SCHOOL OF COMPUTATION,
INFORMATION AND TECHNOLOGY —
INFORMATICS

TECHNISCHE UNIVERSITÄT MÜNCHEN

Master's Thesis in Robotics Cognition Intelligence

Optimizing a minimal language using Pre-trained Language Models

Ajay Narayanan

SCHOOL OF COMPUTATION, INFORMATION AND TECHNOLOGY — INFORMATICS

TECHNISCHE UNIVERSITÄT MÜNCHEN

Master's Thesis in Robotics Cognition Intelligence

Optimizing a minimal language using Pre-trained Language Models

Optimierung einer Minimalsprache mithilfe vorab trainierter Sprachmodelle

Author:	Ajay Narayanan
Examiner:	Vincent Fortuin
Supervisor:	Vincent Fortuin
Submission Date:	15-05-2025

I confirm that this master’s thesis is my own work and I have documented all sources and material used.

Munich, 15-05-2025 Ajay Narayanan

Acknowledgments

I would like to thank Vincent Fortuin, my supervisor for his guidance and support throughout this thesis. I would also like to thank Sara Visconti, who worked with me to develop the language pipeline. I would also like to thank my family and friends for their continued support and encouragement during my work.

Abstract

Contents

1. Introduction

This opening chapter will provide an overview of the research topic, reviewing the study's motivation, the research questions, and the methodology. The chapter will also provide a brief outline of the structure of the thesis.

1.1. Motivation

The goal of human language, much like any other human activity, is to aid our survival and propagation. It helps us to collaborate, compete and influence others [Lev22]. Efficiency is another hallmark of all living beings, as a product of biological evolution [Ha10]. Thus, languages form from a trade-off between expressiveness and efficiency. They tend to follow rules like Zipf's Law of Abbreviations CITE, that words that are more frequently used tend to be shorter than sparsely used words.

This is, of course, an idealization. In the real world, speakers must talk in noisy environments, which means that languages often have redundancy built in. This redundancy helps in error correction and understanding in such noisy environments. In addition, natural languages are not an optimal code in terms of information theory. This is due to the fact that human languages are constrained by the limitations of incremental language processing, which enforces constraints which force systematicity, compositionality and concatenation as means of combination [FH22].

Constructed Languages (ConLangs) are languages constructed by an individual or group of individuals rather than having evolved naturally [Sch21]. ConLangs are used for various purposes, such as to serve as an international auxiliary language (e.g. Esperanto), to create a fictional world (Dothraki, Klingon, or Quenya), to be logically rigorous, (e.g. Lojban), or simple and easy to learn (e.g., Toki Pona). Some, like Ithkuil [24] are designed to express more profound thoughts briefly but clearly.

Naturally, the next question that arises is whether we could design a language that is more optimal or efficient than natural languages. This question itself is more challenging than it seems. What does it mean for a language to be optimal? Is it efficient in terms of information theory? Is it ease of learning? Is it expressiveness? Is it the ability to convey complex thoughts? Is it the ability to be understood in even the most noisy environments? In addition, how do we measure these properties? Therefore, the first motivation of this thesis is to explore what it means for a language to be optimal and whether we can design a language that is more optimal than natural languages. To develop this language, one must define its phonology, orthography, morphology, syntax, and vocabulary.

The second motivation of this thesis is to explore whether Large Language Models (LLMs) can provide insights into this problem. In the process of learning, Large Language Models and Machine learning models in general, learn to encode various features. These features potentially encode information about the structure of language itself, and facts about our world. Trained on a large and varied corpus, they could encode information about nearly everything humans communicate about. Thus, it is possible that LLMs could provide insights that could help us design a more optimal language.

To test these hypotheses, we setup a modular pipeline for generating constructed languages. The code for this pipeline is available at `https://github.com/fortuinlab/conlang`.

This thesis is divided into 6 chapters. Chapter 2 introduces the necessary background information prerequisite for the main topic. It also details the various literature surveyed as part of the research. Chapter 3 introduces the research methodology and the primary research questions. Chapter 4 describes the implementation of the pipeline, and the various modules used in the pipeline. Chapter 5 details the results of the experiments conducted using the pipeline, which are then discussed in Chapter 6, along with the conclusion and areas for future research. Chapter 7 details the individual contributions of the author to the research, and the codebase.

