MASTER'S THESIS
VERSION 1.3
Research Collaboration with SIL International

GUIDE: CREATING SEMANTIC DOMAIN DICTIONARIES FOR LOW-RESOURCE LANGUAGES

GUIDE: Erstellung von Wörterbüchern für Semantische Domänen in ressourcenarmen Sprachen

JONATHAN JANETZKI
jonathan.janetzki@student.hpi.de

SUPERVISORS
Prof. Dr. Gerard de Melo
Prof. Dr. Robert Hirschfeld
Joshua Nemecek
Dr. Daniel Whitenack

RESEARCH GROUP
Artificial Intelligence and Intelligent Systems
Hasso Plattner Institute
University of Potsdam

Potsdam, 24th October 2023

Jonathan Janetzki:
GUIDE: Creating Semantic Domain Dictionaries for Low-Resource Languages,

GUIDE: Erstellung von Wörterbüchern für Semantische Domänen
in ressourcenarmen Sprachen

SUPERVISORS
Prof. Dr. Gerard de Melo
Prof. Dr. Robert Hirschfeld
Joshua Nemecek
Dr. Daniel Whitenack

SUBMITTED ON
24th October 2023
(revised on November 3, 2023)

It is true that there are all kinds of languages in the world.
And they all have meaning.
But if I don't understand what someone is saying,
I am a stranger to the person speaking.
And that person is a stranger to me.

— 1. Corinthians 14:10-11, The Bible (NIRV)

ABSTRACT

Over 7,000 of the world's 7,168 living languages are still low-resourced. This thesis aims to narrow the language documentation gap by creating multiparallel dictionaries, clustered by SIL's semantic domains. This task is new for machine learning, has previously been done manually by native speakers. We propose GUIDE, a language-agnostic tool that uses a GNN to create and populate semantic domain dictionaries, using seed dictionaries and Bible translations as a parallel text corpus. Our work sets a new benchmark, achieving an exemplary average precision of 60% in eight languages without training data and predicting an average of 2,400 dictionary entries. GUIDE demonstrates the potential to support NLP applications for low-resource languages, particularly in machine translation. We share the code, model, multilingual test set, and new dictionaries with the research community.[1]

ZUSAMMENFASSUNG

Mehr als 7000 der 7168 lebenden Sprachen auf der Erde sind noch ressourcenarme Sprachen. Ziel dieser Arbeit ist es, die Lücke in der Sprachdokumentation durch die Erstellung multiparalleler Wörterbücher zu verkleinern, die nach den semantischen Domänen von SIL gruppiert sind. Diese Aufgabe ist neu für maschinelles Lernen, da sie bisher manuell von Muttersprachlern durchgeführt wurde. Wir präsentieren GUIDE, ein sprachunabhängiges Werkzeug, das ein GNN verwendet, um Wörterbücher für semantische Domänen zu erstellen und zu füllen, wobei Quellwörterbücher und Bibelübersetzungen als paralleler Textkorpus verwendet werden. Unsere Arbeit setzt einen neuen Standard, indem sie eine beispielhafte Genauigkeit von 60% in acht Sprachen ohne Trainingsdaten erreicht und dabei durschnittlich 2400 Wörterbucheinträge generiert. GUIDE zeigt Potential zur Unterstützung von NLP-Anwendungen für ressourcenarme Sprachen, insbesondere der maschinellen Übersetzung. Wir stellen den Code, das Modell, den mehrsprachigen Testdatensatz und die neuen Wörterbücher der Forschungsgemeinschaft zur Verfügung.[1]

[1]GitHub repository: `https://github.com/janetzki/GUIDE`

ACKNOWLEDGMENTS

I want like to express my sincere gratitude to everyone who helped me during my studies and especially while I was working on this thesis.

First of all, a big thanks to Joshua Nemecek for his excellent support, proofreading, detailed feedback, and smart solutions when I faced unpredictable challenges. We had many insightful discussions, and he provided the necessary data and setup code. And, of course, he also helped me find native speakers for last-minute evaluations.

A special word of thanks goes to Prof. Dr. Gerard de Melo for his consistent guidance and invaluable feedback during the project. He took much time to discuss the progress and drew attention to the priorities when necessary. I also thank Prof. Dr. Robert Hirschfeld for making this project possible and supervising it.

I thank Dr. Daniel Whitenack, who initiated this project, for all his extraordinarily helpful ideas and advice. He connected me with relevant experts and gave thoughtful feedback on my proposal draft.

Thanks to Maximilian Schall for supporting me with the necessary hardware and providing valuable hints for the proposal. I thank Sophie Bodien and the Hasso Plattner Institute. They quickly helped to reimburse the costs of this project. Thanks to Jonathan Robie for his helpful explanations regarding MARBLE's semantic domains. And I cannot forget to thank Mark Woodward and Delali Idrissiou for sharing useful data with me.

I am grateful to the entire Artificial Intelligence and Intelligent Systems research group. They assisted me with access to the chair's computing resources and helped me discuss and evaluate the approach. Thanks to everyone who completed the questionnaire.

Lukas Ehrig and Jonathan Schneider deserve a big thank you for finetuning this thesis and giving kind and thoughtful suggestions.

A big thank you to my friends and colleagues for the good times and the strong emotional support. My family always helped me, especially when I was not feeling well.

Thank you all!

CONTENTS

ACRONYMS

AER	*Alignment Error Rate*
AWESoME	*Aligning Word Embedding Spaces of Multilingual Encoders*
BLEU	*BiLingual Evaluation Understudy*
BPE	*Byte Pair Encoding*
CNN	*Convolutional Neural Network*
COMET	*Crosslingual Optimized Metric for Evaluation of Translation*
COVID-19	*Coronavirus disease 2019*
FLEx	*FieldWorks Language Explorer*
FLORES	*Facebook Low Resource*
FOCUS	*Fast Overlapping Token Combinations Using Sparsemax*
GAT	*Graph Attention Network*
GCN	*Graph Convolutional Network*
GDFA	*grow-diag-final-and*
GLP	*Graph Label Propagation*
GNN	*Graph Neural Network*
GPT-4	*Generative Pre-trained Transformer 4*
GPU	*Graphics Processing Unit*
GUIDE	*Graph-based Unified Indigenous Dictionary Engine*
ID	*identifier*
ISO	*International Organization for Standardization*
LaBSE	*Language-agnostic BERT Sentence Embedding*
LLM	*Large Language Model*
MAG	*Multilingual Alignment Graph*
MCMC	*Markov Chain Monte Carlo*
METEOR	*Metric for Evaluation of Translation with Explicit Ordering*
MWT	*Multi-Word Token*
NFKD	*Normalization Form Compatibly Decomposition*
NLLB	*No Language Left Behind*
NLP	*Natural Language Processing*
NMT	*Neural Machine Translation*
POS	*Part-of-Speech*
QE	*Quality Estimation*

RAM	*Random-Access Memory*
SDQ	*Semantic Domain Question*
SIL	*Summer Insitute of Linguistics (former name)*
SONAR	*Sentence-Level Multimodal and Language-Agnostic Representations*
URL	*Uniform Resource Locator*
UWN	*Universal Wordnet*
VRAM	*Video Random-Access Memory*
WALS	*World Atlas of Language Structures*
WandB	*Weights and Biases*
WSD	*Word-Sense Disambiguation*
XLM-R	*Cross-lingual Language Model - RoBERTa*

1 THE GLOBAL LANGUAGE DOCUMENTATION GAP

"The limits of my language mean the limits of my world."

— *Ludwig Wittgenstein* [75]

1.1 MOTIVATION

There are 7,168 languages spoken on Earth [26]. Between 1950 and 2010, 230 languages disappeared [91]. Every 14 days, a language vanishes with its last speaker passing away. It is estimated that nearly 50% of languages will disappear by the end of the 21st century because communities give up their native languages in favor of English, Spanish, or Mandarin [80].

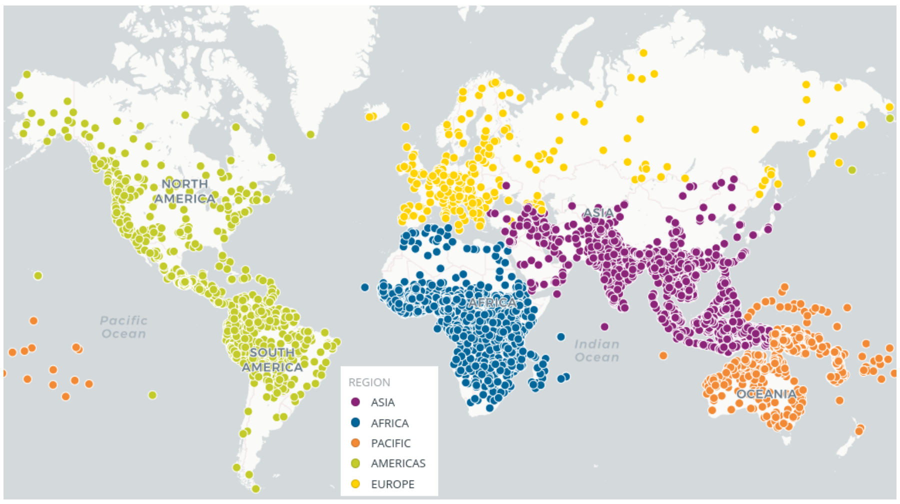


Figure 1.1: All living 7,168 languages, colored by continent (taken from the Ethnologue[1] [26]): The dots show their approximate location of origin.

Figure 1.1 shows the global language distribution. All but three countries have less than five official languages [96], and the remaining languages are often poorly documented. Papua New Guinea, for example, has three official languages and more than 800 living languages in total, making it the country with the most languages in the world. That is more than twice the number of languages spoken in Europe. For comparison, Germany has 19 [26] living languages.

[1] Ethnologue: https://www.ethnologue.com/insights/how-many-languages/ (visited on 19/10/2023)

Public machine translation models currently support up to 498 [52] languages. However, these are only 7% of all living languages. The remaining languages are low-resourced, which means that only a strictly limited amount of textual data is available. Due to this language documentation gap, these languages are currently challenging for machine translation models and *Natural Language Processing* (NLP) models in general.

Creating dictionaries is the first step towards documenting languages, and it is also one of the most effective ways to preserve languages and cultures [1]. A successful approach to creating dictionaries for low-resource languages is *rapid word collection* [12]: A team of linguists travels to about 60 indigenous people for two or three weeks and guides them through a questionnaire. Each question belongs to a *semantic domain* [67], which groups words with similar meanings. Since this process involves traveling, it is expensive and sometimes even impossible (e.g., due to COVID-19). This work aims to increase the speed and reduce the cost of creating dictionaries in low-resource languages.

The main idea of this thesis is to automatically create semantic domain dictionaries for low-resource languages and fill in missing entries using the Bible, a multilingual parallel text corpus, and existing semantic domain dictionaries. Figure 1.2 compares the number of languages that can be addressed using the Bible or other resources.

The basic process is to

1. align words between different languages, which are likely to have the same semantic domain,
2. aggregate (i.e., sum up) the alignments by word, and
3. analyze the graph with a *Graph Neural Network* (GNN) to predict which word belongs to which *Semantic Domain Question* (SDQ). Figure 1.3 shows an example result of this task.

1.2 RESEARCH QUESTIONS

Based on the above-mentioned problems, our work aims to investigate the following research questions.

RQ1. How can we use a GNN to create entries for a dictionary organized by SDQ (also called word-SDQ link in the following), given such dictionaries in other languages?

[2] *FieldWorks Language Explorer* (FLEx) is a tool to document and analyze languages and the most common tool used with Moe's [67] semantic domains. The FieldWorks page describes FLEx: `https://software.sil.org/fieldworks/flex` (visited on 16/10/2023).

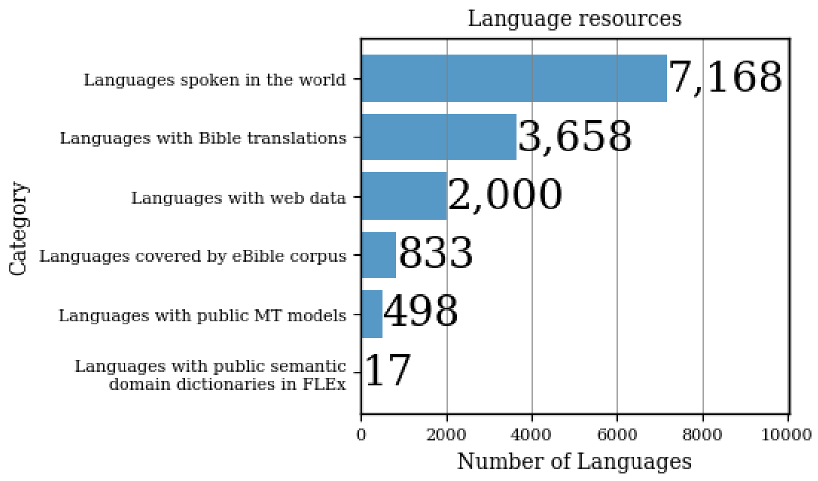
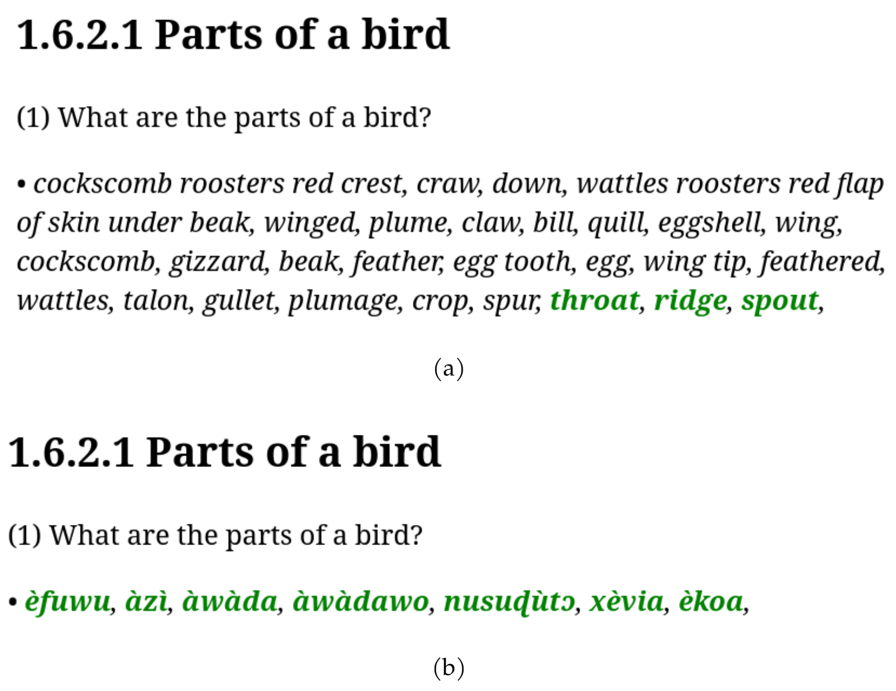


Figure 1.2: The languages covered in the eBible corpus (see Subsection 2.2.3) are the scope of target languages addressed in this thesis: Technically, the presented approach works for more than 3,600 languages that have (partial) Bible translations [98], covering 51% of all spoken languages. These are nearly twice as many languages as web-based approaches could cover. 2,000 languages are estimated to have web data, which has been discovered in the Crúbadán project [6, 82].

RQ2. How does the number of known entries in the source dictionaries during the GNN's training affect the quality of the generated entries in the target dictionary?

RQ3. How can our GNN models become interpretable? Users should be able to understand how the input data affects the model's predictions.

We hypothesize that a GNN provides an effective model architecture for RQ1 because it can exploit the graph structure of the input data [83]. Therefore, GNNs can be more accurate than sequence-to-sequence models in settings with strictly limited training data, which is especially helpful when working with low-resource languages, as Schlichtkrull and Søgaard [84] and Wang et al. [94] did. For RQ2, we hypothesize that the quality of the results increases and saturates as the number of already known words in the target dictionaries increases. Regarding RQ3, there exist dedicated *explainability* methods [72] for complex GNNs. Nevertheless, we try to build a simple GNN that makes intuitive predictions by initializing its weights according to human intuition.


1.6.2.1 Parts of a bird

(1) What are the parts of a bird?

• *cockscomb roosters red crest, craw, down, wattles roosters red flap of skin under beak, winged, plume, claw, bill, quill, eggshell, wing, cockscomb, gizzard, beak, feather, egg tooth, egg, wing tip, feathered, wattles, talon, gullet, plumage, crop, spur,* ***throat***, ***ridge***, ***spout***,

(a)

1.6.2.1 Parts of a bird

(1) What are the parts of a bird?

• ***èfuwu***, ***àzì***, ***àwàda***, ***àwàdawo***, ***nusuɖùtɔ***, ***xèvia***, ***èkoa***,

(b)

Figure 1.3: Existing (black) semantic domain dictionary entries in FLEx[2] and newly predicted (green) ones: (a) shows three entries that our model added to the English dictionary and (b) shows seven entries for the same SDQ in the newly created Mina-Gen dictionary.

1.3 CONTRIBUTIONS

Our thesis makes significant contributions, summarized as follows.

- NEW BENCHMARK Our work establishes a new benchmark for linking words to their SDQs. To the best of our knowledge, we propose the first automated approach to address this task.

- DICTIONARY CREATION We propose the language-agnostic tool *Graph-based Unified Indigenous Dictionary Engine* (GUIDE), which links words in 20 languages and seven language families to their SDQs. It achieves state-of-the-art performance and has an average precision of 65% in twenty languages.

- MULTIPARALLELITY The created dictionaries are not monolingual or bilingual, but multiparallel like a paper stack. By using SIL's language-agnostic semantic domains as a scaffold, we overcome the limitations of 1:1 word-to-word translations.

- LANGUAGE FLEXIBILITY To build a dictionary for a language, GUIDE requires only a Bible translation in that language, which (partially) exists for 3,658 languages and is accessible in a verse-aligned format for at least 833 [102] languages. GUIDE can also be adapted to build dictionaries from any parallel text, such as Bloom Books [55].

- RICHER DICTIONARIES We have predicted 32,000 new word-SDQ links for twelve languages with existing dictionaries that can enrich FLEx[2] (if verified by a native speaker). Three of these languages are low-resourced.
- NEW DICTIONARIES We have predicted 19,000 word-SDQ links for eight languages without an existing or virtually empty dictionary (to be verified by a native speaker). Seven of these languages are low-resourced.
- INTERPRETABILITY We demonstrate that GUIDE's predictions are often intuitively understandable. Users can trace back GUIDE's decision process and determine if an error lies in the input data, the preprocessing pipeline, or the GNN.
- OPEN-SOURCE We have made our code, the pretrained model, and the predicted words publicly available[3].
- FUTURE DIRECTIONS We analyze the limitations as well as possible improvements and propose six possible future research directions opened up by GUIDE.

1.4 STRUCTURE

In this thesis, we focus on dictionary creation for low-resource languages with GNNs as described in Section 1.1. Before diving deeper into our approach, we introduce crucial terminology and give some background information in Chapter 2 on low-resource languages, parallel datasets, semantic domains, word alignment, and GNNs. Moreover, the chapter introduces existing resources and tools and highlights their strengths and weaknesses. Chapter 3 introduces related work in four categories, focusing on dictionary creation.

Chapter 4 outlines the properties of our dataset for semantic domains. The following Chapter 5 presents implementation-specific details of GUIDE. It shows our preprocessing pipeline and presents the architecture of our model. Chapter 6 shows our experimental setup. We both evaluated our predictions with the dataset and by asking language experts. Chapter 7 presents the results obtained and discusses the current challenges in five areas. The thesis suggests three directions for future work in Chapter 8. Finally, Chapter 9 summarizes the most important results and provides a brief answer to the three research questions.

[3]GitHub: `https://github.com/janetzki/GUIDE`

2 FOUNDATIONS

"Data are just summaries of thousands of stories –
tell a few of those stories to help make the data meaningful."

— *Dan Heath* [75]

This chapter introduces crucial terminology and provides background information on low-resource languages, parallel datasets, semantic domains, word alignment, and GNNs. For simplicity, we use the term *"word"* as a synonym for *"token"*, even if the token is a subword.

2.1 DEFINING "LOW-RESOURCE LANGUAGE"

In the following, we look at existing definitions for *"low-resource language"* and choose a definition for this thesis.

2.1.1 *Existing Definitions*

There is no clear and uniform distinction between low-resource languages and high-resource languages because *resourcefulness* is not discrete but a continuum [36]. Therefore, any criterion is arbitrary. Furthermore, the definition can change over time. NLP researchers defined low-resource languages using different criteria, such as the number of native speakers, the number of available datasets, or the availability of NLP tools [34, 38, 77]. Besacier et al. [8] define a low-resource language as a language without a unique writing system, a limited or no presence on the World Wide Web, no linguistic expertise, and/or no electronic resources, such as monolingual and parallel corpora, vocabulary lists, etc. These limitations make training data for these languages logistically difficult to obtain [70, 104]. With data-hungry techniques to train language models becoming mainstream, such as large-scale pretraining [19, 54], high-quality support for low-resource languages is still expensive, if not impossible. The NLLB team [89] defines low-resource languages as languages with fewer than one million sentences of publicly available example translations in 2022.

A complete picture of resourcefulness also includes systematic inequalities in the performance of NLP tasks [11] and societal aspects, such as the bias toward high-resource languages among NLP researchers [104].

2.1.2 *Used Definition*

In the context of this thesis, we use the NLLB Team's definition [89], assuming that every language that is not listed as one of the 53 high-resource languages in their FLORES-200 dataset [33, 35, 89] is a low-resource language.

2.2 PARALLEL DATASETS

This section describes why parallel datasets are helpful for multilingual NLP, reviews existing parallel datasets, and presents the *eBible corpus*, which we use.

2.2.1 *Relevance*

At the turn of the 19th century, French officer Pierre-François Bouchard discovered the Rosetta Stone, probably showing the most famous parallel texts [15, 87]. It is a tablet of black basalt with 100 engraved lines, showing the same text (except lost parts) written in Ancient Greek and two forms of Ancient Egyptian. Due to the discovery of these parallel texts, scholars were able to decipher the Egyptian scripts. This discovery shows how helpful even tiny amounts of parallel data can be in understanding low-resource languages.

Today, parallel datasets are a useful data source for NLP applications, such as machine translation [17, 104]. To train machine translation systems for high-resource languages, tens or hundreds of parallel sentences are available [36]. These sentences are typically available in English combined with another European language, Arabic, or Chinese. For most language pairs, such data is rare or does not exist.

2.2.2 *Review of Existing Parallel Datasets*

Table 2.1 shows six common parallel datasets. While Opus [90] is considered the largest published collection of translated language data [36], it has recently been overtaken by *Bloom Books* and the eBible corpus in terms of the number of languages covered. The eBible corpus [102] is the largest parallel corpus we found, although by crawling multiple sources, Bible translations for even 1,600 languages are available [76]. We decided to use a subset of Bible translations in 18 languages (with two additional languages from other sources, see Section A.1) of the eBible corpus as a parallel data source. It is possible to replace these

Bible translations with other Bible translations, which allows GUIDE to create dictionaries in hundreds – technically even thousands – of languages.

Table 2.1: Summary of large and useful public parallel corpora (taken and adapted from Haddow et al. [36]): Each URL was visited on 16/10/2023.

Corpus name	URL	Source	# Languages
eBible	github.com/BibleNLP/ebible	Bible translations	833
Bloom Books	https://bit.ly/3S3ZVNo	author community	> 650
Opus	opus.nlpl.eu/	gathered from many sources	> 500
FLORES-200	bit.ly/45404Df	translations from web articles	202
WikiMatrix	bit.ly/3DrTjPo	mined from Wikipedia	85
CCMatrix	bit.ly/3Bin6rQ	mined from CommonCrawl	80

Despite some disadvantages (e.g., the lack of modern words), the Bible has been considered a great resource for NLP, in particular for low-resource languages [17]. For example, it has been a useful source of bilingual dictionaries when no human translators or other linguistic resources are at hand (e.g., in emergencies, such as the earthquakes in Haiti [58] and Japan [68]).

