of Shanxi University (Philosophy and Social Sciences)*.

Ziering, Patrick and Lonneke van der Plas (June 2016). “Towards Unsupervised and Language-independent Compound Splitting using Inflectional Morphological Transformations”. In: *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. San Diego, California: Association for Computational Linguistics, pp. 644–653. URL: https://aclanthology.org/N16-1078.