2. Background

2.1. Linguistics

Linguistics, the scientific study of languages is a broad and complex field encompassing various subfields. Although a comprehensive summary of Linguistics is beyond the scope of this thesis, we will briefly discuss some of the subfields and key concepts that are relevant to our work. [TS07] provides a glossary of linguistic terms that can be useful for readers interested in a more detailed overview of the field.

2.1.1. Phonetics and Phonology

Phonetics is the study of the physical sounds of human speech, their production, transmission and reception. [TS07]. The International Phonetic Alphabet (IPA) is a standardized system of phonetic notation that represents the sounds of spoken language. The system is based on the assumption that speech can be represented partly as a sequence of discrete sounds or *segments* [99]. In addition, the IPA also includes symbols for suprasegmental features such as stress and intonation. The full IPA Chart (reproduced here in A.1) shows all the symbols and diacritics used to represent sounds in the IPA. Sounds and words can be transcribed in IPA using [ ] brackets. For example, the sounds for the word *this* can be transcribed as [ðɪs]. The IPA helps linguistics transcribe sounds in a language-agnostic way, allowing them to compare sounds across languages. The IPA Handbook [99] provides a comprehensive guide to the use of the IPA.

Phonology is the study of the sound systems of languages, including the patterns of sounds and the rules that govern their distribution. [TS07]. The key difference in the disciplines is driven by the concept of a *phoneme*. A phoneme is an abstract unit of sound that can distinguished by a native speaker of a language. Phonemes and Phonemic transcriptions are represented using slashes / /. The key points about phonemes are:

1. Letters do not necessarily correspond to phonemes. For example, the English word *this* has four letters but 3 phonemes (/ðɪs/).

2. Phonemes can be realized as different sounds in different contexts. For example,

the English phoneme /p/ can be realized as [pʰ] in the word *pin*([pʰɪn]) and [p] in the word *spin*([spɪn]). i.e. in English, the sounds [p] and [pʰ] are *allophones* of the phoneme /p/.

3. Two sounds are considered different phonemes if changing them can change the meaning of a word. e.g. [dɛn] *den* and [ðɛn] *then* are distinct words in English.

Phonotactics defines the rules that govern the permissible sound sequences in a language [TS07]. For example, in English, the sequence /bl/ is permissible at the beginning of a word (e.g. *bled*) but not the sequence /bn/. Languages usually modify loadwords to fit their own phonotactic constraints. For example, the English word *beer* is borrowed into Japanese as *biru*.

2.1.2. Morphology

Morphology is the study of the structure of words and the rules that govern the formation of words in a language [TS07]. Most studies of morphology focus on the concept of a *morpheme*, the smallest unit of meaning in a language. For example, the word *unhappiness* can be broken down into three morphemes: *un-*, *happy* and *-ness*. Morphemes can be free or bound. Free morphemes can stand alone as words (e.g. *happy*) while bound morphemes must be attached to other morphemes (e.g. *-ness*).

Morphology can be further divided into **inflectional** and **derivational** morphology. **Inflectional morphology** involves the modification of a word for grammatical purposes such as tense, aspect, mood, number, e.g. the English verb *walk* can be inflected to *walked*, *walks*, *walking*, etc. **Derivational morphology** involves the creation of new words from existing words. For example, the English noun *happiness* can be derived from the adjective *happiness* by adding the suffix *-ness*.

Lexicon

The **lexicon** of a language is the vocabulary of a language, i.e. the total set of words available for a speaker. It is better to consider the lexicon, not as a list of word, but a set of lexical resources including morphemes, and processes to construct words from these resources [TS07].

2.1.3. Grammar

Grammar is the set of rules that govern the structure of sentences in a language. Traditional Grammar describes certain terms for basic grammatical components such as *article*, *adjective*, *noun*, etc known as *parts of speech* [Yul20].

1. **Nouns** are words that refer to people, places, things, or abstract ideas, as if they were objects. For example, *cat*, *house*, and *happiness* are all nouns.

2. **Verbs** are words that express the actions or states of nouns. For example, *run*, *is*, and *happen* are all verbs.