2.2.3 *Using the eBible Corpus*

Åkerman et al. [102] compiled the eBible corpus, which covers languages from 75 language families and 1,009 Bible translations in total (some languages have multiple translations), as illustrated in Figure 2.1. A lot of translations are in languages that are even considered extremely low-resourced, and it may be difficult to find additional texts in some languages.

A full Protestant Bible translation (i.e., the 66 canonical books) contains roughly 31,000 verses and 800,000 words in English [17]. Because not all Bible translations in the eBible corpus are complete yet, most of them have less than 30,000 verses, while at least 145 cover more than 30,000 verses. Each Bible translation in the corpus is a text file with one line per verse[1] (i.e., the corpus is verse-aligned).

[1]Some Bible translations span multiple verses in a single line. We ignore these exceptional cases in the following.

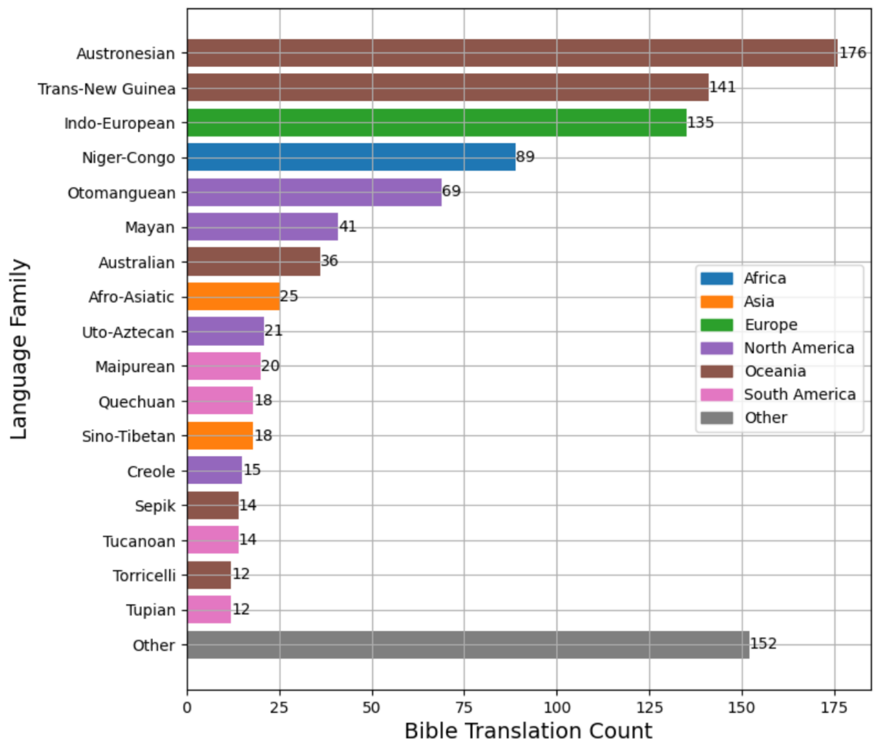


Figure 2.1: Count of Bible translations by language family and region in the eBible corpus (taken from [102]): The majority of translations (38%) belong to Oceania languages.

2.3 SIL'S SEMANTIC DOMAINS

As part of our dataset, we use dictionaries structured by SIL's semantic domains provided by FLEx[2]. This section describes the historical background and structure of semantic domains.

2.3.1 *History*

In 2001 and 2002, Ronald Moe, a linguistics consultant and the creator of SIL's semantic domains, had the goal to compile *"an exhaustive list of domains that could be used for any language in the world"* [67]. Thus, he built a standardized taxonomy of originally 1,600 semantic domains and met speakers of the languages Lugwere, Lunyole, and Kitharaka to compile dictionaries using this new approach. While many published bilingual dictionaries contain 3,000 – 5,000 entries, Moe's workshops collected 13,000 words in nine days with 17 participants on average, increasing the speed of word collection by more than 600 times

[2]A list of languages with existing semantic domain dictionaries is on this FieldWorks page: `https://software.sil.org/fieldworks/download/localizations/` (visited on 16/10/2023). The "17" in Figure 1.2 counts only those languages with a semantic domain translation progress of at least 10%.

compared to previous approaches that collected words at an estimated average rate of 2.5 words/day. At this speed, even creating a modest dictionary would take 20 years.

Since then, semantic domains have been used in *rapid word collection* [12] workshops as a "*mass production technique*" [67] to collect words in low-resource languages. Despite their effectiveness, these workshops with native speakers are still expensive, as they involve traveling and about 200 man days per language. Our work attempts to reduce the cost of semantic domain dictionary creation by automatically creating them and filling up missing words.

2.3.2 *Structure*

SIL's semantic domains [67] (i.e., areas of meanings) are a language-agnostic, standardized taxonomy to create dictionaries that mirror any aspect of the world, arranging words in 1,783 groups, divided into one or more SDQs. For each SDQ, the dictionaries list corresponding words, as shown in Figure 2.2. Semantic domains, questions, and words form a tree-structured graph[3], displayed in Figure 2.3.

Each semantic domain consists of

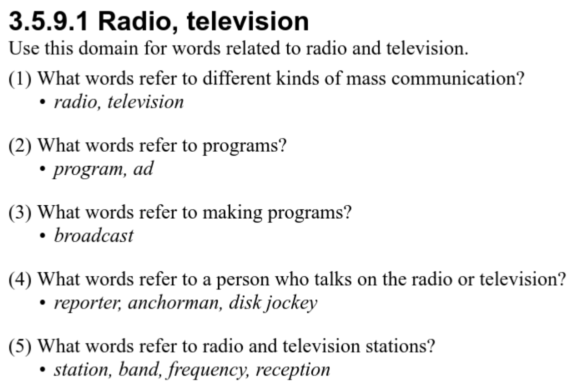
- an *identifier* (ID) (e.g., "*5.6.2*"),
- a name (e.g., "*Bathe*"),
- a short description,
- one or more SDQs, and
- a list of entries (matching words or phrases) for each SDQ.

In the following, we use the notation "*5.6.2-4*" as SDQ ID for the 4th question of semantic domain 5.6.2. Note that every semantic domain, not only leaf semantic domains, has SDQs (i.e., 5.6-1 also exists).

2.4 WORD ALIGNMENT

Word alignment is the task of identifying which word(s) in one language correspond to which word(s) in another language, given a pair of sentences that are translations of each other [103]. Figure 2.4 shows

[3]The entire hierarchy of semantic domains can be explored at the official page: `https://semdom.org/v4/1` (visited on 09/10/2023).



3.5.9.1 Radio, television

Use this domain for words related to radio and television.

(1) What words refer to different kinds of mass communication?
- *radio, television*

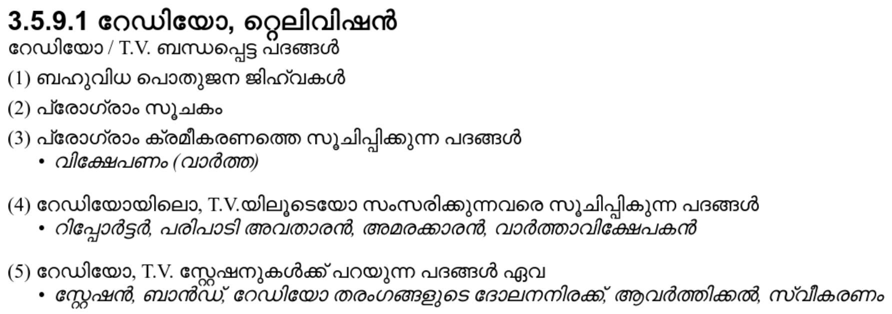
(2) What words refer to programs?
- *program, ad*

(3) What words refer to making programs?
- *broadcast*

(4) What words refer to a person who talks on the radio or television?
- *reporter, anchorman, disk jockey*

(5) What words refer to radio and television stations?
- *station, band, frequency, reception*

(a)



3.5.9.1 റേഡിയോ, റ്റെലിവിഷൻ

റേഡിയോ / T.V. ബന്ധപ്പെട്ട പദങ്ങൾ

(1) ബഹുവിധ പൊതുജന ജിഹ്വകൾ

(2) പ്രോഗ്രാം സൂചകം

(3) പ്രോഗ്രാം ക്രമീകരണത്തെ സൂചിപ്പിക്കുന്ന പദങ്ങൾ
- *വിക്ഷേപണം (വാർത്ത)*

(4) റേഡിയോയിലൊ, T.V.യിലൂടെയോ സംസരിക്കുന്നവരെ സൂചിപ്പികുന്ന പദങ്ങൾ
- *റിപ്പോർട്ടർ, പരിപാടി അവതാരൻ, അമരക്കാരൻ, വാർത്താവിക്ഷേപകൻ*

(5) റേഡിയോ, T.V. സ്റ്റേഷനുകൾക്ക് പറയുന്ന പദങ്ങൾ ഏവ
- *സ്റ്റേഷൻ, ബാൻഡ്, റേഡിയോ തരംഗങ്ങളുടെ ദോലനനിരക്ക്, ആവർത്തിക്കൽ, സ്വീകരണം*

(b)

Figure 2.2: The semantic domain *3.5.9.1 Radio, television* in the English and Malayalam dictionary: It consists of five SDQs. Note that the Malayalam dictionary contains only entries for three of them.

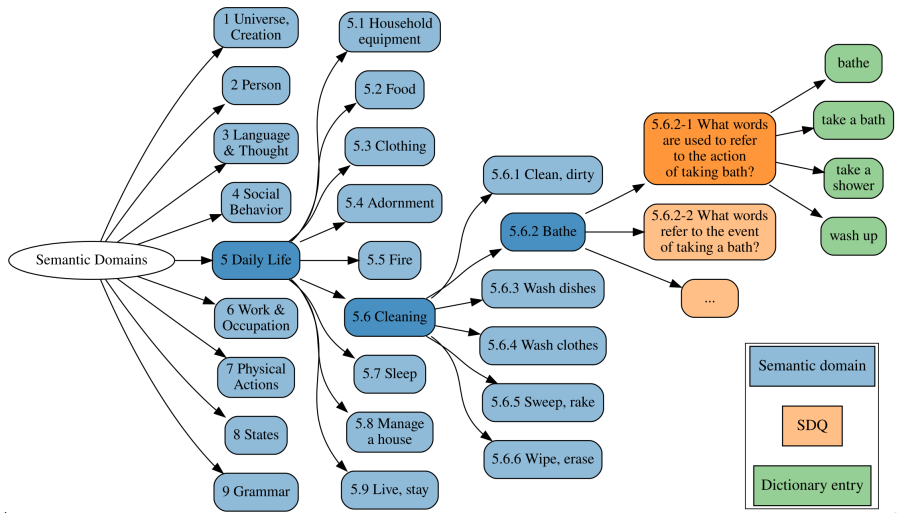


Figure 2.3: A drill-down into the tree-structured hierarchy of semantic domains: Nodes with expanded children are highlighted.

alignments of three pairs of sentences, forming a *Multilingual Alignment Graph* (MAG). The triangle-shaped alignments indicate that the languages have a similar syntax.

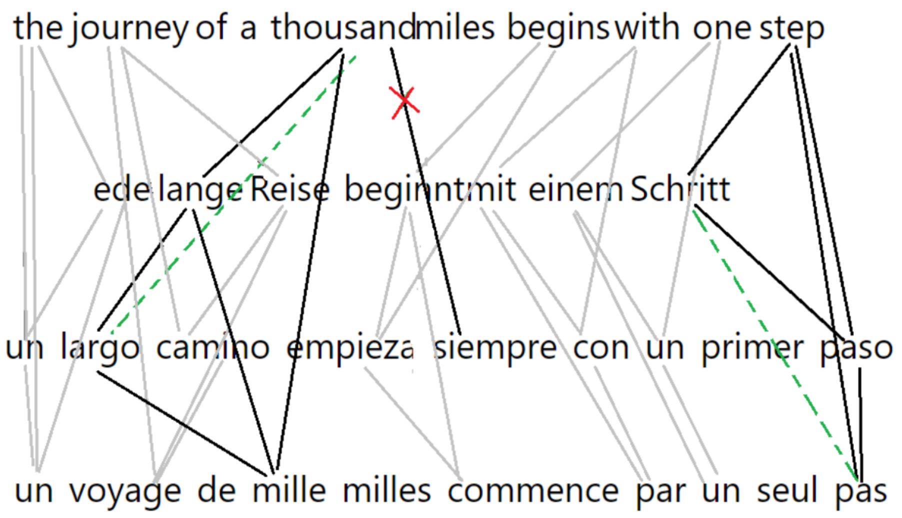

the journey of a thousand miles begins with one step
ede lange Reise beginnt mit einem Schritt
un largo camino empieza siempre con un primer paso
un voyage de mille milles commence par un seul pas

Figure 2.4: A MAG built from bilingual Eflomal alignments of a verse in English, German, Spanish, and French (taken from [40]). The dotted green lines are edges Eflomal did not find. The detected alignment between English "*thousand*" and Spanish "*siempre*" (always) is incorrect.

Although there exist many strong pretrained neural word aligners for high-resource languages (e.g., SimAlign [43] and AWESoME [22]), we assume that statistical, language-agnostic tools have a lower *Alignment Error Rate* (AER) for most low-resource languages. Two common statistical, language-agnostic word aligners are fast-align [25] and Eflomal [103].

Eflomal has a significantly lower AER than fast-align [103], therefore we have chosen Eflomal to align the words in the Bible verses. It is a Bayesian model using *Markov Chain Monte Carlo* (MCMC) inference and does not need any supervision signal apart from the parallel text.

Imani Googhari et al. [41] have also developed a GNN-based alignment method specifically designed for multiparallel corpora (i.e., with more than two parallel texts) that outperforms Eflomal's recall and $F1$ score on three language pairs. Unfortunately, we have not been able to reverse-engineer and adapt their code. Moreover, the same author used Eflomal in a successive paper [42], presumably because of its high precision, and we decided to do the same.

2.5 GRAPH NEURAL NETWORKS

The GNN is a class of neural networks [83] of which we use a subclass, the *Graph Convolutional Network* (GCN), to predict word-SDQ links, as presented in Subsection 5.3.2. This section describes typical applications of GNNs and the mathematical foundations of GCNs.

2.5.1 *Applications*

GNNs [83] have rapidly gained popularity in recent years [3, 100]. This type of machine learning model can effectively solve problems from real-world domains with an underlying graph structure (e.g., social networks, chemo-informatics, or recommender systems) [3, 81]. Previous work has applied GNNs to language data by first transforming raw text into a graph [7, 63, 101], such as for text classification [16]. We also use this approach to investigate whether GNNs are suitable for building semantic domain dictionaries for low-resource languages.

2.5.2 *GCN Layer*

A common GNN architecture is the GCN by Kipf and Welling [46], which has similarities to a *Convolutional Neural Network* (CNN)[49]. Both can be trained with supervision and perform convolutions. The main difference is that a CNN aggregates data using a raster representation (i.e., it is a *2D convolution*), while a GCN aggregates data with a graph representation, as shown in Figure 2.5. Each node holds a vector representation of the sample it represents (e.g., a word). The edges are relationships (e.g., word alignments).

Although nodes initially "know" only about themselves, their representation "learns" about their neighbors – each convolution corresponding to one GCN layer. A notion for this learning process is that the node's perceptive field grows with each step by one hop. It works by *neighborhood aggregation*, as shown in Figure 2.6. In the neighborhood aggregation process, each node collects features from its neighboring nodes to update its representation of the nearby graph structure.

The formula for a GCN layer is given by:

$$GCN(X, A) = \sigma(\tilde{D}^{-\frac{1}{2}} \tilde{A} \tilde{D}^{-\frac{1}{2}} X \Theta)$$

Here, X is the input feature matrix, A is the adjacency matrix, $\tilde{A}$ is the normalized adjacency matrix, $\tilde{D}_{ii} = \sum_{j=0} \tilde{A}_{ij}$ is the diagonal degree matrix, Θ is the weight matrix and σ is a non-linear activation function. A can also contain values different from 0 or 1 representing

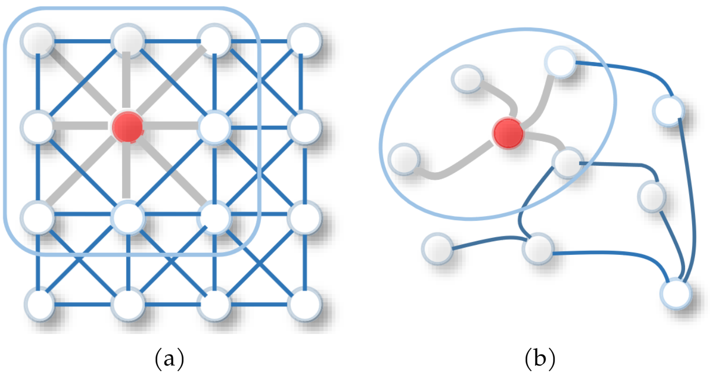


Figure 2.5: 2D convolution compared to graph convolution (taken from [97])

(a) 2D convolution: 2D convolution in images is a special case of graph convolution, for which each pixel is a node. The 2D convolution takes the weighted average of the pixel values of the red node and its neighboring nodes [97].

(b) Graph convolution: To obtain the embedding of the red node, a straightforward approach to graph convolution is to calculate the average of the red node's and its neighbors' features. Unlike for image data, the node degree can vary [97].

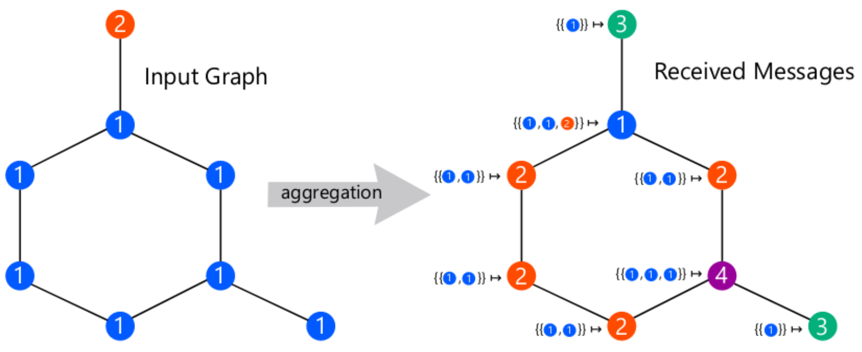


Figure 2.6: Node aggregation in a single-layer GCN (taken from [81]): The *messages* are the feature vectors from neighbor nodes. Colors and numbers denote features and embeddings. The same color and number indicate the same vector. Note that there are no self-loops in this illustration.

edge weights. Note that this formula has no self-loops, which corresponds to the GCN implementation we use.

Knowing the essential terminology of the research field, we present relevant existing work in the next chapter.

3 RELATED WORK

"Dictionaries are like watches, the worst is better than none and the best cannot be expected to go quite true."

— Samuel Johnson [14]

Various teams did exciting research on topics similar to dictionary creation for low-resource languages. This chapter presents some selected works in the categories of dictionary creation, machine translation, domain adaption, and annotation projection.

3.1 DICTIONARY CREATION

Existing approaches to creating dictionaries address different sets of languages and map words to ontologies or words of other languages. This section gives an overview of what their teams already have attained and highlights similarities to our work as well as the research gaps that we address.

The *Universal Wordnet* (UWN) is a graph-structured knowledge base for more than 200 languages that Melo and Weikum [65] automatically generated. They used several data sources, including existing monolingual wordnets, bilingual dictionaries, and parallel corpora. As a scaffold, they used *Princeton WordNet* [28], which provides a semantic hierarchy of English terms and enriched it with more than 1.5 million new semantic links for more than 800,000 words. Our work has a similar goal, as we investigate how to automatically create and enrich another linguistic resource. We use the semantic domains as a scaffold and focus on low-resource languages, for which we assume only a small amount of parallel text.

Knyazeva, Mareüil, and Vernier [47] successfully demonstrate the usefulness of even tiny amounts of multiparallel text. They use Aesop's fable "*The North Wind and the Sun*" as a parallel data source to create bilingual dictionaries. It has only five sentences and has been translated into more than 200 languages/dialects from France and French Polynesia (from the Indo-European and Austronesian language family, respectively). They have achieved correct mappings for 96 – 97% of the words. Similar to our approach, they use a word aligner. A difference is that they also take orthography (i.e., cognates) into account due to the relatedness of the languages/dialects by aligning words with a small Levenshtein distance [57]. Our work, on the other hand, does not assume that the languages are related, but we attempt to create dictionaries in any language for which a Bible translation exists.

In their inspiring paper, Alnajjar et al. [2] show how to find new translations of words in three endangered Uralic languages. Their key idea is to construct a graph of words in these and other languages, with known translations as unweighted bidirectional edges. An advantage of their approach is that it does not require parallel texts or word alignment. By predicting missing links in this graph, they built new bilingual dictionaries that help preserve these endangered languages. Similarly, we build a graph in which words in different languages are separate nodes. But there are two important differences:

1. GUIDE also builds dictionaries for languages without labeled data.
2. Instead of predicting word-to-word translations, we group words by their SDQs. This approach allows us to build highly multi-parallel dictionaries because words often have no 1:1 translations across languages but have different semantic ranges. SDQs provide for this flexibility.

To the best of our knowledge, Biswas et al. [10] have proposed the only approach to automatically fill SIL's semantic domain dictionaries (besides this work). They were able to map words in more than 500 languages to their semantic domains. For 50 languages, they used *ML-Commons*[1] as a parallel data source and *PanLex* [44] for the other languages. These corpora provide manual translations from non-English to English words, and they could successfully map 44% of non-English words through these translations. Although 76% of these words could be mapped to their semantic domains directly, they were able to map the remaining 24% by searching for synonyms in the *word2vec* [66] embedding space. However, PanLex contains at most 207 words per language, which bottlenecks the dictionary creation for low-resource languages. Unfortunately, we could not find an evaluation of their approach's precision, making it difficult to use their work as a baseline. Although this work has a quite similar goal, there are two key differences, and we decided to establish a new benchmark:

1. We link words to SDQs instead of their semantic domain, which have a 4.6 times finer granularity (see Section 4.3).
2. We do not rely on existing word-to-word translations, which allows us to link thousands of words per language to their SDQs (see Subsection 7.1.1).

[1] `https://mlcommons.org/` (visited on 21/10/22)

Based on the reviewed related work on dictionary creation, the research gap can be summarized as follows: There is a need to create highly multiparallel dictionaries for low-resource languages without labeled data, which is possible by using existing parallel texts.

3.2 MACHINE TRANSLATION

Machine translation goes one step further than dictionary creation because it can translate entire sentences, not just words. This research field is relevant to us because dictionaries created with GUIDE can help improve machine translation, as discussed in Section 8.2 and Section 8.3.

A recent breakthrough for low-resource languages was the public release of *No Language Left Behind* (NLLB)-200 [89], a *Neural Machine Translation* (NMT) model that supports 202 languages. Meta AI has trained this model on aligned sentences from their *Facebook Low Resource* (FLORES)-200 [33, 35, 89] dataset, which is closer to web content than the Bible. FLORES-200 consists of 3,001 parallel sentences with 21 words on average. To create this dataset, they used large-scale bitext mining, which automatically aligns sentences in non-aligned monolingual datasets. Their bitext mining approach is based on the concept of learning a multilingual sentence embedding space and then utilizing a similarity measure in that space to determine whether two sentences are parallel or not. They first collected about a billion bitext sentence pairs, in which they identified 842 distinct multiparallel web articles. Although NLLB-200 overcomes the limited domain diversity of the Bible, it is limited to languages with monolingual data on the Web. Our approach, on the other hand, also works for languages for which the expensive bitext mining is not a promising option because there is too little monolingual data on the Web, as shown in Figure 1.2.