3. **Adjectives** are words that describe or modify nouns. For example, *happy*, *red*, and *tall* are all adjectives.

4. **Adverbs** are words that describe or modify verbs, adjectives, or other adverbs. For example, *really*, *very*, and *well* are all adverbs.

5. **Articles** are words that define a noun as specific or unspecific. For example, *the* is a definite article, while *a* and *an* are indefinite articles.

6. **Pronouns** are words that take the place of noun phrases, typically when they are already known. For example, *he*, *she*, and *they* are all pronouns.

7. **Prepositions** are words that show the relationship between a noun or pronoun and other words in a sentence. For example, *in*, *on*, and *at* are all prepositions.

8. **Conjunctions** are words that connect words, phrases, or clauses, and indicate the relationship between them. For example, *and*, *but*, and *or* are all conjunctions.

Sometimes, parts of speech exhibit multiple forms, used in different grammatical circumstances. Each of these forms indicate a certain *grammatical category* or *feature*. For example, in English, verbs can be inflected for tense, aspect, mood, person, number, etc. This is known as *agreement*.

A language may choose to explicitly mark these features, i.e. *grammaticalize* them [Ros10]. All features can be expressed in any language (perhaps by adding explicit information), but every language chooses to express only a subset of these features grammatically.

2.1.4. Grammatical Features

Grammatical features or categories provide some extra information about the sentence, and different parts of speech may exhibit different forms to indicate these features. we will briefly discuss some of the most common grammatical features.

Grammatical Gender

In many languages, nouns are often classified into different classes, and different parts of speech will often agree with the specific class. These classes are called Grammatical Gender, and it often need not have anything to do with sex or gender. Languages with gender may only have 2 gender classes, but can also have many more. The gender assignment of many objects are often arbitrary.

Grammatical Number

Grammatical Number is a grammatical category that expresses count distinctions. The most common distinction is between singular(one) and plural(many), but some languages also have dual(two), trial(three), paucal(few), and other forms. For example, in English, the noun *cat* is singular, while *cats* is plural.

Grammatical Case

The **Grammatical Case** indicates one or more functions of a noun or noun phrase in a sentence. Many different cases have been identified in the worlds languages, such as

1. **Nominative** case, which indicates the subject of a verb.
2. **Accusative** case, which indicates the direct object of a verb.
3. **Dative** case, which indicates the indirect object of a verb.
4. **Genitive** case, which indicates possession.
5. **Locative** case, which indicates location.
6. **Instrumental** case, which indicates the means by which an action is performed.

These descriptions are not exact, and precise distinctions can heavily depend on the specific language.

Tense, Aspect and Mood

Tense, aspect and modality all provide some kind of information that is temporal in nature, or tell us about the status of the action or verb. They are often grouped together as **TAM** (Tense, Aspect, Modality).

Tense is a grammatical category that indicates the time at which an action takes place. The most common tenses are past, present, and future. For example, in English, the verb *walk* can be inflected to *walked* (past), *walks* (present), and *will walk* (future).

Aspect is a grammatical category that indicates the temporal structure of the action or event described by a verb. It indicates for example, whether the action is bounded, and unitary, or continuous or habitual. For example, in English, the sentences *She danced* and *She was dancing* have different aspects. They are both in the past tense, but the first sentence indicates a *perfective*, or completed aspect, while the second sentence indicates an *continuous*, or *progressive* aspect.

Modality is a grammatical category that indicates the speaker's attitude towards the action or event described by a verb. Modern linguists usually associate it with the expression of obligation, permission, prohibition, necessity, possibility and ability [TS07]. In English, modality is primarily expressed using Auxiliary verbs, such as *can*, *may*, *must*, etc. For example, the sentence *He can dance* indicates ability, while the sentence *He must dance* indicates obligation.

Grammatical Person

Grammatical Person is a grammatical category that indicates the different relationships between the speaker, the listener, and others in the discourse. Languages typically indicate this relationship using pronouns. The most common distinctions are between first person (the speaker), second person (the listener), and third person (others). For example, in English, the pronouns *I* (first person), *you* (second person), and *he/she/they/it* (third person) indicate the grammatical person.