3.3 DOMAIN ADAPTATION

Another notable approach to making NLP in low-resource languages is domain adaptation [78], which uses transfer learning to improve the capabilities of an existing *Large Language Model* (LLM) on a target language.

Dobler and Melo [21] propose *Fast Overlapping Token Combinations Using Sparsemax* (FOCUS), an embedding initialization method that adapts XLM-R [19] to low-resource languages. Their approach outperforms previous systems on a question-answering task in German, Arabic, and

Swahili. FOCUS allows a resource-efficient warm start by reusing the pretrained language model, which requires only about 30 – 60 MB of monolingual training data. Besides our focus on dictionary creation, a key difference to our approach is that we assume no language data other than a Bible translation in the target language, which is about 4 – 12 MB, depending on the script.

3.4 ANNOTATION PROJECTION

Semantic domain dictionary creation can also be understood as a special case of annotation projection [84]: The SDQs are the annotations that we want to project from annotated to yet unannotated words by using aligned translations.

A fascinating development in this area is the *Graph Label Propagation* (GLP) method by Imani et al. [42], an unsupervised GNN-based annotation projection method for *Part-of-Speech* (POS) tagging in low-resource languages. As source data, it uses the word-aligned text of the Bible with Eflomal, annotated with their POS tags. Although we also use Bible translations, there are no publicly available word-wise ground-truth annotations for their SDQs yet[2]. However, we heuristically created such annotations and tried to reuse GLP to propagate links to SDQs instead of POS tags, and we have also been in contact with the main author. Unfortunately, we have achieved an accuracy of only 0.06 and we could also not reproduce their results. A possible explanation for the low performance is that the sheer number of classes (7,425 SDQs instead of 20 POS tags) did not fit the model architecture. We decided to implement a custom GNN that does not require word-wise SDQ annotations but only semantic domain dictionaries in addition to the Bible translations.

As the related work shows, each approach is driven by the available data. The next chapter looks at the data that drives GUIDE in detail.

[2]For a different semantic domain ontology, the *Macula* linguistic dataset provides such mappings: `https://github.com/Clear-Bible/macula-hebrew`, `https://github.com/Clear-Bible/macula-greek` (visited on 22/10/23). Unfortunately, there exists no 1:1 mapping between their and SIL's ontology.

4 DATASET

"Data is a precious thing and will last longer than the systems themselves."

— Tim Berners-Lee [14]

In Subsection 2.2.3, we have explained our choice to use the eBible corpus. This chapter outlines the properties of the dataset that we use for development and testing. It covers 20 languages in total: twelve development (i.e., training) languages and eight test languages. The difference between both is that our dataset also contains semantic domain dictionaries for the development languages, which serve as labels/supervision, while there are no labels for the test dataset. We use the term "*node*" as a synonym for "*word*" because each word becomes a node in the MAG that we build.

First, Section 4.1 shows the languages that we selected for our dataset, followed by a description of its size in Section 4.2. Section 4.3 explains the distribution of classes (i.e., SDQs) and Section 4.4 closes by discussing the quality of the dataset.

4.1 SELECTED LANGUAGES

Table 4.1 displays the twelve languages that we use to train our model and the eight languages that we use for testing. In total, ten of them are low-resource languages. The languages belong to seven distinct language families (counting both Creole languages as members of the same language family). The majority of the languages (eight) are Indo-European languages or use the Latin script (16). We have selected languages based on the availability of data, the availability of language speakers for evaluation, and the language family (to cover a broad spectrum).

African languages have historically been underrepresented in translation efforts [104], and we test our approach on three of them: Igbo, Mina-Gen, and Yoruba. Igbo and Yoruba are an especially rewarding frontier of NLP because they are low-resourced despite having more than 30 million speakers each. [104]

Table 4.1: Language information: Language name, ISO 639 code [26], Number of speakers [26], Language family [26], and resourcefulness for the 20 languages in our dataset. We defined "resourcefulness" in Section 2.1.

Language	ISO	# Speakers	Language family	Res.
DEVELOPMENT				
Bengali	ben	273M	Indo-European	High
Chinese (simplified)	cmn	1.14B	Sino-Tiebetan	High
English	eng	1.46B	Indo-European	High
French	fra	310M	Indo-European	High
Hindi	hin	610M	Indo-European	High
Indonesian	ind	199M	Austronesian	High
Kupang Malay	mkn	350k	Creole (Malay-based)	Low
Malayalam	mal	37.4M	Dravidian	Low
Nepali	npi	25.6M	Indo-European	Low
Portuguese	por	260M	Indo-European	High
Spanish	spa	559M	Indo-European	High
Swahili	swh	71.6M	Niger-Congo	High
TEST				
German	deu	133M	Indo-European	High
Hiri Motu	hmo	95.0k	Austronesian	Low
Igbo	ibo	30.9M	Niger-Congo	Low
Mina-Gen	gej	620k	Niger-Congo	Low
Motu	meu	39.0k	Austronesian	Low
South Azerbaijani	azb	14.9M	Turkic	Low
Tok Pisin	tpi	4.13M	Creole (English-based)	Low
Yoruba	yor	45.9M	Niger-Congo	Low

4.2 DATASET SIZE

Table 4.2 shows the size of our dataset for each of these languages, split in the Bible translations as a parallel text corpus and the semantic domains as labels. Most Bible translations include the entire Old Testament and the entire New Testament. Such translations have approximately 31,000 verses. Kupang Malay and Indonesian are the languages for which we have the lowest amount of verses: Both cover the New Testament and portions of the Old Testament. For the languages English, French, Spanish, Mina-Gen, and Tok Pisin, the data also comprises "*deuterocanonical*" books in addition.

[1]FLEx provides a South Azerbaijani dictionary with nine entries, which we ignore due to its small size.

Table 4.2: Dataset size: "Dicts." means "Semantic domain dictionaries", "V." means "Verses" and "W." means "words". The matched number of words refers to the number of dictionary entries that also appear as words in the respective Bible translation (i.e., it is the intersection of the previous two columns). The Bible translations' source URLs are listed in the Appendix (Section A.1).

	BIBLE TRANSLATIONS			DICTS.	COMBINED	
Language	Sample	# V.	# W.	# Entries	# Matched w.	# Total w.
DEVELOPMENT						
Bengali	আলো হোক	31k	9.1k	0.91k	0.11k	10k
Chinese (simplified)	要有光	31k	32k	24k	4.2k	52k
English	Let there be light	37k	11k	26k	5.1k	32k
French	Que la lumière soit	37k	14k	30k	5.4k	39k
Hindi	उजियाला हो	31k	15k	22k	2.7k	35k
Indonesian	Jadilah terang	11k	4.8k	11k	1.9k	14k
Kupang Malay	Musti ada taráng	9.8k	7.9k	0.33k	0.3k	11k
Malayalam	പ്രകാശം ഉണ്ടാകട്ടെ	31k	9.6k	25k	0.72k	33k
Nepali	उज्यालो होस्	31k	9.6k	14k	1.4k	22k
Portuguese	Que haja luz	31k	12k	21k	3.5k	29k
Spanish	Sea la luz	37k	15k	29k	5.3k	38k
Swahili	na kuwe nuru	31k	8.9k	5.2k	0.90k	13k
TEST						
German	Es werde Licht	31k	20k	0	0	20k
Hiri Motu	Diari ia vara namo	31k	4.9k	0	0	4.9k
Igbo	Ka ìhè dị	31k	7.2k	0	0	7.2k
Mina-Gen	Kẽklẽ ne va e mè	35k	19k	0	0	19k
Motu	Diari aine vara	31k	14k	0	0	1.4k
South Azerbaijani	Qoy işıq olsun	31k	8.8k	0[1]	0	8.8k
Tok Pisin	Lait i mas kamap	36k	4.3k	0	0	4.3k
Yoruba	Jẹ́ kí ìmọ́lẹ̀ kí ó wà	31k	8.8k	0	0	8.8k
Sum			232k	216k	32k	415k

4.3 CLASS DISTRIBUTION

This section quantitatively describes the distribution of word-SDQ links in the development dataset. Throughout this thesis, we often say "*link*" instead of "*class*" because each word can belong to multiple SDQs and vice-versa. There are nine top-level semantic domains (see Figure 2.3), and we ignore the last one, called *Grammar*, because we consider grammar too language-dependent to be identifiable via alignments. In total, there are 7,908 SDQs of which 7,425 remain without *Grammar*. These 7,425 SDQs belong to 1,624 instead of 1,783 semantic domains, respectively.

SDQ PROPERTIES There are 4.6 SDQs per semantic domain on average and 81,632 word-SDQ links in total. 15 out of all 7,425 SDQs (i.e., 0.2%) have no word links. The SDQ with the most word links is 1.6.1.2-3:

"What species of bird are there?" with 999 linked words. SDQs have 40 word links on average with a standard deviation of 35.

WORD PROPERTIES 122,803 out of all 327,526 words in the development dataset (i.e., 37%) have no SDQ links (i.e., they are unmatched). The word with the most SDQ links is the Hindi word "किसी" (any) with 96 linked SDQs. Words have 0.56 SDQ links on average with a standard deviation of 1.3.

4.4 DATA QUALITY

Table 4.2 shows that four languages in our development dataset have less than 1,000 labeled (i.e., matched) words: Bengali, Kupang Malay, Malayalam, and Swahili. Of course, more words in these languages belong to SDQs. These low numbers of labeled words are mainly caused by three factors:

- The semantic domain dictionary is incomplete, which is the case for any language, especially Kupang Malay.
- The number of unique words in the Bible is low, as in the case of Indonesian.
- The entries in the semantic domain dictionary consist of multiple words (e.g., *"harvest moon"*). We match only single words from the Bible translations with dictionary entries and accept only exact matches. This exact matching means that we ignore the multi-word dictionary entries unless they can be matched with *Multi-Word Token* (MWT) expansion (see paragraph 5.2.2), which rarely happens.

Each of these three factors affects any language in the dataset. These limitations mean that we do not use *gold-standard* but rather *"silver"* data because it is clean but incomplete. Therefore, we expect that GUIDE has a high precision but a low recall. We further describe the word matching process in Subsection 5.2.4 and discuss the limitations of the dataset in paragraph 7.4.2 as well as ways to overcome them.

Having taken a look at GUIDE's input data, the next section shows how the tool preprocesses it and predicts new word-SDQ links.

5 DICTIONARY CREATION WITH GUIDE

"The art of being wise is the art of knowing what to overlook."

— *William James* [14]

GUIDE is a tool for linguists or any user of FLEx to create new entries for existing and empty semantic domain dictionaries. This chapter outlines its technical implementation. It describes the graph structure in Section 5.1, the preprocessing pipeline in Section 5.2, and finally, how the GNN works in Section 5.3.

5.1 GRAPH STRUCTURE

We transform our dataset (see Chapter 4) into a graph to take advantage of a GNN. Each node in this graph is a word in one of the 20 languages in the dataset. The unique key of each node is its language code and the word itself (e.g., "*eng: pelican*"). This spelling-based key means that we do not create separate nodes for identically spelled words with different meanings (e.g., the English word "*love*" [59]). The edges are the alignments between these words. Therefore, we call the graph MAG. We first create a *raw MAG*, which uses absolute alignment counts as edge weights. We then transform it into the *final MAG*, which uses normalized edge weights and contains only a filtered subset of the raw MAG's nodes and edges. This graph structure is similar to the MAG that Imani Googhari et al. [41] created to perform multiparallel word alignment, with one difference: While they create one MAG for each verse, we create a single MAG that aggregates all verses. We implemented the graph with the *NetworkX* [37] library.

Figure 5.1 shows the neighborhood of the English word "*pelican*" in the aggregated MAG. Four words from the development languages are in the middle, while the words from the test languages form a star around them because each of them has been aligned with the four words from the development languages. It is visible that the Portuguese word "*pelicano*" is missing a link to the SDQ.

Figure 5.2 shows the node distribution in the graph by language. These numbers are also the vocabulary sizes that GUIDE eventually uses to predict word-SDQ links. The Chinese tokenizer (see Subsection 5.2.1) produces many subwords that are not actual Chinese words, which is why the Chinese vocabulary is significantly bigger than the

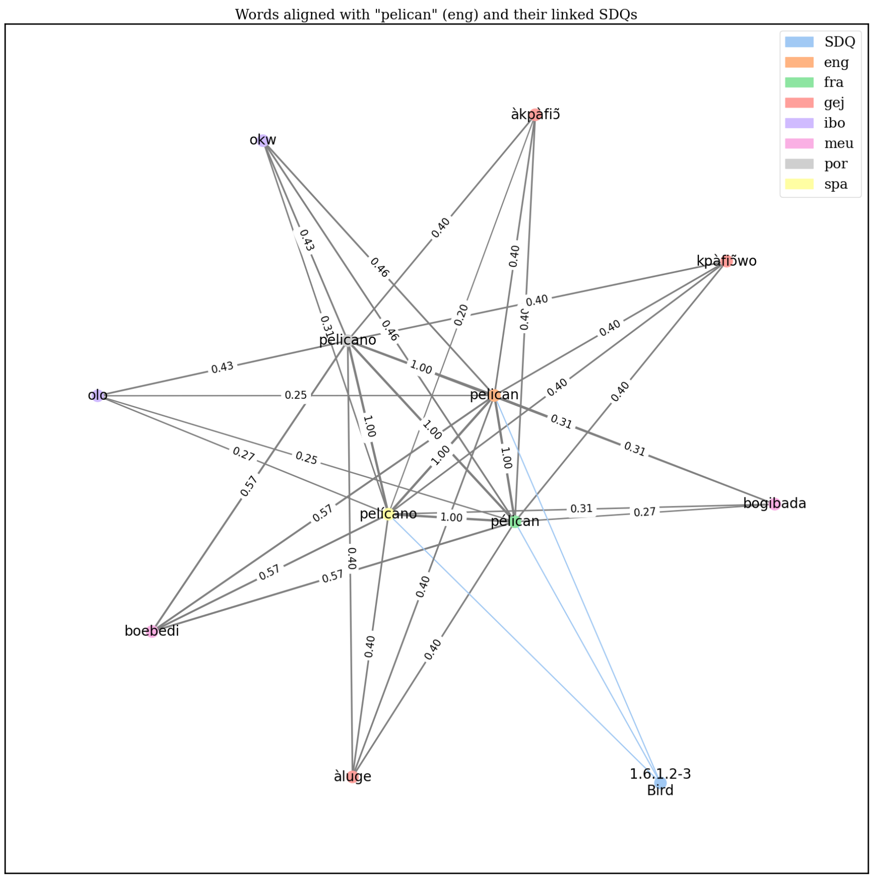


Figure 5.1: A subgraph from the final MAG showing the 1-hop neighborhood of the English word "*pelican*":
The gray edges are word alignments with their normalized strength. Edges with higher strengths are thicker. The blue edges are SDQ links. The shown SDQ 1.6.1.2-3 is "*What species of bird are there?*"
Restrictions: This SDQ is shown here as a separate node, although it is technically part of the word nodes' feature vectors. To improve readability, the graph excludes the following languages: Bengali, Chinese, Hindi, Malayalam, Nepali, German, Hiri Motu, Motu, South Azerbaijani, Tok Pisin, and Yoruba.

other languages' vocabularies. We discuss this limitation in Subsection 7.1.3.

5.2 PREPROCESSING PIPELINE

GUIDE's preprocessing pipeline converts our dataset into the raw MAG (see Figure 5.3) and converts the raw MAG into the final MAG (see Figure 5.4). The final MAG is then enriched with node features (see Subsection 5.3.1) and is used as input for the GNN (see Subsection 5.3.2). In the following, we describe the preprocessing steps in detail.

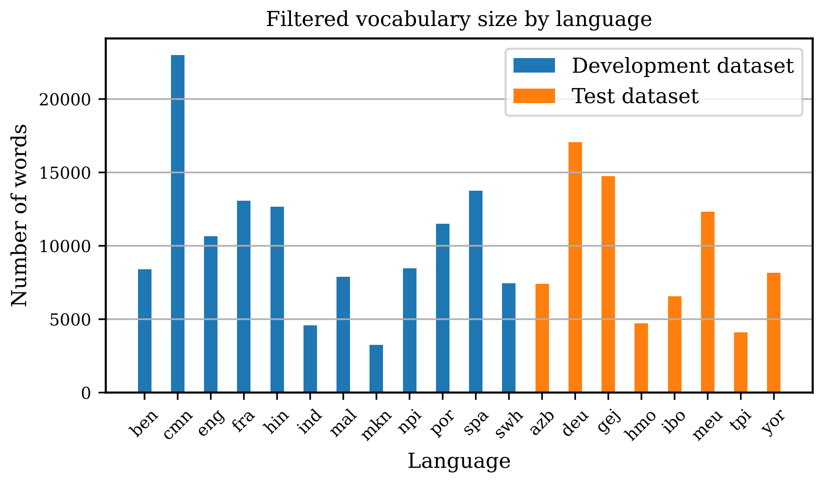


Figure 5.2: The number of words/nodes by language after the preprocessing described in Section 5.2: In total, the final MAG contains 199,605 words.

5.2.1 *Tokenization*

The first step of our preprocessing pipeline is tokenization. Depending on the language, we use different tokenizers.

STANZA TOKENIZER *Stanza* [74] is the successor to *Stanford CoreNLP* [61]. It has been trained using the *Universal Dependency Treebanks* [64] as well as other multilingual corpora. The Stanza tokenizers are our preferred tokenizers for three reasons:

1. The Stanza tokenizers have a large community and are well-tested.
2. The Stanza tokenizers have been built for each language individually instead of relying on general heuristics.
3. Stanza provides tokenizers for 80 human languages[1] in total (the highest number we found for toolkits that also fulfill (1) and (2)).

As shown in Figure 5.3, such a tokenizer exists for eight of the 20 languages in our dataset: Chinese (simplified), English, French, Hindi, Indonesian, Portuguese, Spanish, and German. All of them are high-resource languages. Due to (3), GUIDE's Stanza tokenization step is extensible to 72 more languages.

[1]Stanza: https://stanfordnlp.github.io/stanza/performance.html (visited on 04/10/2023)

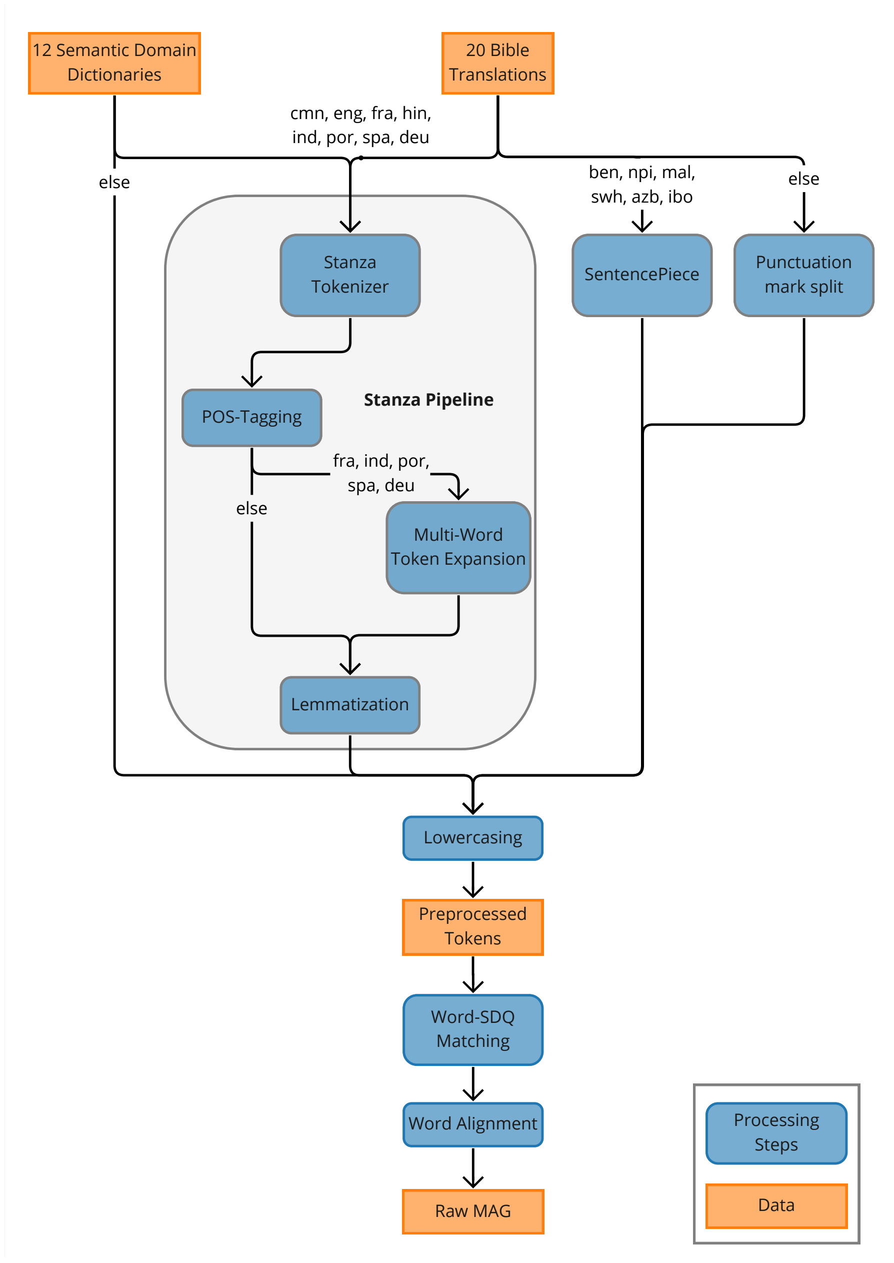


Figure 5.3: The first part of the preprocessing pipeline (graph creation)

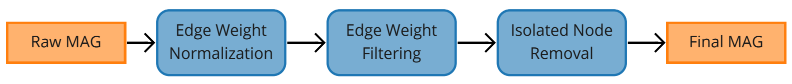


Figure 5.4: The second part of the preprocessing pipeline (graph refinement)

SENTENCEPIECE If the Stanza toolkit does not provide a tokenizer, we use a language-agnostic tokenizer. For six agglutinative languages (Bengali, Malayalam, Nepali, Swahili, South Azerbaijani, and Igbo), we use *SentencePiece* [51], which identifies subwords (e.g., with *Byte Pair Encoding* (BPE) [85] and a unigram language model [50]). We use SentencePiece for three reasons:

1. SentencePiece often splits agglutinated words into tokens with an atomic meaning.
2. SentencePiece is language-agnostic.
3. SentencePiece does not require supervision.

An additional advantage of SentencePiece is that it makes GUIDE extensible to languages without spaces (e.g., Thai), although we do not have such a language in our dataset.

We train the SentencePiece tokenizer for each of these six languages with a vocabulary size of 10,000. 10,000 words are also in the same order of magnitude as the numbers for the other languages with no fixed vocabulary size.

PUNCTUATION MARK SPLIT If we cannot use Stanza and the language is not agglutinative, we simply split at punctuation marks (including whitespaces). The advantage of this simple tokenization is that words cannot become fragmented. We use punctuation mark split for Kupang Malay, Hiri Motu, Mina-Gen, Motu, Tok Pisin, and Yoruba.

5.2.2 *Further Stanza Processing Steps*

For each language covered by the Stanza toolkit [74], we perform additional preprocessing steps.

POS-TAGGING The first step is POS-Tagging, which is a prerequisite for the lemmatization process.

MULTI-WORD TOKEN EXPANSION For French, Indonesian, Portuguese, Spanish, and German, Stanza also provides models for MWT expansion. This process merges common combinations of tokens. It produces, for example, "*arc-en-ciel*" (rainbow) in French and "*guarda-costa*" (coastguard) in Portuguese.

LEMMATIZATION We lemmatize words for two reasons:

1. We are interested in the base form of words, which often are the most suitable form for a dictionary entry.

2. We can aggregate more alignments per word by condensing the vocabulary language. However, the Stanza lemmatizer sometimes acts as a stemmer. For example, the German word "*Hüfte*" (hip) becomes "*Huft*" (which does not exist). We discuss this limitation in Subsection 7.4.5.

5.2.3 *Lowercasing*

We normalize the words in all languages with Latin script by lowercasing them, which is the final step in preprocessing the tokens.

5.2.4 *Word-SDQ Matching*

For all languages in our development dataset, we assign all matching SDQs to each word. We perform this matching by simply looking for exact matches in the semantic domain dictionary for the respective language. Note that we do not match words with partially matching dictionary entries. For example, we do not match "*harvest*" or "*moon*" with the dictionary entry "*harvest moon*". The word-SDQ matching ignores such dictionary entries with multiple tokens, except if they have been merged into a single word via MWT expansion (see paragraph 5.2.2). Many words do not match any SDQ (see Section 4.3). We also keep those words to predict their SDQs.

5.2.5 *Word Alignment*

The core assumption of this thesis is that words with similar meanings align. Similar to Imani Googhari et al. [41], we use the Eflomal statistical word aligner (see Section 2.4) to generate bilingual alignments for each language pair in our dataset, except pairs of two test languages because both have no labels. We post-process Eflomal's unidirectional, asymmetrical alignments with atools[2] and the *grow-diag-final-and* (GDFA) heuristic [48] to obtain symmetric bilingual alignments. The main advantage of Eflomal is that it does not require supervision apart from parallel data, which makes it applicable to any language pair that has been tokenized. We also aggregate all alignments by word, resulting in the raw MAG.

[2]fast-align repository: `https://github.com/clab/fast_align` (visited on 20/10/2023)

5.2.6 *Graph Refinement*

Before we can pass the MAG to a GNN to make predictions, three processing steps further refine it.

EDGE WEIGHT NORMALIZATION In the raw MAG, each edge between two word nodes u and v has a weight $w_{raw}(u, v) \in \mathbb{N}^+$ that we convert to a normalized weight $w_{norm}(u, v) \in (0, 1]$:

$$w_{norm}(u, v) = 2 * \frac{w_{raw}(u, v)}{S_{lang(v)}(u) + S_{lang(u)}(v)}$$

$S_{lang(v)}(u)$ is the strength of node u concerning the language of v (i.e., it is the sum of the edge weights of all edges from word u to a word in language v). Verbally expressed, the normalized edge weight is the raw edge weight divided by the average strength of both nodes concerning the other language.

EDGE WEIGHT FILTERING To reduce noisy alignments, we remove all edges (u, v) with a weight $w_{norm}(u, v) < 0.2$. We empirically discovered that 0.2 works well.

ISOLATED NODE REMOVAL As the final preprocessing step, we remove all words from the graph that have no edge to a word in the development dataset, including words from such development languages. We call such words *isolated* even though they may have neighbors in a test language, such as Mina-Gen. This process reduces the number of nodes in the MAG by 52% – from 414,964 to 199,605.

We assume that any predicted SDQ link for an isolated word would be likely wrong since the GNN does not "see" any information about its neighbor's SDQ links. Deleting these nodes has two advantages: It makes the model more precise, and the learning and prediction are faster. However, it would be possible to make meaningful predictions for nodes that have 2nd order neighbors with SDQ links if we use a GNN with two or more GNN layers.

At the end of the preprocessing pipeline, the 20 Bible translations and 12 semantic domain dictionaries have been transformed into the final MAG: a graph of words, connected by edges that denote their normalized alignment count. This graph is suitable for inference with a GNN.

5.3 GNN SETUP

This section describes GUIDE's GNN and how it performs multi-class multi-label classification. Each class is an SDQ. The task can be described as: *"Given a word W that aligns with these words from other languages and that belong to these SDQs, what are the SDQs of word W?"* An approach that focuses on high recall would link the word *W* to all the SDQs of its neighbors, while an approach that focuses on high precision would link the word *W* only to those SDQs that are more often linked to neighbors with high edge weights. We decided to focus on precision because a small and mostly correct dictionary can be used more easily by linguists or in a language community than a large but mostly incorrect dictionary.

5.3.1 *Node Features*

We train the GNN by representing each node with a set of features, using two main types of node features [23]:

- graph structural features and
- word meaning features.

GRAPH STRUCTURAL FEATURES. We incorporate *node degree* and *weighted node degree* as additional graph structural information. These two features are continuous numerical values.

WORD MEANING FEATURES. We use *SDQ count* and *SDQ link* features. While the SDQ count is a continuous number, the SDQ links are a multi-hot vector with 7,425 dimensions (i.e., these links are categorical features).

In total, each node/word receives a vector with 7,428 feature values. Subsection 7.2.1 examines the importance of each of these node features. We do not use a language feature because we cannot train this feature for the test languages. We also do not use embeddings from pretrained multilingual models (e.g., XLM-R [19]) because these models are not available for most low-resource languages, too. Note that there is noise in the SDQ links, as we are not using gold data but incomplete dictionaries (see Section 4.4).

5.3.2 *Model Architecture*

We implemented the GNN as a GCN using *Python* and *PyTorch Geometric* [31]. Figure 5.5 shows its fairly simple architecture: After adding the node features to the final MAG, the single-layer GCN (a *GCNConv*[3] layer) processes it. The GCN overwrites the features of each node with the aggregated features of its direct neighbors. The results are 7,425 scores per node, one for each SDQ. We normalize these scores with *sigmoid* as a non-linear output activation function.

Finally, we use a threshold, accepting only word-SDQ links with a score ≥ 0.999. We chose a high threshold because we prefer precision over recall. In simple words, a threshold of 0.999 means that the model is "sure" that its prediction is correct. The advantage of high precision is that it reduces the effort for manual post-filtering.

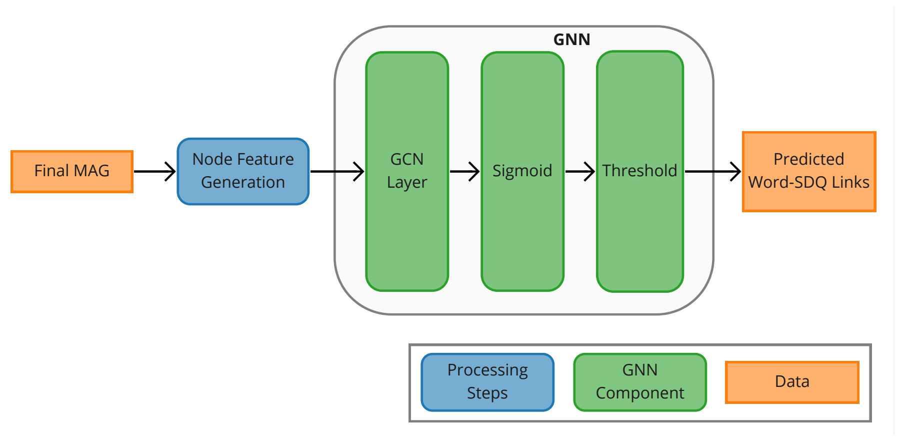


Figure 5.5: Model architecture: GUIDE takes the MAG with node features and predicts SDQ-links for these nodes.

The GCN does not use attention (unlike the *Graph Attention Network* (GAT) [93]). Although the absence of a transformer [92] makes the GCN less flexible regarding the patterns it can learn, it has two advantages:

1. The GCN's behavior is intuitive and easier interpretable (see Subsection 7.1.3).

2. The GCN's results have higher precision and recall than the GAT results (see paragraph 7.2.3).

The GCNConv layer has 55,160,325 parameters. This number results from a weight matrix with $7{,}428 \times 7{,}425$ weights and the corresponding bias vector with 7,425 more weights.

[3]PyTorch Geometric documentation: https://pytorch-geometric.readthedocs.io/en/latest/generated/torch_geometric.nn.conv.GCNConv.html (visited on 10/10/2023)

5.3.3 *Modified Identity Matrix Initialization*

After initializing the weight matrix and bias vector of our model's GCN layer with small random weights, we overwrite parts of it. Our initialization strategy is similar to an identity initialization, which uses an identity matrix as a weight matrix. The intuition behind these weights is that a node is likely to belong to the same classes as its neighbors. Figure 5.6 shows the result of this initialization method.

Our weight matrix has the shape 7,428 x 7,425 (see Subsection 5.3.1). Of the 7,428 input features, 7,425 are a multi-hot vector that encodes the SDQ links. Because this subvector has the same meaning as the output vector, we assume that the GNN should learn that if a word's neighbors belong to the SDQ 1.1.1-1: *"What words refer to the sun?"*, this word is also likely to refer to the sun. We modify the identity initialization by applying it to this 7,425 × 7,425 submatrix and using large weights (50.0). We also initialize the entire bias vector with low weights (-5.0). In consequence, the optimizer's learning process starts at the point that a word can belong only to the SDQ *"What words refer to the sun?"* if at least one neighbor does.

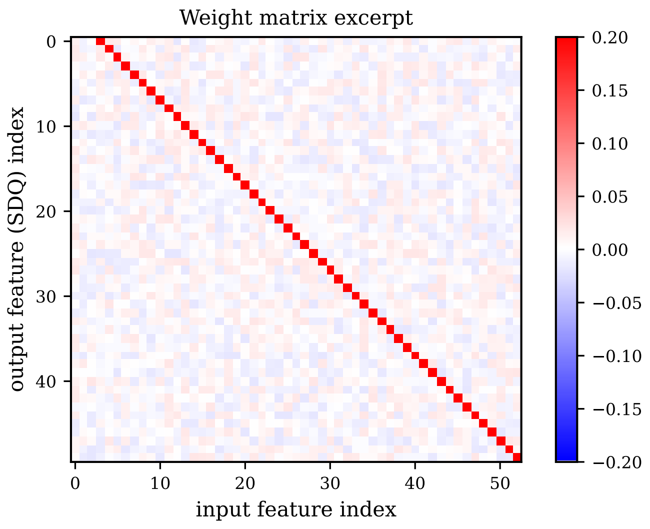


Figure 5.6: A heatmap showing the first 53 × 50 weights of the initial weight matrix: The modified identity matrix initialization sets each weight to 50.0 that corresponds to the same SDQ link feature in the GNN layer's input and output (red diagonal). Note that the color scale has been clipped to improve interpretability (i.e., there are weights greater than 0.2).

5.3.4 *Soft F1 Loss*

As the loss function, we use the soft $F1$ loss[4]. The soft $F1$ loss works similarly to the $F1$ score, except that it works with continuous (*"soft"*) instead of discrete values.

We calculate the *soft true positives* (tp_{soft}), *soft false positives* (fp_{soft}), and *soft false negatives* (fn_{soft}) as:

$$tp_{soft} = \sum y_{true} \odot y_{pred}$$
$$fp_{soft} = \sum (1 - y_{true}) \odot y_{pred}$$
$$fn_{soft} = \sum y_{true} \odot (1 - y_{pred})$$

$\odot$ is the element-wise multiplication.

$$precision_{soft} = \frac{tp_{soft}}{tp_{soft} + fp_{soft} + \epsilon}$$
$$recall_{soft} = \frac{tp_{soft}}{tp_{soft} + fn_{soft} + \epsilon}$$
$$F1_{soft} = max(\epsilon, min(2 \times \frac{precision_{soft} \times recall_{soft}}{precision_{soft} + recall_{soft} + \epsilon}, 1 - \epsilon))$$

Finally, the loss is:

$$loss = 1 - F1_{soft}$$

Here, ϵ is a small constant to ensure numerical stability (e.g., by preventing division by zero).

We have chosen the soft $F1$ loss to adjust the optimization to our goal of generating precise word-SDQ assignments without ignoring the recall. paragraph 12 shows that the model is significantly more accurate using soft $F1$ loss than using the more common *binary cross-entropy* [62] loss.

Knowing the GUIDE's technical details, the next chapter provides information about its configuration and how we evaluate it.

[4]Our implementation is inspired by this GitHub page: `https://gist.github.com/SuperShinyEyes/dcc68a08ff8b615442e3bc6a9b55a354` (visited on 16/10/2023).

6 EXPERIMENTAL SETUP

"The computer was born to solve problems that did not exist before."

— Bill Gates [75]

This chapter provides details about the environment in which we executed GUIDE and how we evaluated it to make our results reproducible.

6.1 HARDWARE

Our hardware is an *ASUS ESC8000 G4*. It consists of 500 GB of RAM, two *AMD Intel Xeon Silver 4214 processors* with twelve cores and 2.20 GHz, and eight *NVIDIA Quadro RTX A6000* (each having 48 GB VRAM) GPUs. We train the model on a single GPU. The entire training process takes less than 30 minutes and the inference time is approximately ten milliseconds per word.

6.2 CONFIGURATION

We use a single GCNConv layer with an input size of 7,428 (node feature vector size) and an output size of 7,425 (number of SDQs). We trained the model ten times in a range of 30 to 40 epochs on the development dataset with a batch size of 6,000 and a learning rate of 0.05 using the *Adam* optimizer [45]. We use early stopping after five epochs with a warm-up time of 30 epochs. We split our data with a random 80%/10%/10% node split.

6.3 EVALUATING GUIDE

A general challenge in evaluating the model's performance is that the development dataset covers only a subset of each language's words and word-SDQ links, as discussed in Section 4.4. Therefore, we use two evaluation methods:

1. evaluation on the incomplete semantic domain dictionaries,
2. manual evaluation with questionnaires.

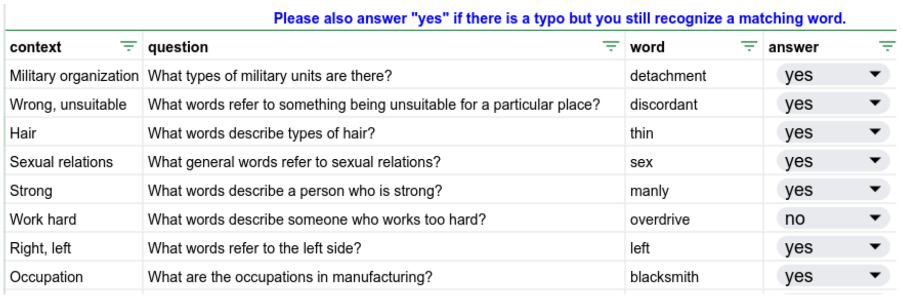
This section describes why we ignore empty questions in the dataset-based evaluation and how we conducted the manual evaluation.

6.3.1 *Ignoring Empty Questions*

In the calculation of soft $F1$ loss as well as precision, recall, and $F1$ score, we ignore "empty" SDQs. An empty SDQ is an SDQ that has no assigned words in the dataset in a specific language. For example, the Malayalam semantic domain dictionary lists no words for the SDQ 3.5.9.1-2: *"What words refer to [TV] programs?"* (see Figure 2.2). Instead of assuming that Malayalam has no words that fit this question, we assume that these words are simply unknown. Therefore, no word that the model assigns to this SDQ should affect precision or recall. Ignoring empty questions allows us to evaluate our model even using incomplete semantic domain dictionaries.

6.3.2 *Manual Evaluation using Questionnaires*

For each language, we have built a questionnaire to evaluate 100 – 120 random and shuffled predicted word-SDQ mappings (see Figure 6.1). The person who answered the questionnaire could select only *"yes"* or *"no"* for each pair. The 20 questions serve as a buffer because some participants skipped questions. However, we always consider only the first 100 answers in the evaluation. The author has not answered any of the questionnaires.



Please also answer "yes" if there is a typo but you still recognize a matching word.

context	question	word	answer
Military organization	What types of military units are there?	detachment	yes
Wrong, unsuitable	What words refer to something being unsuitable for a particular place?	discordant	yes
Hair	What words describe types of hair?	thin	yes
Sexual relations	What general words refer to sexual relations?	sex	yes
Strong	What words describe a person who is strong?	manly	yes
Work hard	What words describe someone who works too hard?	overdrive	no
Right, left	What words refer to the left side?	left	yes
Occupation	What are the occupations in manufacturing?	blacksmith	yes

Figure 6.1: The beginning of the questionnaire that evaluates 100 English predicted word-SDQ links

Section A.2 lists the URLs to the 20 completed questionnaires. To clarify the SDQs, we also provided a list of valid English answers for each SDQ (except in the English questionnaire). The 14 languages for which a tokenizer or lemmatizer could change a word's spelling (see Subsection 5.2.1 and paragraph 5.2.2) also included this note: *"Please also answer "yes" if there is a typo but you still recognize a matching word."* The remaining languages (i.e., all languages except Kupang Malay, Hiri Motu, Mina-Gen, Motu, Tok Pisin, and Yoruba) did not include this note because the tokenizer just splits the words at sentence tokens. Therefore, the words should not contain preprocessing-related *"typos"*. A previously mentioned example of such a *"typo"* is the German word

"*Hüfte*" (hip), which became "*huft*" after lemmatization and lowercasing (see paragraph 5.2.2).

Now that we have looked at GUIDE's execution environment, the next chapter shows and evaluates its predicted dictionary entries.

7 EVALUATION

"What gets measured gets improved."

— Robin S. Sharma [75]

How useful is GUIDE? This chapter evaluates its performance. Section 7.1 shows the model's results. Then, Section 7.2 experimentally analyzes the effect of GUIDE's components. Section 7.3 discusses the findings. Finally, Section 7.4 summarizes the discovered limitations and explains ways to overcome them.

7.1 RESULTS

This section presents GUIDE's results, using quantitative metrics, its learning curves, and qualitatively looking at the correctly and incorrectly predicted word-SDQ links. In addition, it visualizes an excerpt of the GNN's learned weight matrix and analyzes visible patterns.

7.1.1 *Empirical Evaluation*

Table 7.1 shows the evaluation results. To the best of our knowledge, there is currently no other automated approach to automatically linking words from low-resource languages to SDQs. Therefore, our baseline is a random baseline. For each word-SDQ pair, the random classifier predicts an existing word-SDQ link with a probability of 50%. There are $N = 199,605$ words in the MAG with 81,632 links to SDQs. Therefore, the random classifier predicts 741 million $(N \times 7,425/2)$ word-SDQ links of which 40,816 are correct. This ratio leads to a precision of 0.00006, a recall of 0.500, and an $F1$ score of 0.0001. GUIDE surpasses the random baseline in every respect except for recall.

The questionnaire-based evaluation reveals that GUIDE's precision is in fact almost twice as high as the dataset-based evaluation tells (0.65 instead of 0.34). The number of predicted links in the rightmost column correlates with the GUIDE's recall, which we could not directly measure using manual evaluation. Figure 7.1 shows the primary factor in improving GUIDE's recall is likely the amount of parallel text.

GUIDE predicted 71,094 word-SDQ links in total of which 19,166 (37%) belong to test languages. 31,873 (62%) of the links predicted for the de-

Table 7.1: Evaluation results for our twelve development and eight test languages: For each development language, cells with "±" show the average value of ten runs and the standard deviation. In the six bottom rows, "±" shows the average and the respective standard deviation. The six "average" rows show the average values for the development set, test set, and the languages tokenized with Stanza, SentencePiece, and punctuation mark split, respectively (see Subsection 5.2.1), as well as the average of all languages. The number of predicted word-SDQ links in the rightmost column is only from the run that we used to create the questionnaires. The number in parentheses is the number of links that are not already contained in the dataset.

	EVALUATION WITH DATASET			MANUAL	
Language	*Precision*	*Recall*	*F1*	*Precision*	# Predicted links
Random baseline	0.00	**0.500**	0.000	n/a	741,033,563
DEVELOPMENT					
Bengali	0.22 ± 0.11	0.002 ± 0.001	0.004 ± 0.003	0.56	2,809 (2,770)
Chinese (simplified)	0.17 ± 0.02	0.014 ± 0.002	0.026 ± 0.004	0.34	5,752 (**5,036**)
English	**0.63** ± 0.02	**0.125** ± 0.006	**0.208** ± 0.009	**0.86**	7,119 (2,314)
French	0.59 ± 0.03	0.097 ± 0.005	0.167 ± 0.008	0.78	6,993 (2,527)
Hindi	0.25 ± 0.02	0.029 ± 0.003	0.051 ± 0.006	0.78	3,914 (2,835)
Indonesian	0.34 ± 0.05	0.035 ± 0.005	0.064 ± 0.009	0.77	1,799 (1,068)
Kupang Malay	0.14 ± 0.05	0.013 ± 0.005	0.024 ± 0.009	0.79	1,440 (1,351)
Malayalam	0.10 ± 0.03	0.015 ± 0.004	0.026 ± 0.007	0.43	2,768 (2,480)
Nepali	0.20 ± 0.01	0.022 ± 0.002	0.039 ± 0.004	0.38	2,641 (2,156)
Portuguese	0.43 ± 0.02	0.088 ± 0.006	0.146 ± 0.009	**0.86**	6,759 (3,737)
Spanish	0.59 ± 0.02	0.090 ± 0.005	0.155 ± 0.008	0.84	**7,614** (3,579)
Swahili	0.33 ± 0.04	0.018 ± 0.003	0.033 ± 0.005	0.75	2,320 (2,020)
TEST					
German	n/a	n/a	n/a	0.67	**5,022**
Hiri Motu	n/a	n/a	n/a	0.62	1,190
Igbo	n/a	n/a	n/a	0.45	1,405
Mina-Gen	n/a	n/a	n/a	**0.80**	3,063
Motu	n/a	n/a	n/a	0.32	2,731
South Azerbaijani	n/a	n/a	n/a	0.58	2,238
Tok Pisin	n/a	n/a	n/a	0.69	880
Yoruba	n/a	n/a	n/a	0.63	2,637
AVERAGES					
Development set	0.33 ± 0.04	0.046 ± 0.004	0.079 ± 0.007	0.68 ± 0.19	4,327 ± 2,338
Test set	n/a	n/a	n/a	0.60 ± 0.15	2,396 ± 1,324
Stanza	**0.43** ± 0.02	**0.068** ± 0.005	**0.117** ± 0.008	**0.74** ± 0.17	**5,622** ± 1,975
SentencePiece	0.21 ± 0.05	0.014 ± 0.003	0.026 ± 0.005	0.53 ± 0.13	2,364 ± 524
Punctuation mark split	0.14 ± 0.05	0.013 ± 0.005	0.024 ± 0.009	0.64 ± 0.18	1,990 ± 927
Total	0.33 ± 0.04	0.046 ± 0.004	0.079 ± 0.007	0.65 ± 0.18	3,555 ± 2,180

velopment languages are new. Because the total number of matched words in the MAG is 199,605 (see Table 4.2), the model predicts one word-SDQ link per 2.8 words. Taking the model's precision of 0.65 into account, it predicts one correct word-SDQ link per 4.4 words. This num-

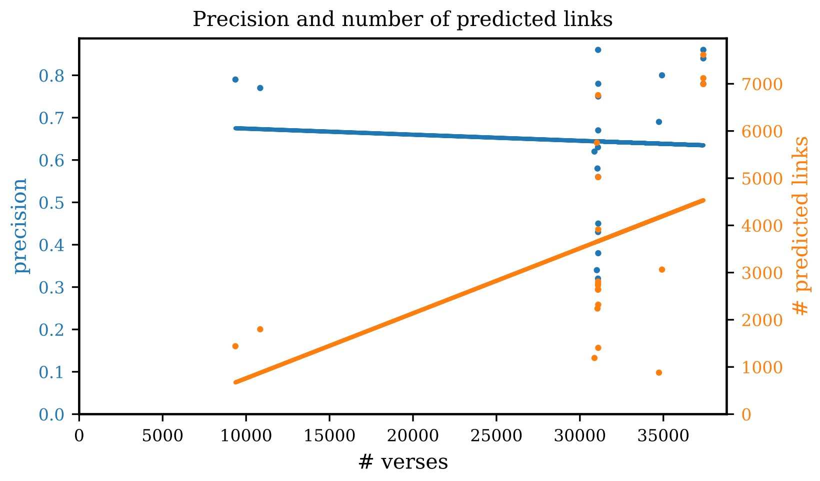


Figure 7.1: A summary of the precision and number of predicted word-SDQ links reported in Table 7.1: Each language has one dot for each of these two metrics. The regression lines indicate that GUIDE's recall increases with more parallel textual data, while its precision depends more on the language.

ber demonstrates GUIDE's few-shot learning capabilities.