Some languages also have a distinction in *clusivity* for the first person plural pronoun. The inclusive form includes the listener, while the exclusive form does not. For example in Malayalam, the pronoun *nammal* includes the listener, while the pronoun *njangal* does not.

2.1.5. Syntax

Syntax is the study of the structure of sentences and the rules that govern the formation of sentences in a language [TS07]. The goal of Syntactic Analysis is to have a finite set of rules that could be used to generate potentially infinite sentences. This set of rules is known as a **Generative Grammar** [Yul20]. We move from the concepts of Nouns to Noun Phrases, and Verbs to Verb Phrases, and so on.

A Noun Phrase is a phrase that is interchangeable with a noun. Take for example the sentence:

__________ bought a new car.

The cloze in the sentence could be filled with a noun, like "John", but also by say , "The young man".

With these definitions in mind, we could define a sentence as a Noun Phrase followed by a Verb Phrase. We can also have production rules for Noun Phrases, Verb Phrases, and so on. For example, a Noun Phrase could be defined as a determiner followed by an adjective followed by a noun. With these rules, known as **Phrase Structure Rules**, we can generate a tree structure for a sentence, known as a **Parse Tree** [JM25].

2.2. Constructed Languages

Constructed Languages are languages that have not naturally evolved, but were artifically constructed. Some conlangs are created for fictional word-building, like *Quenya* and *Sindarin* from J.R.R. Tolkien's Middle-Earth, or *Dothraki* and *High Valyrian* from George R.R. Martin's A Song of Ice and Fire.

2.2.1. The Process of Language Creation

2.3. Evaluation

2.3.1. Evaluation of Machine Translations

Evaluation of machine translations is a complex task, and is essential for assessing the accuracy and fluency of the translations. Human evaluations are often expensive and time-consuming [Pap+02], and are not always feasible for large datasets. As a result, many researchers have developed automatic evaluation metrics to assess the quality of machine translations. Classic methods like BLEU [Pap+02] measures the similarity between the machine translation and a reference translation by comparing n-grams. ROUGE [Lin04] is another popular metric that measures recall as opposed to precision. It is often used for evaluating the quality of summaries, but can also be used for machine translation evaluation. METEOR [BL05] improves upon such methods by for example, considering synonyms and stemming.

We use these machine translation evaluation metrics by comparing the detranslated text with the original text. Although the actual values for these metrics would be dependent on the model and its parameters, it would still be useful to compare the performance between different generated conlangs.

BLEU: Bilingual Evaluation Understudy

BLEU (Bilingual Evaluation Understudy) [Pap+02] is one of the most widely used automatic metrics for evaluating machine translations. It measures the similarity between a machine translations and reference translations by analyzing their *n-gram* overlap. The BLEU score is given by:

$$\text{BLEU} = \text{BP} \cdot \exp\left(\sum_{n=1}^{N} w_n \log p_n\right) \tag{2.1}$$

where p_n is the geometric average of the precision of n-grams, w_n are weights assigned to different n-grams, and BP (the brevity penalty) penalizes short translations to prevent artificially high scores. BLEU focuses primarily on precision but does not consider recall or semantic meaning.

ROUGE: Recall-Oriented Understudy for Gisting Evaluation

ROUGE (Recall-Oriented Understudy for Gisting Evaluation) [Lin04] is a set of metrics primarily used for text summarization but also applicable to machine translation. Unlike BLEU, which focuses on precision, ROUGE focuses on recall, i.e. how much of the reference translation appears in the generated text. The most common variant, ROUGE-N, computes the recall of n-gram matches:

$$\text{ROUGE-N} = \frac{\sum_{s \in \text{ref}} \sum_{gram_n \in s} \text{count}_{match}(gram_n)}{\sum_{s \in \text{ref}} \sum_{gram_n \in s} \text{count}(gram_n)} \tag{2.2}$$

Another variant, ROUGE-L, measures the longest common subsequence (LCS) between the reference and candidate translations, capturing fluency better. ROUGE is especially useful for evaluating translations with different word orders but similar meanings.