As usual for multilingual NLP models, our model also performs best for Indo-European languages because seven of the twelve development languages come from this language family (see Table 4.1).

7.1.2 *Learning Curves*

The learning curves in the Appendix (see Section A.4) show that the model does not overfit because the validation loss does not increase. We also see that the training loss is only slightly lower than the validation loss, which further shows that the model generalizes well. Another observation is that the model starts by predicting too many word-SDQ links and learns to make fewer, but more precise predictions. The recall plummets at first, but flattens after the second epoch, while the precision slowly converges.

7.1.3 *Example Predictions*

Now we examine the reasons for true positive, false positive, as well as false negative predictions and demonstrate GUIDE's interpretability. Because the model has only a single GCN layer, it is easily interpretable:

Each word's SDQ links depend solely on its directly aligned words.

7.1.3.1 *True positives*

This part explains two examples of correctly predicted word-SDQ links. By looking at the 1-hop neighborhood of words for which GUIDE predicted an SDQ link, the word alignments that caused this prediction become clear (i.e., there is no *"neural black magic"*). Figure 7.2 and Figure 7.3 demonstrate this interpretability for two words from African low-resource languages.

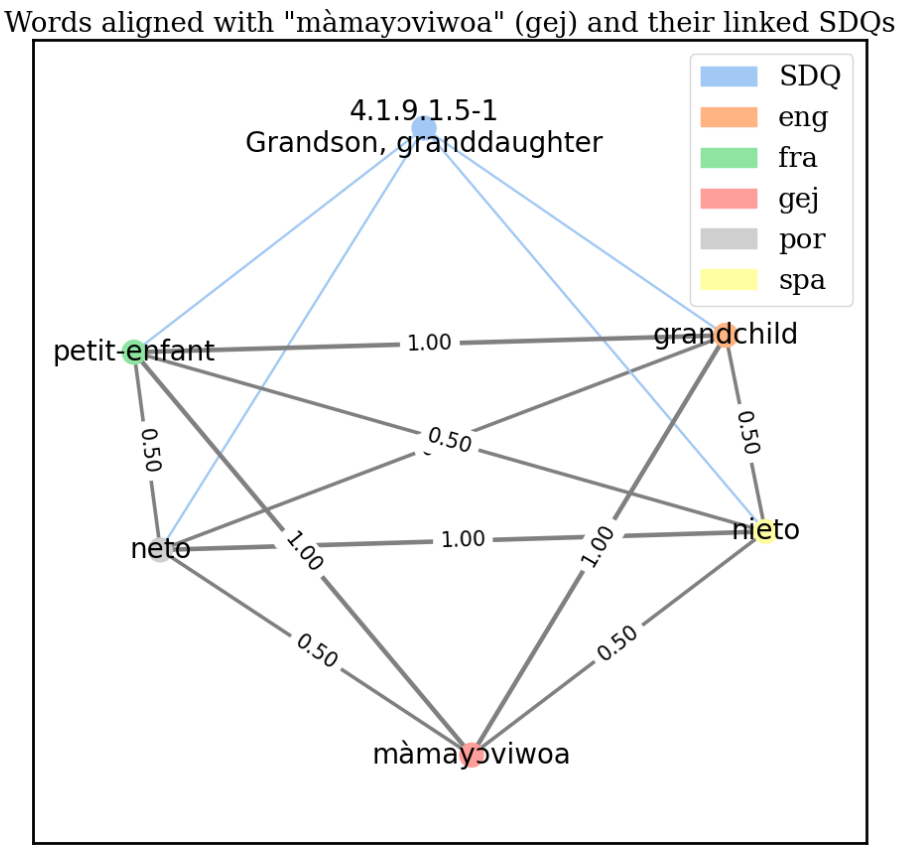


Figure 7.2: The subgraph shows why GUIDE correctly links the Mina-Gen word *"màmayɔviwoa"* to the SDQ 4.1.9.1.5-1: *"What words refer to the children of your children?"*. The word has four strong edges to neighbors that are known to belong to the same SDQ. The Mina-Gen speaker who answered the questionnaire verified this prediction and noted that *"màmayɔviwoa"* refers only to a woman's grandchildren. Its male counterpart is missing in the data.

The same restrictions apply to this figure as noted in the caption of Figure 5.1.

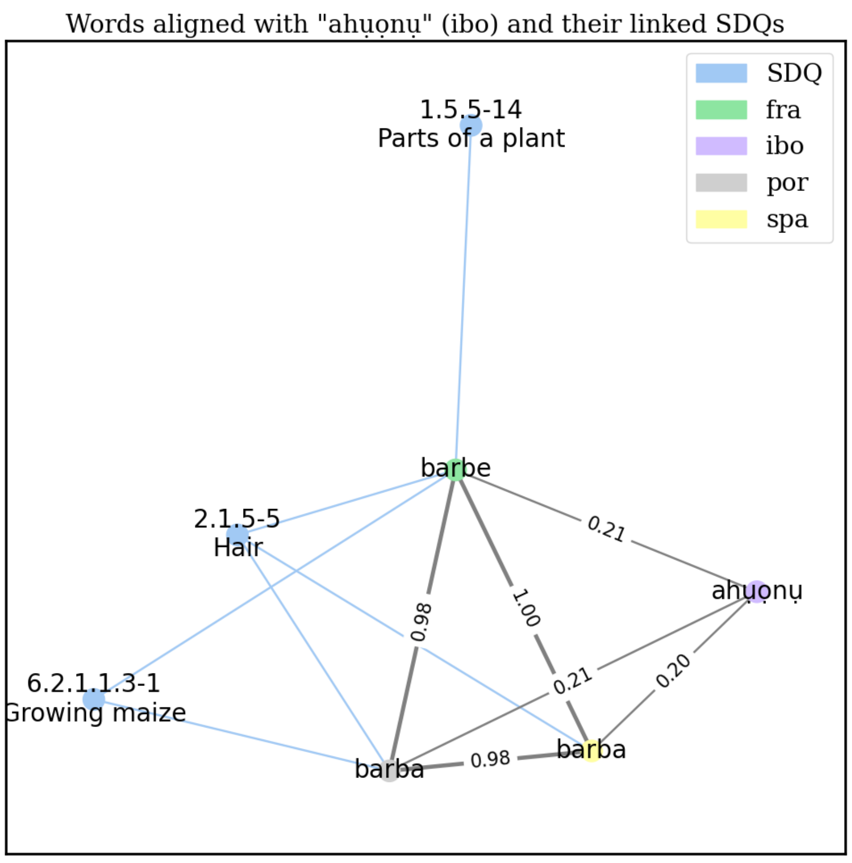


Figure 7.3: The subgraph shows why GUIDE correctly links the Igbo word *"ahụọnụ"* (beard) to the SDQ 2.1.5-5: *"What words refer to hair on the face?"*. The word has three edges to neighbors that are known to belong to the same SDQ. However, GUIDE correctly did not link *"ahụọnụ"* to the other two SDQs that are linked to the word's neighbors because one or two edges with a weight of 0.21 are not strong enough. The Igbo speaker who answered the questionnaire verified that *"ahụọnụ"* belongs to 2.1.5-5 and not the other two SDQs.

The same restrictions apply to this figure as noted in the caption of Figure 5.1.

7.1.3.2 *False positives*

This part explains two examples of incorrectly predicted word-SDQ links. Figure 7.4 and Figure 7.5 show that the alignment of words with ambiguous words is one of the main reasons for such misclassifications.

Regarding Figure 7.4, *"overdrive"* appears only in a single verse in the used English Bible: *"the flocks and herds with me have their young, and if they overdrive them one day, all the flocks will die."* (Genesis 33:13) An-

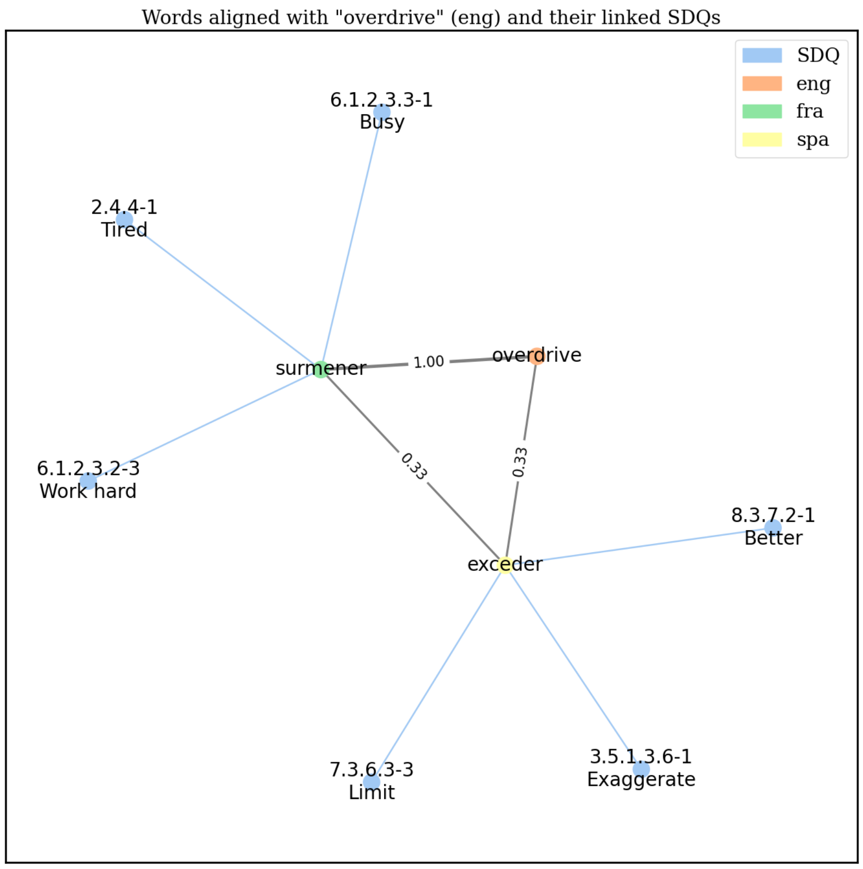


Figure 7.4: The subgraph shows why GUIDE incorrectly links the English word *"overdrive"* to the SDQ 6.1.2.3.2-3: *"What words describe someone who works too hard?"*. *"overdrive"* is ambiguous and aligns with the French word *"surmener"* (overwork) that belongs to this SDQ. The native English speaker who answered the questionnaire confirmed that this link is a false positive. Although *"overdrive"* can also be a metaphor for *"overwork"*, it would be an unusual word choice in this context. The same applies to the SDQs 2.4.4-1 (*"What words describe someone who is tired?"*) and 6.1.2.3.3-1 (*"What words describe a person who is busy and has a lot to do?"*). The subgraph shows that the Eflomal also aligns words with overlapping but different meanings, which leads to false positives. Consequently, the GNN fails to capture their nuanced semantic differences. However, the GNN correctly does not link *"overdrive"* to any of the SDQs linked to the Spanish word *"exceder"* because a single edge with a weight of 0.33 is too weak.

The same restrictions apply to this figure as noted in the caption of Figure 5.1.

other valid but more natural-sounding word choice in the context of driving animals would be *"driven hard"*. By tracing back the misclassification to this verse, we see that nowadays uncommon word choices

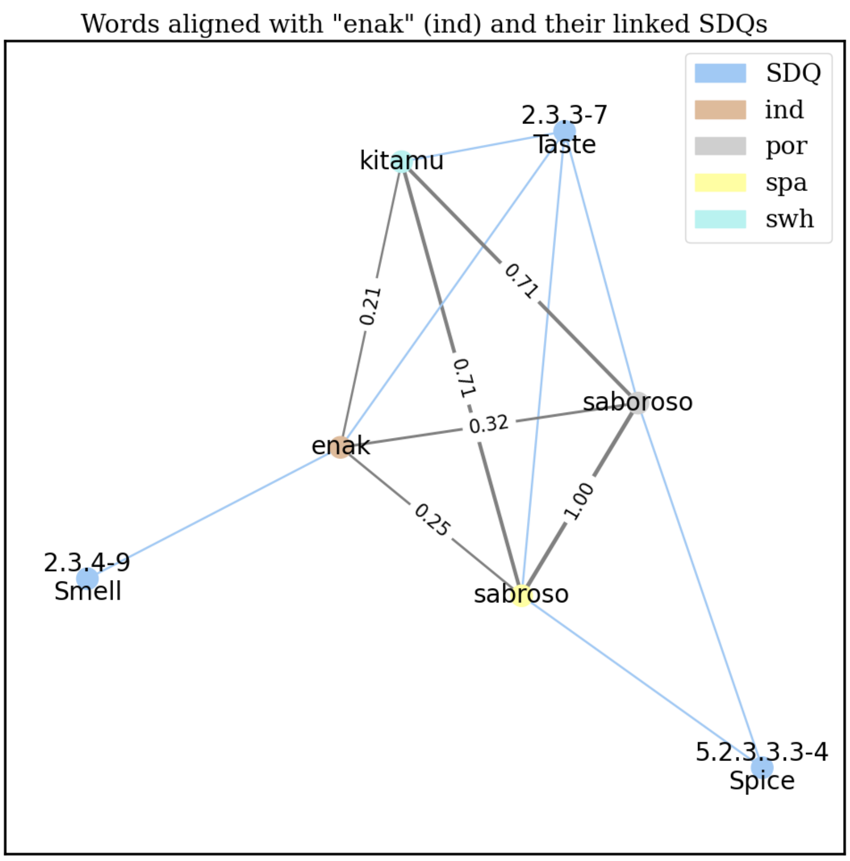


Figure 7.5: The subgraph shows why GUIDE incorrectly links the Indonesian word *"enak"* (delicious) to the SDQ 5.2.3.3.3-4: *"What words describe food to which spice has been added?"*. *"enak"* has two neighbors that belong to this SDQ. The Indonesian speaker who answered the questionnaire confirmed that this link is a false positive. The Spanish word *"sabroso"* and the Portuguese word *"saboroso"* mean *"flavorful"* or *"tasty"* in English. This ambiguity causes misleading word alignments.

The same restrictions apply to this figure as noted in the caption of Figure 5.1.

add noise to the data, especially if they occur just once in the data.

Another reason for false positives not shown in the graphs is that the tokenizer produces tokens that are no actual words, and also do not become such by lemmatization. These tokenization errors occur especially often in Chinese. For example, the tokens "例中", "吸风", and "这棕" are combinations of arbitrary Chinese characters, according to the native Chinese speaker who answered the questionnaire.

7.1.3.3 *False negatives*

Now we show two examples of missing word-SDQ links in the predictions. Figure 7.6 and Figure 7.7 show that missing or too rare word alignments are a major reason for such misclassifications.

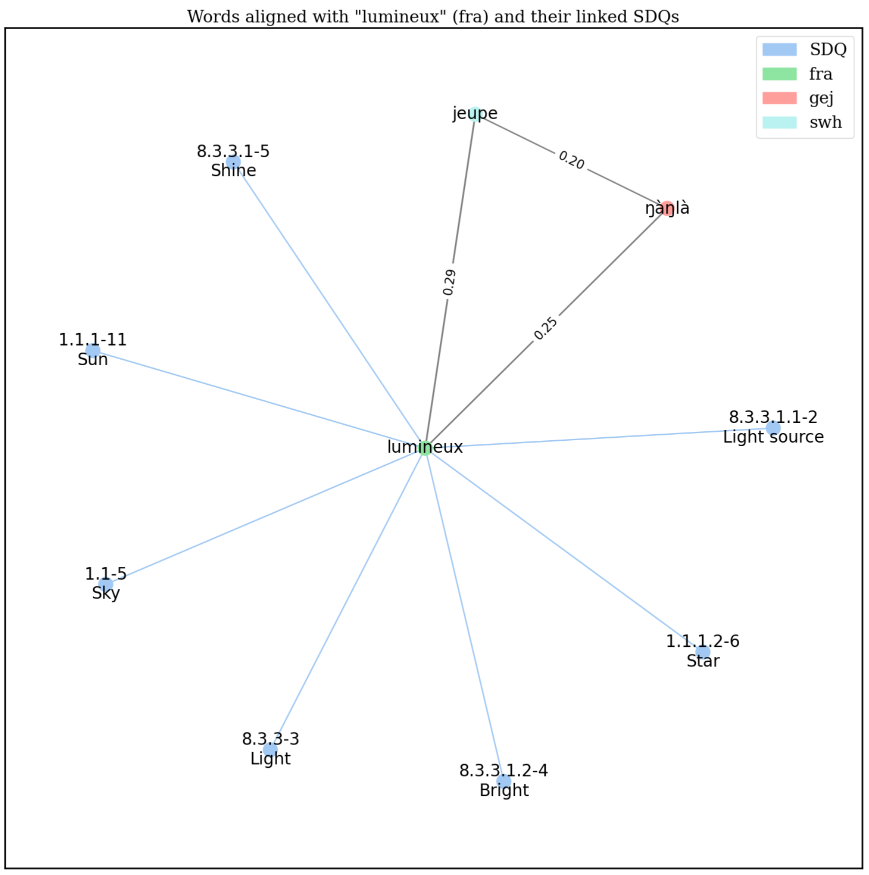


Figure 7.6: The subgraph shows why GUIDE misses linking the French word *"lumineux"* (luminous) to any of the seven displayed SDQs, such as 8.3.3-3: *"What words describe something that has light shining on it?"*. *"lumineux"* has two neighbors, and none of them has a link to these SDQs. The Swahili word *"jeupe"* (white) has a different meaning, and the Mina-Gen word *"ŋàŋlà"* belongs to the unlabeled test languages. Without labeled aligned words, GUIDE does not predict a word-SDQ link.

The same restrictions apply to this figure as noted in the caption of Figure 5.1.

7.1.4 *Weight Visualization*

This part explains GUIDE's decisions and learning process by looking at its weight matrix, which shows notable patterns.

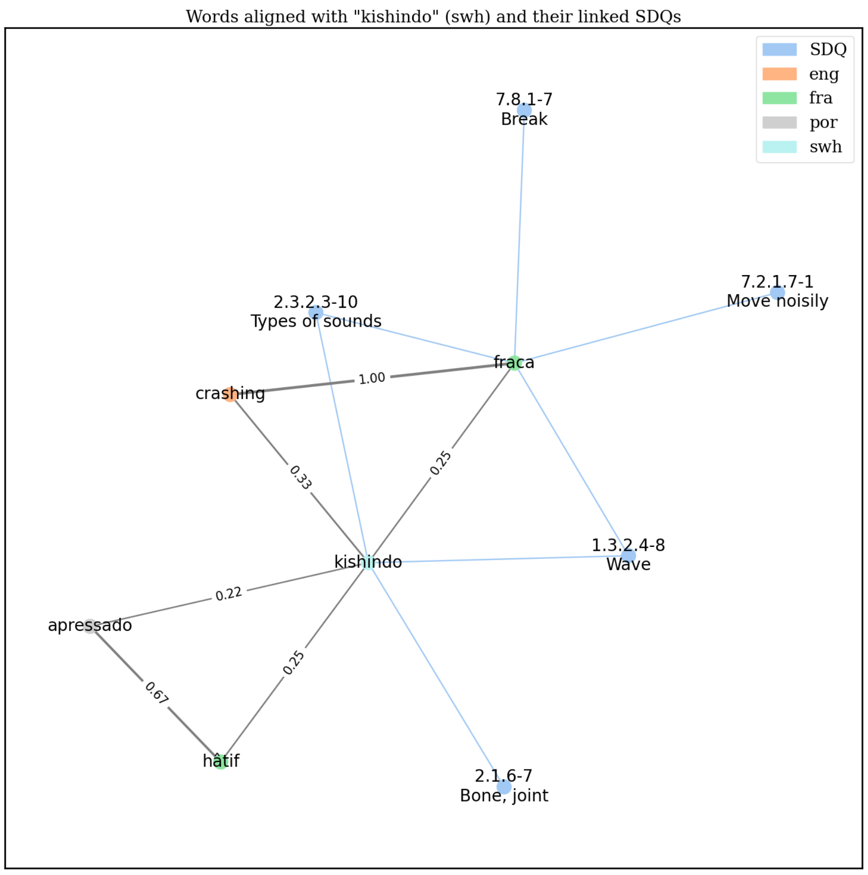


Figure 7.7: The subgraph shows why GUIDE misses linking the Swahili word *"kishindo"* (roar) to any of the three SDQs it is linked to in the development dataset, such as 1.3.2.4-8: *"What sounds do waves make?"*. *"kishindo"* has only one neighbor linked to this SDQs: the French word *"fraca"* (smash). Although the weak edge weight of 0.25 prevents the GNN from predicting the link to two correct SDQs links, it also prevents it from predicting two incorrect SDQs links (7.2.1.7-1: *"What words refer to making a noise while moving?"* and 7.8.1-7: *"What sounds do things make when they break?"*). We consider this behavior advantageous, as we favor precision over recall.

The same restrictions apply to this figure as noted in the caption of Figure 5.1.

Figure 7.8 shows that the model learned positive and negative correlations of SDQ links. As shown in Table 7.2, GUIDE detects overlapping semantics even if they are subtle and the SDQs are phrased quite differently. For example, the model learned the topmost correlation from the Portuguese word *"render"*, which means *"yield"* (linked to SDQ 4.8.3.4-1) or *"render"* (linked to SDQ 3.2.2.3-1) in English and also aligns with these two words. The German word *"ergeben"* (surrender / yield / to result in) also illustrates this semantic overlap.

Table 7.2: The five most strongly correlated non-identical SDQs in the learned weight matrix: GUIDE learned that a word probably belongs to the SDQ on the right if its neighbors belong to the respective SDQ on the left.

Weight	Input SDQ	Output SDQ
7.5	4.8.3.4-1: What words refer to surrendering to an enemy?	3.2.2.3-1: What words refer to the process of determining the truth of something?
7.4	8.4.6.5.5-1: What words describe a modern machine or system?	8.4.6.3-1: What words refer to the present?
6.1	3.2.8-1: What words indicate that the speaker thinks that something tends to be a certain way?	3.3.1.4-2: What words refer to what is intended?
5.9	3.1-5: What words refer to the part of a person that affects his behavior?	3.3.1.4-2: What words refer to what is intended?
5.7	3.2.3.1-2: What words describe something or someone that is known to many people?	3.3.1.2-1: What words refer to choosing something?

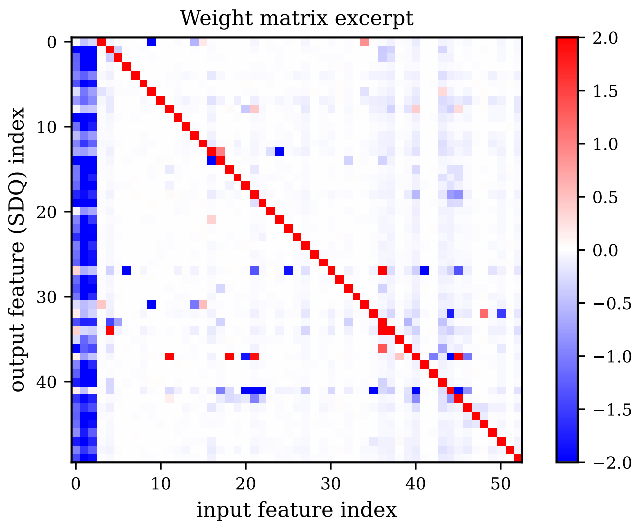


Figure 7.8: A heatmap showing an excerpt of the final weight matrix, showing the first size 53 × 50 weights. The red diagonal is the result of the modified identity matrix initialization (see Subsection 5.3.3), which sets each weight to 50.0 that corresponds to the same SDQ link feature in the GNN layer's input and output. The thick blue bar on the left corresponds to the three node features that are not SDQ link features. The single red weights show positively correlated SDQs while single blue weights show negatively correlated SDQs. The class imbalance (see Section 4.3) leads to a banded structure. Stronger bands indicate a stronger presence of an SDQ in the dataset. Note that the color scale has been clipped to improve interpretability (i.e., there are weights lower than -2.0 and greater than 2.0).

7.2 EXPERIMENTS

In this section, we experimentally analyze the effect of GUIDE's components by removing them, replacing them, and adding new ones. Although manual evaluation is more reliable than the dataset-based evaluation (see Subsection 7.1.1), we evaluate the experiments using the incomplete dataset because it is less costly and allows the measurement of the recall.

7.2.1 *Ablation Experiments*

This section shows how GUIDE's performance changes when components are removed. We investigate the usefulness of the preprocessing steps (Section 5.2), the node features (Subsection 5.3.1), and modified identity matrix initialization (Subsection 5.3.3), as shown in Table 7.3.

The ablations reveal that the removal of the SDQ link node feature reduces the model's performance to almost zero. Another observation

Table 7.3: Changes in GUIDE's performance for 13 ablations: Cells with "±" show the average value of three runs and the standard deviation. Deactivating the Stanza pipeline and SentencePiece tokenization means that we used tokenization by punctuation mark split instead (see Figure 5.3). "n/a" means that the execution failed because it consumed too much VRAM.

	ΔPrecision	*ΔRecall*	*ΔF1*
GUIDE (reference values)	0.33 ± 0.04	0.046 ± 0.004	+0.079 ± 0.007
PREPROCESSING			
¬ Stanza pipeline	-0.01 ± 0.04	-0.017 ± 0.005	-0.027 ± 0.008
¬ MWT expansion	+0.02 ± 0.03	-0.001 ± 0.003	-0.001 ± 0.006
¬ Lemmatization	+0.00 ± 0.03	-0.016 ± 0.003	-0.025 ± 0.006
¬ SentencePiece tokenization	-0.00 ± 0.04	-0.005 ± 0.003	-0.008 ± 0.006
¬ Lowercasing	+0.01 ± 0.05	-0.001 ± 0.005	-0.002 ± 0.008
¬ Edge weight normalization	n/a	n/a	n/a
¬ Edge weight filtering	n/a	n/a	n/a
¬ Isolated node removal	-0.02 ± 0.03	+0.013 ± 0.006	+0.019 ± 0.009
NODE FEATURES			
¬ Degree	-0.00 ± 0.04	+0.012 ± 0.005	+0.016 ± 0.007
¬ Weighted degree	+0.01 ± 0.04	+0.007 ± 0.004	+0.010 ± 0.007
¬ SDQ count	+0.02 ± 0.04	+0.001 ± 0.004	+0.002 ± 0.007
¬ SDQ links	-0.33 ± 0.00	-0.045 ± 0.000	-0.077 ± 0.001
OTHER			
¬ Modified identity matrix initialization	-0.05 ± 0.07	-0.038 ± 0.001	-0.063 ± 0.003

is that the modified identity matrix initialization helps the optimizer reach a deeper local minimum in the loss space. From the ablated components, these two have the strongest influence on the $F1$ score.

Interestingly, four components harm the model's $F1$ score: the isolated node removal and the other three components of the node feature vector. The performance drop indicates that the additional features do not correlate with the node's actual SDQ links.

For the ablation of edge weight normalization and edge weight filtering, the execution of the GNN crashes because the larger graph contains more edges than the GPU can hold. Furthermore, the ablation of the edge weight normalization also shows that it is necessary for successive edge weight filtering because the absolute weights before normalization are ≥ 1, which is far above the filter threshold of 0.2.

7.2.2 *Parameter Tuning*

Now, we look at the effect of two parameters on the model's performance: the training split size and the threshold.

EFFECT OF TRAINING SPLIT SIZE We expect that the model's performance decreases when we do not use a random node split of 80%/10%/10% but $x\%/10\%/(90\%-x\%)$ with $x < 80$. Figure 7.9 confirms this assumption, although $x = 0.1$ is only slightly less precise than $x = 80$. As seen in Figure 7.9, the precision increases slightly until around 50% of the words; then it remains mostly unchanged. This finding surprised us, as it shows that the GNN even works with tiny amounts of training data. This behavior is likely an effect of the modified identity matrix initialization, which anticipates much of the learning process. A visible confirmation of this effect is the weight matrix shown in Figure 7.10, which is quite similar to the weight matrix learned with more training data displayed in Figure 7.8. However, this experiment alone does not show that the model also works with low amounts of parallel text and only sparse semantic domain dictionaries.

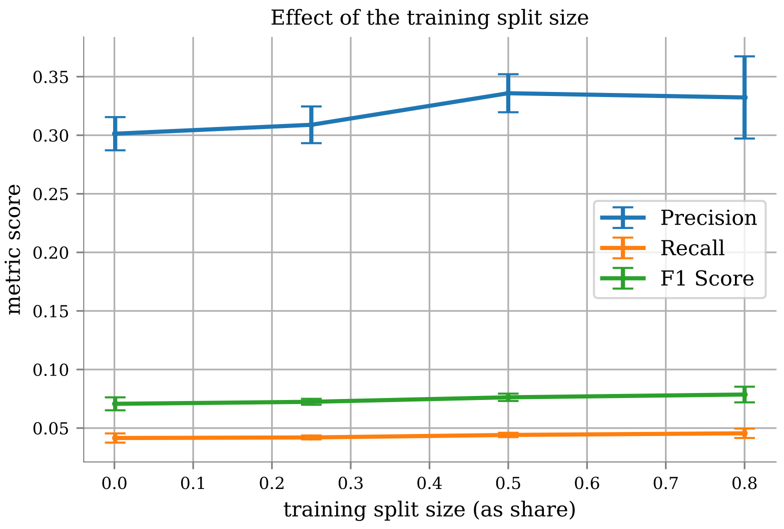


Figure 7.9: The model's performance is almost the same for a training split size of 0.1% (leftmost dots) and 80% of the MAG's words. The error bars show the standard deviation (from three runs). The standard deviation increases with an increasing amount of training data likely because it is computed on the test split, which becomes smaller. The validation split is constant (10% of the MAG's words), and the test split is the remaining nodes minus the training split.

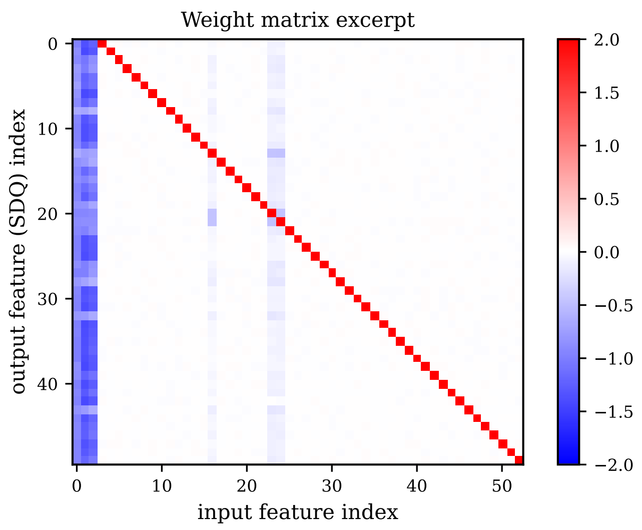


Figure 7.10: A heatmap showing an excerpt (the first size 53 × 50 weights) of the learned weight matrix, using only 0.1% of the dataset as training data. The blue columns in the middle correspond to the SDQs that appeared in the smaller training data split. Note that the color scale has been clipped to improve interpretability (i.e., there are weights lower than -2.0 and greater than 2.0).

EFFECT OF THE THRESHOLD We expect that a higher threshold leads to higher precision and lower recall. Figure 7.11 confirms this assumption. Because we prefer precision over recall, we even accept a lower $F1$ score to obtain cleaner dictionaries.

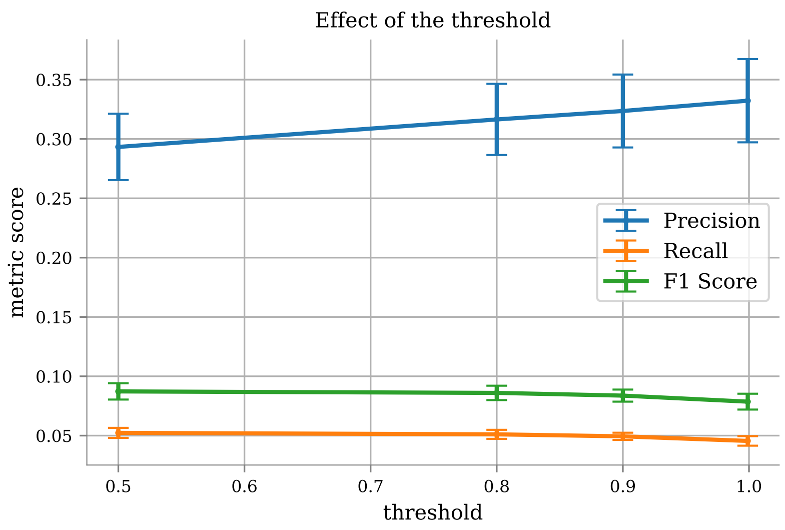


Figure 7.11: GUIDE uses a threshold of 0.999 (rightmost dots). The error bars show the standard deviation (from three runs).

7.2.3 *Other Experiments*

Now, we show how the model's performance changes when components are replaced or added. Table 7.4 shows the summary, and we discuss each of the experiments in the following. The only experiment that has led to an improved performance is the addition of NFKD normalization.

Table 7.4: Changes in GUIDE's performance for seven experiments, evaluated on the development dataset: Cells with "±" show the average value of three runs and the standard deviation. "+" shows an experiment with a component that has been added to GUIDE. "∗" shows an experiment in which one of GUIDE's components has been replaced.

	Δ*Precision*	Δ*Recall*	Δ*F1*
GUIDE (reference values)	0.33 ± 0.04	0.046 ± 0.004	0.079 ± 0.007
PREPROCESSING			
+ Heuristic stemming	+0.01 ± 0.03	-0.001 ± 0.003	-0.002 ± 0.005
+ NFKD normalization	+0.02 ± 0.04	+0.002 ± 0.004	+0.003 ± 0.006
∗ Intersection heuristic	-0.00 ± 0.04	-0.003 ± 0.003	-0.004 ± 0.006
+ Stop-word removal	+0.01 ± 0.04	-0.002 ± 0.003	-0.003 ± 0.006
GNN			
∗ DNAConv layer (attention)	-0.11 ± 0.05	-0.038 ± 0.001	-0.064 ± 0.002
+ 2nd GCN Layer	-0.31 ± 0.04	-0.044 ± 0.001	-0.077 ± 0.001
∗ Binary cross-entropy loss	-0.33 ± 0.00	+0.078 ± 0.009	-0.076 ± 0.000

EFFECT OF STEMMING When there is no linguistic knowledge about a language available, a statistical, corpus-based stemming approach could be an effective alternative to rule-based stemming. [99] We implemented a heuristic, language-independent stemmer to detect and remove common suffixes in Kupang Malay. The reason for this choice is that this language is the only language in the development dataset that we tokenize by splitting words at punctuation marks (see Subsection 5.2.1). Tokens processed by the Stanza pipeline are already lemmatized, and tokens generated with SentencePiece are often stripped from their suffixes. Therefore, these languages do not require additional stemming.

The stemmer builds a *suffix trie* to learn the suffixes and then removes as many as possible from each token, similar to the approach of Lyras, Sgarbas, and Fakotakis [60]. We also stem the words in the semantic domain dictionaries in the same languages so that they match the stemmed words from the Bible translations. In contrast to the lemmatization, we lowercase before stemming because we assume that the stemmer benefits from normalized characters.

The evaluation shows that while the heuristic stemming slightly increases precision, the recall and $F1$ score decrease. Thus, we decided not to use the stemmer.

EFFECT OF NFKD NORMALIZATION In addition to lowercasing, we analyzed the effect of *Normalization Form Compatibly Decomposition* (NFKD) normalization [95], a Unicode normalization method that unifies characters that can be encoded in multiple ways. It also removes diacritics. NFKD normalization slightly increases GUIDE's precision and recall, indicating that there are words that belong to the same SDQs and are written differently. However, a drawback of NFKD normalization is that the generated dictionary entries could differ from their original spellings, similar to the effect of stemming.

EFFECT OF INTERSECTION HEURISTIC A common alternative to the GDFA heuristic (see Subsection 5.2.5) is the intersection heuristic to create bilingual word alignments. The intersection heuristic keeps only alignment edges that the word aligner found for the language pairs $A - B$ and $B - A$. We assumed that this stricter filtering would increase the model's precision while decreasing its recall. However, only the latter happened. A possible explanation for the missing increase in precision is that the edge weight filtering removes almost the same alignment edges.

EFFECT OF STOP-WORD REMOVAL A common NLP method to remove noise from textual data is stop-word removal. Normally, stop-word removal is done via predefined stop-word lists, which often are not available for low-resource languages. We implemented a heuristic stop-word removal system by removing words that often align with English stop-words. As expected, stop-word removal increases the model's precision, while it decreases its recall. These results indicate that stop-words are also often linked to SDQs in the development dataset. Because the increase in precision is low, we decided not to "sacrifice the minority for the majority".

EFFECT OF ATTENTION Attention [92] makes the model more flexible about the patterns it can learn. Therefore, we replaced the GC-

NConv layer with a *DNAConv* layer[1] [30]. We chose the DNAConv layer because it also supports edge weights. Because this transformer layer has three weight matrices instead of one, we do not apply modified identity matrix initialization. Table 7.4 shows that using the DNAConv layer results in an even larger decrease in precision as omitting only modified identity matrix initialization (see Table 7.3). This performance drop indicates that the amount of training data is too low to learn the higher number of parameters.

EFFECT OF 2ND GCN LAYER If each node "sees" not only its 1-hop but also 2-hop neighbors, the recall could drastically increase because the model has more information about its potential SDQs. Therefore, we added a *relu* activation function after the first layer and added a 2nd GCNConv layer of size 7,425 × 7,425. While we still apply modified identity matrix initialization to the first layer, we apply normal identity initialization to the second layer (i.e., with a weight of 1.0 instead of 50.0 on the diagonal) because high weights would amplify the first layer's output too much. Furthermore, we initialize the bias vector of the second layer with -1.0. Additional adjustments we made are the reduction of the batch size from 6,000 to 1,000 because the VRAM does not suffice otherwise. To prevent data leakage, we mask the words for which the GNN is predicting the SDQ links without masking its neighboring words. Otherwise, a word could simply copy its own SDQ links because it is its own 2-hop neighbor. Shockingly, the additional GCN layer makes the performance drop to almost zero. This result also indicates that the amount of training data is too low to learn the higher number of parameters.

EFFECT OF LOSS FUNCTION We use soft $F1$ loss (see Subsection 5.3.4) and also experimented with *binary cross-entropy* [62] loss, which is suited for multi-class multi-label classification if each sample's classes are represented as a multi-hot vector, which is the case for GUIDE. Because these vectors are sparse, the ratio of zeros and ones is imbalanced. Therefore, we normalize the loss function by assigning higher weights to positive word-SDQ links. The change of the loss function results in a model that makes far more predictions, of which the vast majority are wrong. In conclusion, the soft $F1$ loss is more suitable for the learning goal, which is a high $F1$ score.

[1]PyTorch Geometric documentation: https://pytorch-geometric.readthedocs.io/en/latest/generated/torch_geometric.nn.conv.DNAConv.html (visited on 16/10/2023)

7.3 DISCUSSION OF THE RESULTS

This section discusses the most important findings from the results and the above experiments. GUIDE's (manually evaluated) precision of 0.65 in comparison to the (dataset-based) recall of 0.046 shows that the model predicts mostly correct word-SDQ links but it creates only fractions of complete semantic domain dictionaries. Nevertheless, the recall is likely to be higher in practice because the evaluation with the incomplete dataset fails to recognize true positive predictions. Because the manually evaluated precision is twice as high as the dataset-based precision, we assume that the same holds for the actual recall, which is about 0.09. These adjusted metric scores result in an $F1$ score of 0.16. While GUIDE cannot replace linguists who compile semantic domain dictionaries, it can provide an initial dictionary with thousands of entries, of which most are correct. Furthermore, the experiments show that GUIDE's performance can be improved by three measures:

- removing all node features but the SDQ link feature,
- omitting isolated node removal, and
- applying NFKD normalization.

The next section discusses additional methods to make GUIDE even more useful.

7.4 LIMITATIONS

To conclude the evaluation chapter, we discuss the limitations of our approach across multiple components as well as ways to overcome them.

7.4.1 *Computational Limitations*

The node feature matrix is a memory bottleneck. It is saved as a dense vector of size $N \times 7,428$, where N is the number of nodes/words. The model allocates approximately 3.5 GB of VRAM per language. Therefore, 48 GB of VRAM (see Section 6.1) limit us to loading approximately 13 languages. Therefore, we cannot load the entire MAG of 20 languages at once but load a subgraph of the twelve development languages plus only a single test language. This approach does not affect the quality of the results because the test languages do not have labeled data and cannot learn word-SDQ mappings from each other. A possible solution is to use a sparse feature matrix.

7.4.2 *Dataset*

The used Bible translations and semantic domain dictionaries cause various limitations that we discuss in the following.

BIBLE TRANSLATIONS The general challenge of using only the Bible as parallel data is the narrow domain [27]. The Bible does not include all the words used in today's world, particularly those related to technology, science, and current culture, such as "*smartphone*". The language in the Bible is often elevated and does not reflect the way people talk in everyday life (e.g., by using slang and idioms). Although different Bible translations convey the same meaning, they differ in their proximity to the original text (in Hebrew, Aramaic, and Greek). While some are literal translations, others paraphrase a lot to be understandable to a modern audience. These different approaches to Bible translation cause noise in the word alignments.

Possible solutions are:

- USING MORE PARALLEL DATA (e.g., the Bloom Books corpus (see Subsection 2.2.2). Alwosheel, Van Cranenburgh, and Chorus [4] mention a common rule of thumb is that to train a neural network with p parameters, a dataset needs $10 \times p$ samples. They even propose an even more conservative heuristic of using at least $50 \times p$ samples. We have 55 million parameters but only 150,000 samples (i.e., nodes) in the development dataset. According to this rule, we would need at least 18,000 times more words. This rule also explains why our model cannot learn unless we apply modified identity matrix initialization, which anticipates much of the learning process.
- USING MORE BIBLE TRANSLATIONS Currently, we use only one Bible translation per language, although some languages have multiple translations in the eBible corpus. By using multiple translations per language, we can match more words with SDQs and also increase Eflomal's performance. [39] use this approach and explain that if language A has two and language B has three different translations, we can use six pairs of aligned translations.

SEMANTIC DOMAIN DICTIONARIES The semantic domain dictionaries are incomplete (see Section 4.4). They cover a part of the languages' vocabularies and are also missing SDQ links for the words they cover. This limitation is the nature of language data because living languages are constantly evolving. They receive new words and new meanings for existing words. However, we found only a handful of incorrect SDQ links in our training data Section A.3. A possible solution for this impairment is GUIDE itself. In Section 8.1, we explain how the tool can be used to incrementally fill the dictionaries in an active learning [86] environment.

7.4.3 *Preprocessing Pipeline*

As shown in the error analysis in Subsection 7.1.3, the preprocessing pipeline produces a MAG that contains misleading edges and lacks useful edges and nodes. We now discuss three reasons for these limitations.

AMBIGUITY Word are often ambiguous (e.g., "*overdrive*") and thus align to different words in another language. The preprocessing pipeline treats them as if they are the same word, which confuses semantic patterns in the MAG, leading to misclassifications. A possible solution would be to use *Word-Sense Disambiguation* (WSD) [56] for as many languages as currently possible.

NOISY ALIGNMENTS There is a lot of noise in word alignments because we train Eflomal on a small corpus that contains many words only once. We mitigate this noise by aggregating all alignments from all languages. A possible solution is to remove words from the MAG that occur less than three times. However, this filtering would further reduce the model's recall.

COLLOCATIONS We ignore most word groups (so-called collocations [87], e.g., "*harvest moon*") in the semantic domain dictionaries unless Stanza provides an MWT expansion model for their language. A possible solution is to chunk tokens by their alignments to words in other languages before aggregating the alignments. For example, the German word "*Vollmond*" aligns with the English words "*harvest*" and "*moon*", which are right next to each other. A common pattern, such as this one, allows us to conclude that "*harvest moon*" is a collocation.

7.4.4 *Node Features*

Although three node features turned out to harm the model's performance, it could also ignore other potentially useful node properties. Imani Googhari et al. [41] enrich their node feature vectors with node centrality [13] and node community [18, 20] features. They found that the centrality feature contributed to approximately 80% of their model's $F1$ score, which aligns words. A possible solution is to try out these features, too.

Another possible solution is to leverage language information, adding a tree-structured graph to the MAG that contains languages and their linguistic and geographic relations. Subsequently, this additional information would allow us to find cognates in closely related languages (e.g., *"diari"* (light) in Motu and Hiri Motu). Possible data sources are the Ethnologue [26] and *World Atlas of Language Structures* (WALS) [32]. Leveraging orthography is especially helpful in creating a semantic domain dictionary for a language that is related to a language with an existing dictionary.

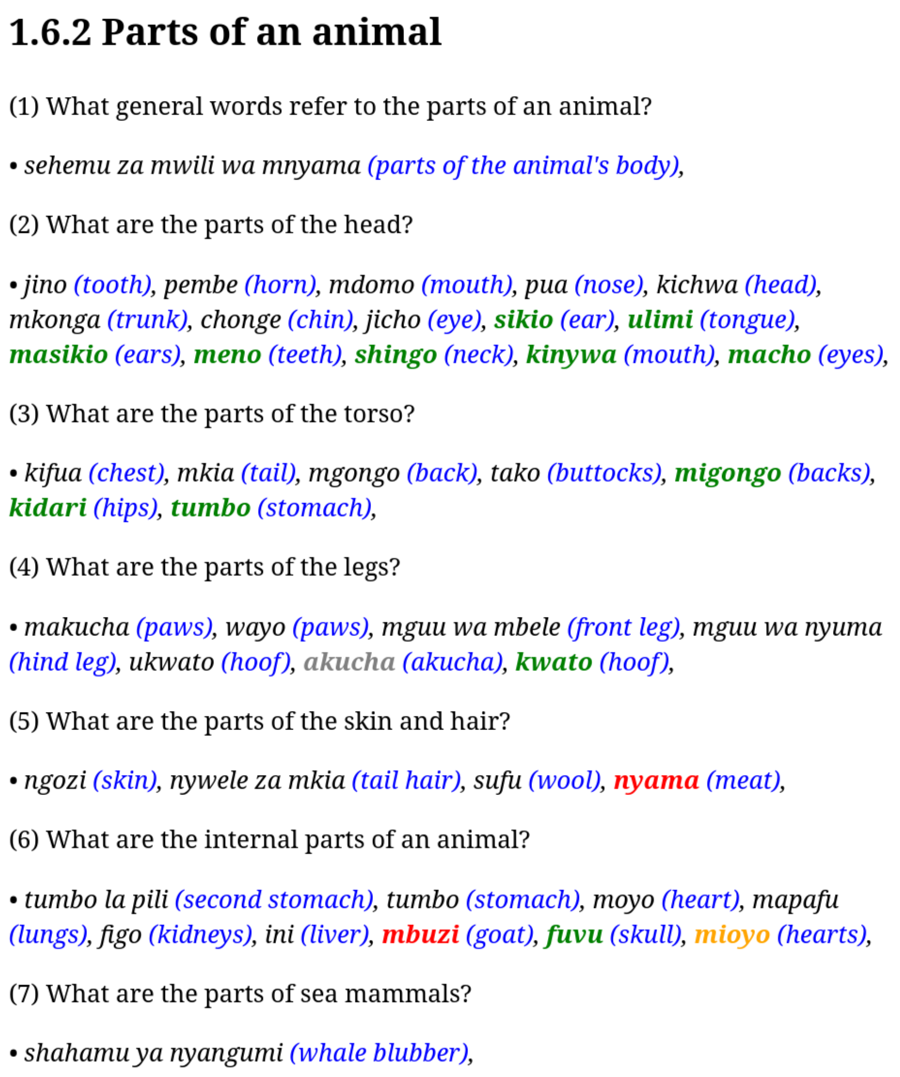
Even languages from different language families may have some similarities in orthography (e.g., proper names and loanwords, such as *"kristal"* (crystal) in Indonesian). Therefore, the use of orthography would be potentially beneficial even without additional language information.