METEOR: Metric for Evaluation of Translation with Explicit ORdering

METEOR (Metric for Evaluation of Translation with Explicit ORdering) addresses some of BLEU's limitations by incorporating recall, stemming, synonym matching, and word order penalties. METEOR aligns words between candidate and reference translations using exact matches, stemmed matches, and synonym matches. The metric is computed as:

$$\text{METEOR} = F_{mean} \cdot (1 - Penalty) \tag{2.3}$$

where F_{mean} is the harmonic mean of precision and recall, and Penalty reduces the score for word order mismatches. METEOR can correlate better with human judgments than BLEU due to its ability to consider semantic variations.

2.3.2. Information Theoretic Evaluation of Languages

Natural language, being a means of communication consists primarily of information transmission [DB20]. There have been several attempts to quantify the information content of a language using information theory. Shannon [Sha51] introduced a new method of estimating the entropy and in turn the redundancy of a language. Entropy in a sense measures the average amount of information for each letter in a text. Natural language however, is not a minimal length source code. It operates under the constraints of incremental language processing, which enforces systematicity, compositionality and concatenation as means of combination [FH22]. Redundancy often serves another important purpose, which is to help in error correction and understanding in noisy environments [Gib+19].

2.3.3. Reading Comprehension Evaluation of Language Models

One of the primary goals of Natural Language Processing is teaching computers to read and understand the meaning of text. Reading comprehension thus presents itself as a suitable benchmark to evaluate a model's language understanding ability [Zen+20]. The goal of a reading comprehension task is to require the model to read one or more text passages and then answers questions about them.

MRC tasks are generally classified into four categories.

1. **Cloze Style** tasks require the model to fill in the blanks in a passage.
2. **Multiple-Choice** tasks require the model to select the correct answer from a set of options.
3. **Span Prediction** tasks require the model to extract a span of text from the passage that answers the question.
4. **Freeform** tasks require the model to generate a free-form answer to the question.

In our work, we decided to use Multiple-Choice tasks as a proxy for evaluating the constructed language. By evaluating the dataset on a model, first with the original passage and then with the detranslated passage, we get a proxy measure for the information loss in the translation process. Although some of this loss may be due to the model's inability to understand the constructed language, we can still use this for comparative analysis of different constructed languages.

RACE-C Dataset

The RACE-C Dataset [LLY19] is a large-scale reading comprehension dataset based on college english exams from China. It improves upon the previous RACE dataset

[Lai+17] by providing a more challenging set of questions and answers. It consists of 4275 passages and 14122 questions. Figure 2.1 shows a sample article and question from the dataset.

People have been painting pictures for at least 30,000 years. The earliest pictures were painted by people who hunted animals. They used to paint pictures of the animals they wanted to catch and kill. Pictures of this kind have been found on the walls of caves in France and Spain. No one knows why they were painted there. Perhaps the painters thought that their pictures would help them to catch these animals. Or perhaps human beings have always wanted to tell stories in pictures.
. . .
These days, we can write down a story, or record information, without using pictures. But we still need pictures of all kinds: drawing, photographs, signs and diagrams. We find them everywhere: in books and newspapers, in the street, and on the walls of the places where we live and work. Pictures help us to understand and remember things more easily, and they can make a story much more interesting.

Question: Pictures of animals were painted on the walls of caves in France and Spain because ________________.
Options:

A. the hunters wanted to see the pictures

B. the painters were animal lovers

C. the painters wanted to show imagination

D. the pictures were thought to be helpful

Correct Answer: D

Figure 2.1.: Sample Article and Comprehension Question from RACE-C

2.4. Language Models and Embeddings

2.4.1. Clustering of Embeddings

3. Methodology

3.1. Methodology

This chapter outlines the methodological approach we used to construct and evaluate constructed languages. The methodology follows the step by step approach from existing literature on language construction [Pet15; Ros10]. The language description contains the phonemic inventory, phonotactic rules, grammar and vocabulary of the new language. We also generate translation of an existing text, which can then be used for evaluation.

3.1.1. Research Design

We adopt a modular, computational experimental design. The overall aim is to investigate whether LLMs can be systematically guided to generate conlangs that exhibit properties found in natural human languages. To this end, a pipeline-based framework was developed, consisting of discrete modules for phonology, morphology, syntax, and vocabulary generation, followed by a suite of evaluation metrics. This modular design allows for targeted ablative analysis of each linguistic subsystem. This would also allow further extensions to the pipeline in the future.