7.4.5 *Predictions*

The last area of limitations that we examine is the quality of the predictions. In addition to the previously discussed issue of false positive and false negative predictions, they can also be redundant, as shown in Figure 7.12.

GUIDE adds redundant dictionary entries for languages without lemmatization. While these different lemmas are often valid answers and useful training data, they are not needed from the perspective of a native speaker. Our heuristic, language-agnostic stemmer (see paragraph 7.2.3) was an attempt to mitigate this issue. However, stems are also often undesirable dictionary entries (e.g., *"huft"* (see paragraph 5.2.2)). Therefore, it is necessary to manually review the predicted word-SDQ links, ideally asking native speakers. These people can easily name the word's base form and remove false entries. We describe this approach in more detail in the next chapter (see Section 8.1).

[2]translated by Google Translate: `https://translate.google.com/` (visited on 13/10/2023)



1.6.2 Parts of an animal

(1) What general words refer to the parts of an animal?

• *sehemu za mwili wa mnyama (parts of the animal's body),*

(2) What are the parts of the head?

• *jino (tooth), pembe (horn), mdomo (mouth), pua (nose), kichwa (head), mkonga (trunk), chonge (chin), jicho (eye),* ***sikio*** *(ear),* ***ulimi*** *(tongue),* ***masikio*** *(ears),* ***meno*** *(teeth),* ***shingo*** *(neck),* ***kinywa*** *(mouth),* ***macho*** *(eyes),*

(3) What are the parts of the torso?

• *kifua (chest), mkia (tail), mgongo (back), tako (buttocks),* ***migongo*** *(backs),* ***kidari*** *(hips),* ***tumbo*** *(stomach),*

(4) What are the parts of the legs?

• *makucha (paws), wayo (paws), mguu wa mbele (front leg), mguu wa nyuma (hind leg), ukwato (hoof),* ***akucha*** *(akucha),* ***kwato*** *(hoof),*

(5) What are the parts of the skin and hair?

• *ngozi (skin), nywele za mkia (tail hair), sufu (wool),* ***nyama*** *(meat),*

(6) What are the internal parts of an animal?

• *tumbo la pili (second stomach), tumbo (stomach), moyo (heart), mapafu (lungs), figo (kidneys), ini (liver),* ***mbuzi*** *(goat),* ***fuvu*** *(skull),* ***mioyo*** *(hearts),*

(7) What are the parts of sea mammals?

• *shahamu ya nyangumi (whale blubber),*

Figure 7.12: A semantic domain in the Swahili dictionary with already known words (black), added words (green, red, orange, gray), and English translations[2] (blue): Based on the translations, we manually labeled whether the entry is correct (green), incorrect (red), unknown (gray), or redundant (orange). The word "*mioyo*" is a duplicate because it is the plural form of "*moyo*". We do not automatically detect and remove such redundant entries because we do not lemmatize Swahili words.

8 OUTLOOK

"The best way to predict the future is to invent it."

— *Alan Kay* [14]

In this chapter, we show the use cases for GUIDE and suggest three future directions in more detail. To the best of our knowledge, this is the first successful approach to automatically link words to SIL's SDQs. This opens up a multitude of possible use cases for working with low-resource languages. We present six of those in the following.

8.1 GUIDED HUMAN-IN-THE-LOOP DICTIONARY CREATION

Human feedback from native speakers is essential to clean up GUIDE's results and can even improve the model by training it on the incrementally updated data. This active learning [86] platform would ask native speakers questions, such as *"Would you use the word X to answer the question Y?"* (similarly to our questionnaires presented in Subsection 6.3.2). Because GUIDE is capable of few-shot learning, humans need to evaluate only a fraction of its predictions to make the GNN recognize new patterns. Users of this platform could not review just predictions but also add words from domains that the Bible does not cover. This platform can be implemented as *Streamlit* app[1], which is compatible with *PyTorch*.

In comparison to compiling bilingual dictionaries, semantic domain dictionaries have the advantage that even people with only monolingual expertise can perform a sanity check: In a list of words linked with the same SDQ, wrong words stand out, such as in *"ice, water, hour, steam"*.

8.2 BITEXT MINING

Another future direction is to use semantic domains for *bitext mining*, which is a basis for building cross-lingual encoders (e.g., XLM-R [19], LaBSE [29], SONAR [24], NLLB-200 [89]). It means finding/aligning sentences in different languages with the same meaning, so we can use them as training data. Extending training data is not trivial, as

[1]Streamlit page: `https://streamlit.io/` (visited on 06/10/2023)

most training data [73, 82] on the web is not labeled. Semantic domain dictionary entries can help to find such sentence pairs by comparing sets of SDQs, and it can even help create word-word alignments. More accurate alignments also improve bitexts, resulting in more accurate cross-lingual encoders. These encoders, for example, improve machine translation.

8.3 TRANSLATION EVALUATION

Translation evaluation is the task of determining if a translation accurately conveys the intended meaning of a source text (i.e., nothing has been added or removed). It is necessary to evaluate the quality of a machine translation system.

Bianchi, Nozza, and Hovy [9] propose to evaluate translation quality using language-invariant properties. These features preserve semantics as well as pragmatics and should not change. Therefore, semantic domains are one expressive feature for evaluating translation quality.

Common methods for translation evaluation are *Quality Estimation* (QE) and quality metrics, such as BLEU [71]. In the following, we present how semantic domain identification can enhance QE and compare it to using BLEU.

AUGMENTED QUALITY ESTIMATION One method for translation evaluation is QE [88]. QE happens without any reference translation for comparison. QE is especially useful for low-resource languages because it does not require high amounts of labeled data, and it reduces post-editing effort. Traditionally, language-specific models have been used for QE [88]. In combination with automatically created semantic domain dictionaries and semantic domain identification, existing QE methods could become language-agnostic. Such a quality metric could be: If the set of identified semantic domains in two sentences in different languages is similar (e.g., computed as cosine similarity), it can be considered to be high-quality translations.

BEYOND BLEU The *BiLingual Evaluation Understudy* (BLEU) metric compares a translation with a reference sentence that shows how the translation should look. Therefore, the score depends on the lexical choices of the translation system. Even if the translation perfectly conveys the meaning, it might receive a low score because it used synonyms of the expected words. Although other metrics have been proposed to address BLEU's weaknesses, such as METEOR [5] and COMET [79], BLEU is still popular [53, 89, 102]. Semantic domains are more general than 1:1 word matches and would capture different word choices. They also do not require the time-consuming creation of reference sentences. Creating reference sentences is especially expensive as BLEU uses multiple reference sentences or if it is applied in a low-resource setting. Semantic-domain-based evaluation could allow for more accurate and less costly translation evaluation.

8.4 OTHER FUTURE DIRECTIONS

Because low-resource languages are a mostly uncharted area in NLP, there are many more potential use cases for GUIDE. To conclude this chapter, we briefly highlight three of them.

LANGUAGE STANDARDIZATION Semantic domain dictionaries can help ensure standardized spellings in poorly documented languages.

THESAURUS Linguists or local language experts working on writing projects can look up words in the same SDQ and semantic domain to find alternative words. This thesaurus allows for more nuanced translations than literally translating words. For example, if a translator wants to translate the phrase "*daily bread*" into a language spoken in Papua New Guinea, trying to keep its idiomatic meaning, she could find the term for *sago food* listed under the same SDQ in the target language's semantic domain dictionary. Although sago food[2] is different from bread, it is the cultural equivalent of a common staple food, compared to bread in Western countries.

SENTENCE EMBEDDINGS *Language-agnostic BERT Sentence Embedding* (LaBSE) [29] currently supports 109 languages and can be fine-tuned to low-resource languages by computing presumptive sentence similarity scores based on overlapping semantic domains.

[2] *Sago food* on Wikipedia: `https://en.wikipedia.org/wiki/Sago` (visited on 05/10/2023)

9 CONCLUSION

"Languages shape our tools, and our tools shape languages."

— *GPT-4* [69]

This thesis presents the language-agnostic tool GUIDE, which creates and fills up multiparallel semantic domain dictionaries in 20 languages from seven language families. The model achieves state-of-the-art performance in linking words to their SDQs and supports 833 languages – technically even more than 3,500 languages. Although GUIDE has an estimated recall of only 9%, we show that it has a precision of 60% even in languages for which it has no training data, probably due to language similarity and the model's multilingual nature. We release the code, the pretrained model, the multilingual test set, and the new dictionaries on GitHub[1].

Based on our experiments, we can answer the research questions:

RQ1. We use a GCN to create entries for semantic domain dictionaries by performing a 1-hop graph convolution on a MAG built from Bible translations, annotated with the SDQ links from existing semantic dictionaries. This approach achieves a precision of 65% in 20 languages.

RQ2. Surprisingly, a higher number of known entries in the source dictionaries during the GCN's training increases the model's precision only slightly because its training process fine-tunes the initial weight matrix.

RQ3. The GNN is easily interpretable due to its single-layer architecture and the absence of an attention mechanism. It predicts a word-SDQ link because an aligned word has a link to the same SDQ. Therefore, users can trace GUIDE's decision process back to the input data.

In addition to our answers to the research questions, we propose 32,000 new word-SDQ links for twelve existing dictionaries. We have also contributed to the development of eight new dictionaries with 19,000 word-SDQ links in total. Ten out of these 20 languages are low-resource languages.

[1]GitHub repository: `https://github.com/janetzki/GUIDE`

In our eyes, the most promising future direction is the development of an active learning platform that takes advantage of the GUIDE's few-shot learning capabilities by incrementally refining the training data with targeted feedback from indigenous people.

A APPENDIX

A.1 BIBLE TRANSLATION SOURCES

Table A.1 shows the web source of the Bible translations we used.

A.2 QUESTIONNAIRES

Table A.2 provides the links to the questionnaires that we used to manually evaluate GUIDE's performance.

A.3 INCORRECT SEMANTIC DOMAIN DICTIONARY ENTRIES

Table A.3 shows incorrect entries that we discovered in the development dataset.

A.4 LEARNING CURVES

Figure A.1 shows GUIDE's learning curves for the ten training runs that we used to report its precision, recall, and $F1$ score in Table 7.1.

[1]translated by Google Translate: `https://translate.google.com/` (visited on 13/10/2023)

Language	Bible translation URL
DEVELOPMENT	
Bengali	https://github.com/BibleNLP/ebible/blob/main/corpus/ben-ben2017.txt
Chinese	https://github.com/BibleNLP/ebible/blob/main/corpus/cmn-cmn-cu89s.txt
English	https://github.com/BibleNLP/ebible/blob/main/corpus/eng-eng-web.txt
French	https://github.com/BibleNLP/ebible/blob/main/corpus/fra-frasbl.txt
Hindi	https://github.com/BibleNLP/ebible/blob/main/corpus/hin-hin2017.txt
Indonesian	https://github.com/BibleNLP/ebible/blob/main/corpus/ind-ind.txt
Kupang Malay	https://github.com/BibleNLP/ebible/blob/main/corpus/mkn-mkn.txt
Malayalam	https://github.com/BibleNLP/ebible/blob/main/corpus/mal-mal.txt
Nepali	https://github.com/BibleNLP/ebible/blob/main/corpus/npi-npiulb.txt
Portuguese	https://github.com/BibleNLP/ebible/blob/main/corpus/por-porbrbsl.txt
Spanish	https://github.com/BibleNLP/ebible/blob/main/corpus/spa-spablm.txt
Swahili	https://github.com/BibleNLP/ebible/blob/main/corpus/swh-swhulb.txt
TEST	
German	https://github.com/BibleNLP/ebible/blob/main/corpus/deu-deu1951.txt
Hiri Motu	https://github.com/BibleNLP/ebible/blob/main/corpus/hmo-hmo.txt
Igbo	https://ebible.org/details.php?id=ibo
Mina-Gen	https://www.bible.com/sl/versions/2236-gen-gegbe-biblia-2014
Motu	https://github.com/BibleNLP/ebible/blob/main/corpus/meu-meu.txt
South Azerbaijani	https://github.com/BibleNLP/ebible/blob/main/corpus/azb-azb.txt
Tok Pisin	https://github.com/BibleNLP/ebible/blob/main/corpus/tpi-tpi.txt
Yoruba	https://github.com/BibleNLP/ebible/blob/main/corpus/yor-yor.txt

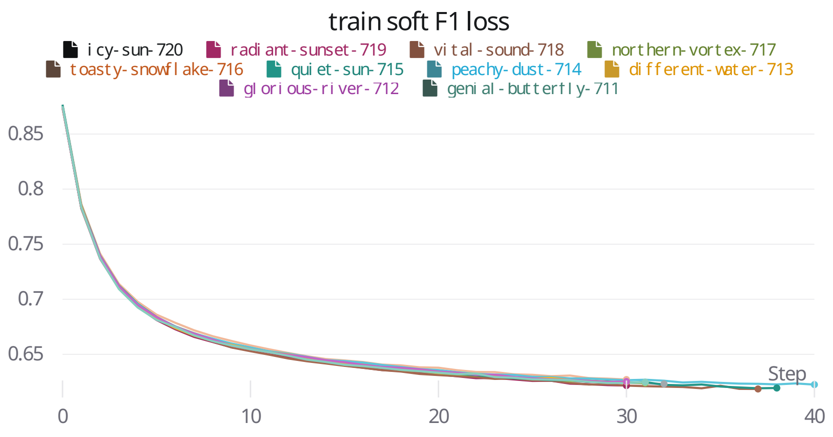
Table A.1: The links show the source of the Bible translations: All translations are from ebible.org, except the Mina-Gen Bible, which was provided by a language expert. We downloaded the Igbo Bible from ebible.org because it is not in the eBible corpus (i.e., on GitHub). All URLs have been visited on 21/10/2023.

Language	Questionnaire URL
DEVELOPMENT	
Bengali	https://docs.google.com/spreadsheets/d/1_qoYnswufDY0gVZuebcoQ1DD9BLqSVD8NozWzqiGWR8/edit?usp=sharing
Chinese (simplified)	https://docs.google.com/spreadsheets/d/1sppwKhC5Ev3frbQ8Mq_MoQGc5ehym6QdQSPcjjdWNPg/edit?usp=sharing
English	https://docs.google.com/spreadsheets/d/1zt_3gqNrbSYsIOzjwm3BxewOaulFY1laVXshCZIYGLU/edit?usp=sharing
French	https://docs.google.com/spreadsheets/d/1eWkOK5T9ttWx-9ZmETc-fUY1Q7HzihbTr6mK8irZ83g/edit?usp=sharing
Hindi	https://docs.google.com/spreadsheets/d/14D6pGKgQtoHG5LWORaU9Ko5XUn_wDh0-x4Hnxj2nHag/edit?usp=sharing
Indonesian	https://docs.google.com/spreadsheets/d/13iVFF0xxwpQ_pXf-zKFW2jebA3TZnIiSL9rFpD-dWPY/edit?usp=sharing
Malayalam	https://docs.google.com/spreadsheets/d/1-DFjBkS1wjCahowBjg-iGLBV-moZww-J8lKpO0HN44Y/edit?usp=sharing
Nepali	https://docs.google.com/spreadsheets/d/1n-f9LbF0vYfO4gtu1YmD6LZB1_Gyo-VxV35WaYBN9_Q/edit?usp=sharing
Portuguese	https://docs.google.com/spreadsheets/d/1_WKQmj5KHDE6p8MsCFawvQoxOcLn3MYPWb-4aYpgV6U/edit?usp=sharing
Kupang Malay	https://docs.google.com/spreadsheets/d/1EP1ctJ7yl5QYFdY6eV6KYDQg1_mz90j-J8jDGwT9yJY/edit?usp=sharing
Spanish	https://docs.google.com/spreadsheets/d/1-2ZwbunnsqOYBW_beI9Rax3XW1Zjpacl5GrrzlPtfF0/edit?usp=sharing
Swahili	https://docs.google.com/spreadsheets/d/1H9RVi1mCkL9WmcH2zXYuOwwj73CAMg1My6P-jAAWYgI/edit?usp=sharing
TEST	
German	https://docs.google.com/spreadsheets/d/1mPtzuD3_NFWOhLBXUElRNeAGtmiiXcKx_7n7Er3kZsc/edit?usp=sharing
Hiri Motu	https://docs.google.com/spreadsheets/d/1gTiNxhvRV9UtUq84Q0E3itJ2nYS8Gv4ElpIp3mEEHAE/edit?usp=sharing
Igbo	https://docs.google.com/spreadsheets/d/1yU8FCS19KRIWkbqm1aQBUCBEjVA4zoC60TuM0fTQN8Q/edit?usp=sharing
Mina-Gen	https://docs.google.com/spreadsheets/d/1Ib-xD6-1FuBLQ9M3F2UVbocnnbK7NBgv62Lg9h0p_6o/edit?usp=sharing
Motu	https://docs.google.com/spreadsheets/d/1e45Hw000K6OrluBQxe-8ifAR3h_Dz725sg3ZBlVnXRM/edit?usp=sharing
South Azerbaijani	https://docs.google.com/spreadsheets/d/1q8WfBhZDlOzRsihUbjH-wFyruudA11Chosotgf71rx0/edit?usp=sharing
Tok Pisin	https://docs.google.com/spreadsheets/d/1EENt0FJpTdDHpm2P1i-MQ56ZkdQKkBi5GnUDw30gx_o/edit?usp=sharing
Yoruba	https://docs.google.com/spreadsheets/d/11LBgUSHSnUFOP3Zp2ikgTp8vjGB6xR8cQkcdZeKNaSQ/edit?usp=sharing

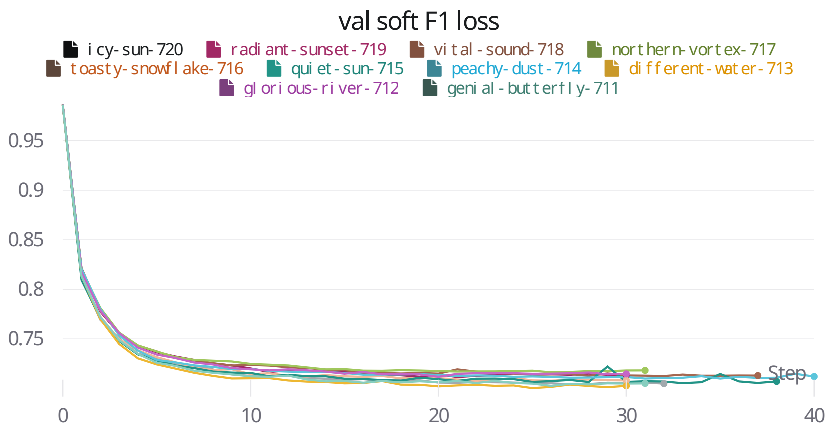
Table A.2: The completed questionnaires on Google Sheets for each of the 20 languages: We instructed the participants to answer 100 - 120 questions (see Subsection 6.3.2).

Language	Word	Translation[1]	SDQ ID	SDQ
English	stock		3.2.5.1-1	What words refer to believing that something is true?
Hindi	राल	resin	1.2.2.4-2	What types of minerals are there?
Portuguese	estoque	stock	3.2.5.1-1	What words refer to believing that something is true?
Portuguese	rebelião	rebellion	4.5.4.6-10	What do the authorities do to stop a rebellion?
Portuguese	estoque	stock	4.7.7.3-7	What means are used to restrain prisoners?
Portuguese	deter	detain	3.4.2.1.2-1	What words refer to feeling hateful?
Portuguese	carmesim	crimson	8.3.3.3.4-7	What are the shades of blue?
Spanish	rebelión	rebellion	4.5.4.6-10	What do the authorities do to stop a rebellion?
Spanish	sedición	sedition	4.5.4.6-10	What do the authorities do to stop a rebellion?

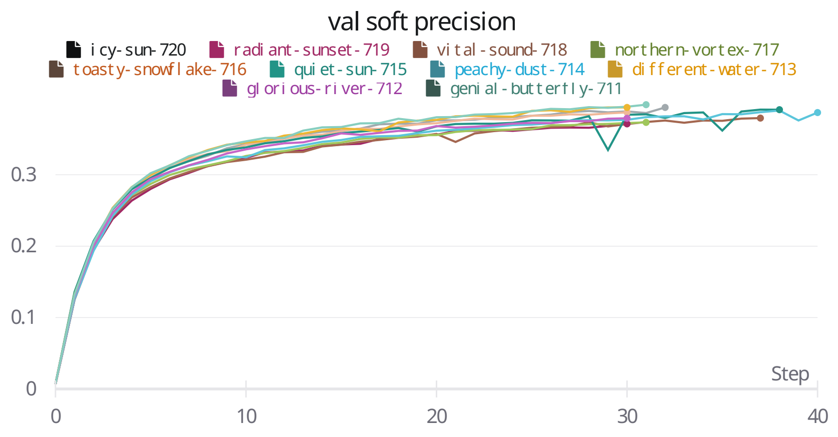
Table A.3: Nine incorrect entries in the semantic domain dictionaries that we discovered, verified by native speakers: The incorrect word-SDQ links show accumulations around certain topics, although they are rare.


train soft F1 loss
icy-sun-720
radiant-sunset-719
vital-sound-718
northern-vortex-717
toasty-snowflake-716
quiet-sun-715
peachy-dust-714
different-water-713
glorious-river-712
genial-butterfly-711
0.85
0.8
0.75
0.7
0.65
Step
0
10
20
30
40

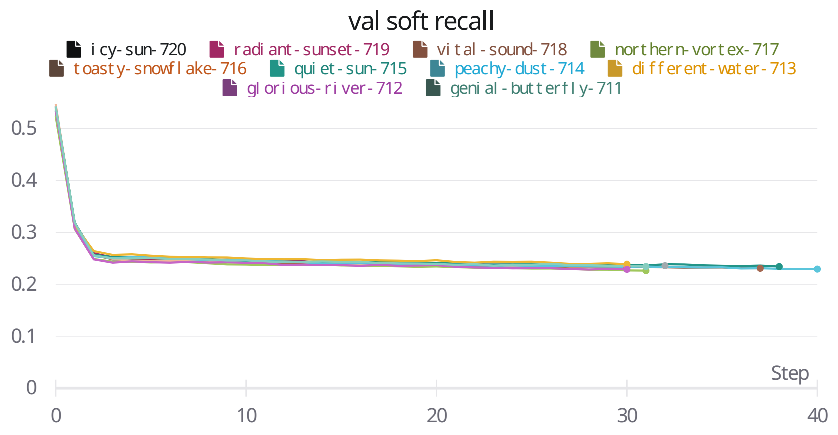
(a)


val soft F1 loss
icy-sun-720
radiant-sunset-719
vital-sound-718
northern-vortex-717
toasty-snowflake-716
quiet-sun-715
peachy-dust-714
different-water-713
glorious-river-712
genial-butterfly-711
0.95
0.9
0.85
0.8
0.75
Step
0
10
20
30
40

(b)



(c)



(d)

Figure A.1: Learning curves for the training and validation split: Each curve corresponds to one of the ten runs. Note that the soft recall is much higher than the final recall because we compute the soft recall based on the filtered nodes only (see paragraph 5.2.6). The respective learning curves for each of the twelve development languages are on WandB[2].

[2] WandB report: `https://api.wandb.ai/links/janetzki/pdacjt0s` (visited on (16/10/2023)

BIBLIOGRAPHY

[1] C. J. Abah, J. W. K. Ling and A. Govindasamy. *Root-Oriented Words Generation: An Easier Way Towards Dictionary Making for the Dusunic Family of Languages*. In: *MJSSH* (2018) (cit. on p. 2).

[2] K. Alnajjar, M. Hämäläinen, N. Tapio Partanen and J. Rueter. *Using Graph-Based Methods to Augment Online Dictionaries of Endangered Languages*. In: ComputEL. Association for Computational Linguistics, 2022 (cit. on p. 18).

[3] U. Alon and E. Yahav. *On the Bottleneck of Graph Neural Networks and its Practical Implications*. In: *ICLR* (2021) (cit. on p. 14).

[4] A. Alwosheel, S. Van Cranenburgh and C. G. Chorus. *Is your dataset big enough? Sample size requirements when using artificial neural networks for discrete choice analysis*. In: *Journal of Choice Modelling* 28 (Sept. 2018) (cit. on p. 59).

[5] S. Banerjee and A. Lavie. *METEOR: An Automatic Metric for MT Evaluation with Improved Correlation with Human Judgments*. In: *ACL* (2005) (cit. on p. 65).

[6] A. Bapna et al. *Building Machine Translation Systems for the Next Thousand Languages*. In: arXiv, 6 July 2022 (cit. on p. 3).

[7] J. Bastings, I. Titov, W. Aziz, D. Marcheggiani and K. Sima'an. *Graph Convolutional Encoders for Syntax-aware Neural Machine Translation*. In: *EMNLP*. 2017 (cit. on p. 14).

[8] L. Besacier, E. Barnard, A. Karpov and T. Schultz. *Automatic speech recognition for under-resourced languages: A survey*. In: *Speech Communication* 56 (Jan. 2014) (cit. on p. 7).

[9] F. Bianchi, D. Nozza and D. Hovy. *Language Invariant Properties in Natural Language Processing*. In: arXiv, 1 Oct. 2021 (cit. on p. 64).

[10] R. Biswas, A. Datta, N. Wagh, N. Katiki, R. Mahato and M. A. Lanham. *Building a Massive Multilingual Database of Words Mapped to Semantic Domains*. Massive Multilingual Knowledge Graph of Semantic Domains. 2022. URL: `http://matthewalanham.com/Students/2022/(2022,%20INFORMS%20BA)%20Massive%20Multilingual%20Knowledge%20Graph%20of%20Semantic%20Domains.pdf` (cit. on p. 18).

[11] D. Blasi, A. Anastasopoulos and G. Neubig. *Systematic Inequalities in Language Technology Performance across the World's Languages*. In: ACL. Association for Computational Linguistics, May 2022 (cit. on p. 7).

[12] B. H. Boerger. *Rapid Word Collection, dictionary production, and community well-being*. In: ICLDC. 3 Mar. 2017 (cit. on pp. 2, 11).

[13] P. Boldi and S. Vigna. *Axioms for Centrality*. In: arXiv, 6 Nov. 2013 (cit. on p. 60).

[14] BrainyQuote. *Inspirational Quotes at BrainyQuote*. URL: `https://www.brainyquote.com/` (visited on 12/10/2023) (cit. on pp. 17, 21, 25, 63).

[15] E. A. W. Budge. *The Rosetta Stone*. British Museum, 1913 (cit. on p. 8).

[16] M. Bugueño and G. de Melo. *Connecting the Dots: What Graph-Based Text Representations Work Best for Text Classification using Graph Neural Networks?* In: *arXiv* (2023) (cit. on p. 14).

[17] C. Christodoulopoulos and M. Steedman. *A massively parallel corpus: the Bible in 100 languages*. In: *LREC* (2015) (cit. on pp. 8, 9).

[18] A. Clauset, M. E. J. Newman and C. Moore. *Finding community structure in very large networks*. In: *Physical Review E* 70.6 (6 Dec. 2004) (cit. on p. 60).

[19] A. Conneau, K. Khandelwal, N. Goyal, V. Chaudhary, G. Wenzek, F. Guzmán, E. Grave, M. Ott, L. Zettlemoyer and V. Stoyanov. *Unsupervised Cross-lingual Representation Learning at Scale*. In: ACL. Online: Association for Computational Linguistics, 2020 (cit. on pp. 7, 19, 32, 63).

[20] G. Cordasco and L. Gargano. *Community Detection via Semi-Synchronous Label Propagation Algorithms*. In: 23 Mar. 2011 (cit. on p. 60).

[21] K. Dobler and G. de Melo. *FOCUS: Effective Embedding Initialization for Specializing Pretrained Multilingual Models on a Single Language*. In: *arXiv* (2023) (cit. on p. 19).

[22] Z.-Y. Dou and G. Neubig. *Word Alignment by Fine-Tuning Embeddings on Parallel Corpora*. In: EACL. 2021 (cit. on p. 13).

[23] C. T. Duong, T. D. Hoang, H. Dang, Q. V. H. Nguyen and K. Aberer. *On Node Features for Graph Neural Networks*. In: *arXiv* (20 Nov. 2019) (cit. on p. 32).

[24] P.-A. Duquenne, H. Schwenk and B. Sagot. *SONAR: Sentence-Level Multimodal and Language-Agnostic Representations*. In: *arXiv* (2023) (cit. on p. 63).

[25] C. Dyer, V. Chahuneau and N. A. Smith. *A Simple, Fast, and Effective Reparameterization of IBM Model 2*. In: NAACL. 2013 (cit. on p. 13).

[26] D. M. Eberhard, G. F. Simons and C. D. Fennig. *Ethnologue: Languages of the World*. twenty-sixth edition. SIL International, 2023 (cit. on pp. 1, 22, 61).

[27] A. Ebrahimi and K. Kann. *How to Adapt Your Pretrained Multilingual Model to 1600 Languages*. In: *ACL/IJCNLP* (2021) (cit. on p. 59).

[28] C. Fellbaum. *WordNet: An Electronic Lexical Database*. In: *Language* 76 (2000) (cit. on p. 17).

[29] F. Feng, Y. Yang, D. Cer, N. Arivazhagan and W. Wang. *Language-agnostic BERT Sentence Embedding*. In: ACL. Association for Computational Linguistics, 2022 (cit. on pp. 63, 65).

[30] M. Fey. *Just Jump: Dynamic Neighborhood Aggregation in Graph Neural Networks*. In: arXiv, 15 Apr. 2019. arXiv: `1904.04849[cs, stat]` (cit. on p. 57).

[31] M. Fey and J. E. Lenssen. *Fast Graph Representation Learning with PyTorch Geometric*. In: ICLR. 2019 (cit. on p. 32).

[32] R. Forkel. *The World Atlas of Language Structures Online*. The World Atlas of Language Structures Online. 2009. URL: `https://wals.info/` (visited on 11/10/2023) (cit. on p. 61).

[33] N. Goyal, C. Gao, V. Chaudhary, P.-J. Chen, G. Wenzek, D. Ju, S. Krishnan, M. Ranzato, F. Guzmán and A. Fan. *The FLORES-101 Evaluation Benchmark for Low-Resource and Multilingual Machine Translation*. In: *ACL* 10 (4 May 2022) (cit. on pp. 8, 19).

[34] J. Gu, H. Hassan, J. Devlin and V. Li. *Universal Neural Machine Translation for Extremely Low Resource Languages*. In: *arXiv* (2018) (cit. on p. 7).

[35] F. Guzmán, P.-J. Chen, M. Ott, J. Pino, G. Lample, P. Koehn, V. Chaudhary and M. Ranzato. *The FLoRes Evaluation Datasets for Low-Resource Machine Translation: Nepali-English and Sinhala-English*. In: *arXiv* (4 Feb. 2019) (cit. on pp. 8, 19).

[36] B. Haddow, R. Bawden, A. V. M. Barone, J. Helcl and A. Birch. *Survey of Low-Resource Machine Translation*. In: *COLING* 48.3 (1 Sept. 2022) (cit. on pp. 7–9).

[37] A. Hagberg, D. Schult and P. Swart. *Exploring Network Structure, Dynamics, and Function using NetworkX*. In: *SciPy*. 2008 (cit. on p. 25).

[38] M. A. Hedderich, L. Lange, H. Adel, J. Strötgen and D. Klakow. *A Survey on Recent Approaches for Natural Language Processing in Low-Resource Scenarios*. In: arXiv, 9 Apr. 2021 (cit. on p. 7).

[39] A. Imani Googhari, M. Jalili Sabet, P. Dufter, M. Cysou and H. Schütze. *ParCourE: A Parallel Corpus Explorer for a Massively Multilingual Corpus*. In: ACL/IJCNLP. Online: Association for Computational Linguistics, 2021 (cit. on p. 59).

[40] A. Imani Googhari, M. Jalili Sabet, L. K. Senel, P. Dufter, F. Yvon and H. Schütze. *Graph Algorithms for Multiparallel Word Alignment*. In: EMNLP. Online and Punta Cana, Dominican Republic: Association for Computational Linguistics, 2021 (cit. on p. 13).

[41] A. Imani Googhari, L. K. Şenel, M. J. Sabet, F. Yvon and H. Schütze. *Graph Neural Networks for Multiparallel Word Alignment*. In: ACL. arXiv, 2022 (cit. on pp. 13, 25, 30, 60).

[42] A. Imani, S. Severini, M. J. Sabet, F. Yvon and H. Schütze. *Graph-Based Multilingual Label Propagation for Low-Resource Part-of-Speech Tagging*. In: EMNLP. arXiv, 2022 (cit. on pp. 13, 20).

[43] M. Jalili Sabet, P. Dufter, F. Yvon and H. Schütze. *SimAlign: High Quality Word Alignments Without Parallel Training Data Using Static and Contextualized Embeddings*. In: EMNLP. Online: Association for Computational Linguistics, 2020 (cit. on p. 13).

[44] D. Kamholz, J. Pool and S. Colowick. *PanLex: Building a Resource for Panlingual Lexical Translation*. In: International Conference on Language Resources and Evaluation. 1 May 2014 (cit. on p. 18).

[45] D. P. Kingma and J. Ba. *Adam: A Method for Stochastic Optimization*. In: *ICLR* (22 Dec. 2014) (cit. on p. 37).

[46] T. N. Kipf and M. Welling. *Semi-Supervised Classification with Graph Convolutional Networks*. In: arXiv, 22 Feb. 2017 (cit. on p. 14).

[47] E. Knyazeva, P. B. de Mareüil and F. Vernier. *Aesop's fable "The North Wind and the Sun" Used as a Rosetta Stone to Extract and Map Spoken Words in Under-resourced Languages*. In: *LREC* (2022) (cit. on p. 17).

[48] P. Koehn, A. Axelrod, A. Birch, C. Callison-Burch, M. Osborne and D. Talbot. *Explorer Edinburgh System Description for the 2005 IWSLT Speech Translation Evaluation*. In: IWSLT. 2005 (cit. on p. 30).

[49] A. Krizhevsky, I. Sutskever and G. E. Hinton. *ImageNet classification with deep convolutional neural networks*. In: *ACM* 60.6 (24 May 2017) (cit. on p. 14).

[50] T. Kudo. *Subword Regularization: Improving Neural Network Translation Models with Multiple Subword Candidates*. In: arXiv, 29 Apr. 2018 (cit. on p. 29).

[51] T. Kudo and J. Richardson. *SentencePiece: A simple and language independent subword tokenizer and detokenizer for Neural Text Processing*. In: EMNLP. Association for Computational Linguistics, 2018 (cit. on p. 29).

[52] S. Kudugunta et al. *MADLAD-400: A Multilingual And Document-Level Large Audited Dataset*. In: arXiv, 8 Sept. 2023 (cit. on p. 2).

[53] A. Kumar, R. Baruah, A. Pratap, M. Swarnkar and A. K. Singh. *Exploiting Language Relatedness in Machine Translation Through Domain Adaptation Techniques*. In: *arXiv* (2023) (cit. on p. 65).

[54] G. Lample and A. Conneau. *Cross-lingual Language Model Pre-training*. In: NeurIPS. 22 Jan. 2019 (cit. on p. 7).

[55] C. Leong, J. Nemecek, J. Mansdorfer, A. Filighera, A. Owodunni and D. Whitenack. *Bloom Library: Multimodal Datasets in 300+ Languages for a Variety of Downstream Tasks*. In: arXiv, 2022 (cit. on p. 4).

[56] M. Lesk. *Automatic Sense Disambiguation Using Machine Readable Dictionaries: How to Tell a Pine Code from an Ice Cream Cone*. In: SIGDOC. ACM Press, 1986 (cit. on p. 60).

[57] V. Levenshtein. *Binary codes capable of correcting deletions, insertions, and reversals*. In: *Soviet physics. Doklady* (1965) (cit. on p. 18).

[58] W. Lewis. *Haitian Creole: How to Build and Ship an MT Engine from Scratch in 4 days, 17 hours, & 30 minutes*. In: EAMT. Saint Raphaël, France: European Association for Machine Translation, 27 May 2010 (cit. on p. 9).

[59] Lewix, C.S. *From: The Four Loves*. In: *INTAMS review*. ISSN: 1783-1474, 1783-1474 Issue: 2. 1998 (cit. on p. 25).

[60] D. P. Lyras, K. N. Sgarbas and N. D. Fakotakis. *Using the Levenshtein Edit Distance for Automatic Lemmatization: A Case Study for Modern Greek and English*. In: ICTAI. Vol. 2. Oct. 2007 (cit. on p. 55).

[61] C. Manning, M. Surdeanu, J. Bauer, J. Finkel, S. Bethard and D. McClosky. *The Stanford CoreNLP Natural Language Processing Toolkit*. In: ACL. Association for Computational Linguistics, June 2014 (cit. on p. 27).

[62] S. Mannor, D. Peleg and R. Rubinstein. *The Cross Entropy Method for Classification*. In: ICML. Association for Computing Machinery, 7 Aug. 2005 (cit. on pp. 35, 57).

[63] D. Marcheggiani, J. Bastings and I. Titov. *Exploiting Semantics in Neural Machine Translation with Graph Convolutional Networks*. In: *NAACL*. 2018 (cit. on p. 14).

[64] M.-C. de Marneffe, C. D. Manning, J. Nivre and D. Zeman. *Universal Dependencies*. In: *Computational Linguistics* 47.2 (13 July 2021) (cit. on p. 27).

[65] G. de Melo and G. Weikum. *Towards a Universal Wordnet by Learning from Combined Evidence*. In: ACM. Hong Kong, China: ACM Press, 2009 (cit. on p. 17).

[66] T. Mikolov, K. Chen, G. Corrado and J. Dean. *Efficient Estimation of Word Representations in Vector Space*. In: arXiv, 6 Sept. 2013 (cit. on p. 18).

[67] R. Moe. *Compiling Dictionaries Using Semantic Domains*. In: *Lexikos* (2010) (cit. on pp. 2, 10, 11).

[68] G. Neubig, Y. Matsubayashi, M. Hagiwara and K. Murakami. *Safety Information Mining — What can NLP do in a disaster—*. In: IJCNLP. Asian Federation of Natural Language Processing, Nov. 2011 (cit. on p. 9).

[69] OpenAI. *GPT-4 Technical Report*. In: *arXiv* (2023) (cit. on p. 67).

[70] I. Orife et al. *Masakhane - Machine Translation For Africa*. In: *arXiv* (13 Mar. 2020) (cit. on p. 7).

[71] K. Papineni, S. Roukos, T. Ward and W.-J. Zhu. *BLEU: a method for automatic evaluation of machine translation*. In: ACL. Philadelphia, Pennsylvania: Association for Computational Linguistics, 2001 (cit. on p. 64).

[72] P. E. Pope, S. Kolouri, M. Rostami, C. E. Martin and H. Hoffmann. *Explainability Methods for Graph Convolutional Neural Networks*. In: *CVPR* (June 2019) (cit. on p. 3).

[73] M. Prasad, T. Breiner and D. Van Esch. *Mining Training Data for Language Modeling Across the World's Languages*. In: SLTU. ISCA, 29 Aug. 2018 (cit. on p. 64).

[74] P. Qi, Y. Zhang, Y. Zhang, J. Bolton and C. D. Manning. *Stanza: A Python Natural Language Processing Toolkit for Many Human Languages*. In: arXiv, 23 Apr. 2020 (cit. on pp. 27, 29).

[75] Quotefancy. *Inspirational Quotes on Beautiful Wallpapers*. URL: `https://quotefancy.com/` (visited on 12/10/2023) (cit. on pp. 1, 7, 37, 41).

[76] A. Ramachandran and G. de Melo. *Cross-Lingual Emotion Lexicon Induction using Representation Alignment in Low-Resource Settings*. In: COLING. International Committee on Computational Linguistics, 2020 (cit. on p. 8).

[77] S. Ranathunga, E.-S. A. Lee, M. P. Skenduli, R. Shekhar, M. Alam and R. Kaur. *Neural Machine Translation for Low-Resource Languages: A Survey*. In: arXiv, 29 June 2021 (cit. on p. 7).

[78] I. Redko, E. Morvant, A. Habrard, M. Sebban and Y. Bennani. *Advances in Domain Adaptation Theory*. ISTE Press - Elsevier, 28 Aug. 2019 (cit. on p. 19).

[79] R. Rei, C. Stewart, A. C. Farinha and A. Lavie. *COMET: A Neural Framework for MT Evaluation*. In: arXiv, 19 Oct. 2020 (cit. on p. 65).

[80] R. Rymer. *Vanishing Voices*. Magazine. 1 July 2012. URL: `https://www.nationalgeographic.com/magazine/article/vanishing-languages` (visited on 21/10/2023) (cit. on p. 1).

[81] R. Sato. *A Survey on The Expressive Power of Graph Neural Networks*. In: *arXiv* (2020) (cit. on pp. 14, 15).

[82] K. P. Scannell. *The Crúbadán Project: Corpus building for under-resourced languages*. In: SIGWAC. 2007 (cit. on pp. 3, 64).

[83] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner and G. Monfardini. *The Graph Neural Network Model*. In: *IEEE Transactions on Neural Networks and Learning Systems* 20.1 (Jan. 2009) (cit. on pp. 3, 14).

[84] M. Schlichtkrull and A. Søgaard. *Cross-Lingual Dependency Parsing with Late Decoding for Truly Low-Resource Languages*. In: EACL. Association for Computational Linguistics, 2017 (cit. on pp. 3, 20).

[85] R. Sennrich, B. Haddow and A. Birch. *Neural Machine Translation of Rare Words with Subword Units*. In: ACL. Association for Computational Linguistics, Aug. 2016 (cit. on p. 29).

[86] B. Settles. *Active Learning Literature Survey*. Technical Report 1648. Madison: University of Wisconsin–Madison, 2009 (cit. on pp. 60, 63).

[87] F. Smadja, V. Hatzivassiloglou and K. R. McKeown. *Translating Collocations for Bilingual Lexicons: A Statistical Approach*. In: *COLING* 22.1 (1996) (cit. on pp. 8, 60).

[88] S. Sun, M. Fomicheva, F. Blain, V. Chaudhary, A. El-Kishky, A. Renduchintala, F. Guzmán and L. Specia. *An Exploratory Study on Multilingual Quality Estimation*. In: *AACL* (2020) (cit. on p. 64).

[89] N. Team et al. *No Language Left Behind: Scaling Human-Centered Machine Translation*. In: *arXiv* (2022) (cit. on pp. 7, 8, 19, 63, 65).

[90] J. Tiedemann. *Parallel Data, Tools and Interfaces in OPUS*. In: *LREC* (2012) (cit. on p. 8).

[91] UNESCO. *UNESCO Atlas on the World's Languages in Danger*. 2010 (cit. on p. 1).

[92] A. Vaswani, N. M. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser and I. Polosukhin. *Attention is All you Need*. In: *arXiv* (12 June 2017) (cit. on pp. 33, 56).

[93] P Veličković, A Casanova, P Liò, G Cucurull, A Romero and Y Bengio. *Graph Attention Networks*. In: *arXiv* (2018). In collab. with Apollo-University Of Cambridge Repository and University Of Cambridge (cit. on p. 33).

[94] Z. Wang, X. Liu, P. Yang, S. Liu and Z. Wang. *Cross-lingual Text Classification with Heterogeneous Graph Neural Network*. In: ACL/IJCNLP. Online: Association for Computational Linguistics, 2021 (cit. on p. 3).

[95] K. Whistler. *A Unicode Standard Annex (UAX) #15*. Technical Report 15. The Unicode Consortium, 2023 (cit. on p. 56).

[96] Wikipedia. *List of countries by the number of recognized official languages*. Wikipedia. 22 Aug. 2023. URL: `https://en.wikipedia.org/w/index.php?title=List_of_countries_by_the_number_of_recognized_official_languages&oldid=1171644848` (visited on 09/10/2023) (cit. on p. 1).

[97] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang and P. S. Yu. *A Comprehensive Survey on Graph Neural Networks*. In: *IEEE Transactions on Neural Networks and Learning Systems* 32.1 (Jan. 2021) (cit. on p. 15).

[98] Wycliffe Global Alliance. *Global Scripture Access*. Global Scripture Access. 2023. URL: `https://www.wycliffe.net/resources/statistics/` (visited on 22/09/2023) (cit. on p. 3).

[99] J. Xu and W. B. Croft. *Corpus-based stemming using cooccurrence of word variants*. In: *ACM Transactions on Information Systems* 16.1 (Jan. 1998) (cit. on p. 55).

[100] K. Xu, W. Hu, J. Leskovec and S. Jegelka. *How Powerful are Graph Neural Networks?* In: *arXiv* (1 Oct. 2018) (cit. on p. 14).

[101] L. Yao, C. Mao and Y. Luo. *Graph Convolutional Networks for Text Classification*. In: *AAAI*. 2019 (cit. on p. 14).

[102] V. Åkerman, D. Baines, D. Daspit, U. Hermjakob, T. Jang, M. Martin, J. Mathew and M. Schwarting. *The eBible Corpus: Data and Model Benchmarks for Bible Translation for Low-Resource Languages*. In: *arXiv* (2023) (cit. on pp. 4, 8–10, 65).

[103] R. Östling and J. Tiedemann. *Efficient Word Alignment with Markov Chain Monte Carlo*. In: *PBML* 106.1 (1 Oct. 2016) (cit. on pp. 11, 13).

[104] ∀ et al. *Participatory research for low-resourced machine translation: A case study in African languages*. In: *EMNLP*. Online: Association for Computational Linguistics, Nov. 2020 (cit. on pp. 7, 8, 21).

DECLARATION OF AUTHORSHIP

I certify that the material contained in this thesis is my own work and does not contain unreferenced or unacknowledged material.

Hiermit versichere ich, dass ich die vorliegende Arbeit selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel verwendet habe.

Potsdam, 24th October 2023

Jonathan Janetzki