3.1.2. Modular Language Generation Pipeline

The constructed language is generated through a modular pipeline, where each module can modify and append to the previous module's output. The design purposefully avoids requiring a strict ordering of modules, and each module can specify what it requires from the previous modules. Each module represents an independent variable in the generation process, which are manipulated to compare and contrast the generated languages, and answer the following research questions:

1. How does the number of phonemes in the phonemic inventory affect the generated language?

2. How do the phonotactic rules affect the generated language?

3. How do different grammatical structures affect the generated language?

4. How does different vocabulary generation methods affect the generated language?

3.1.3. Evaluation Methodology

Once the linguistic modules generate a language, we can run the evaluator modules to benchmark the results. Each evaluator module can run its evaluations and the results can be stored along with the language in the results. In particular, we run the following evaluations:

Information Loss Metrics

The Information loss metrics evaluate how well the generated language can retain information from the original text. The different benchmarks we use for this metric are:

1. **Machine Translation Scores**: We translate a text from English to the generated language and back to English, and evaluate the translation score using BLEU [Pap+02], ROUGE [Lin04] and METEOR [BL05]. The translation is done using a pre-trained LLM, and the scores are calculated using the generated text.
2. **Reading Comprehension Scores**: We evaluate the Reading Comprehension score of the generated text using a pre-trained LLM. The LLM is given a passage in both English and the generated language and a set of questions, and it is evaluated on how well it can answer the questions. The score is calculated as the percentage of questions answered correctly.

Simplicity Metrics

The simplicity metrics evaluate how simple the generated language is.

1. **BERT Fine-tuning**: We fine-tune a BERT like model on the generated language and evaluate the perplexity of the model.
2. Vocabulary Size
3. Phonemic Inventory
4. Syllable count

4. Implementation and Experimental Setup

In this chapter, we will describe the pipeline we have set up to generate constructed languages. The codebase is built in Python, and we used Poetry for Dependency Management. The pipeline is modular, to allow us to perform ablation studies.

4.1. Pipeline Overview

In order to generate constructed languages, we setup a modular pipeline. Figure 4.1 shows the Structure of a generation pipeline. Although we ended up having the modules in a specific order, the codebase is flexible enough that we can reorder modules, as long as the input to any module has all the features required for its execution. This is facilitated by the `LanguageDescription` class, which contains all the features of a language. This class is piped through the modules, and each module can add or modify features of the language. Each module also generates a results `dict` which is stored as a `JSON` file after the run. If the module requires a certain element of the description to be generated beforehand, it check for those features and can throw an error if they are not present.

A pipeline can be setup by subclassing the `Pipeline` class, which takes care of executing the modules and saving the results. Each module of the pipeline is a subclass of the `Module` class, which has an `execute` method that takes a `LanguageDescription` object and other optional arguments, and returns the modified `LanguageDescription` object and other optional results.

During the course of development, we found ourselves using more or less similar modules in most of our experiments. We have identified the following modules that are common to most setups:

4.1.1. Phonetics Modules

The goal of a Phonetics Module is to generate the phonemic inventory of the language. The module configures the `PhonemeDataInventory` class, which contains a list of `PhonemeData`. The phoneme segments for this class are based on PHOIBLE [MM19] segments, with a `GlyphID` corresponding to their database. For our purposes, we also

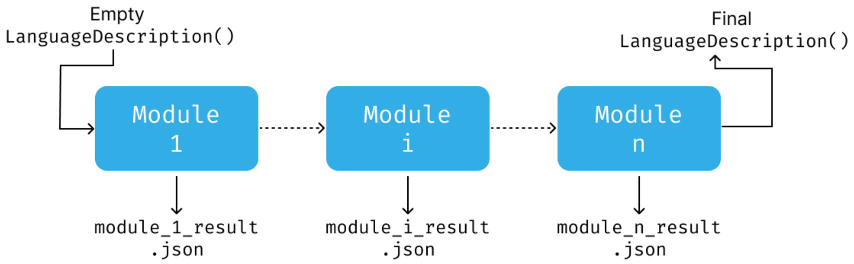


Figure 4.1.: The pipeline structure for generating constructed languages. The modules are run in order, and each module can add or modify features of the language.

implemented an `alphabet` attribute, to have a simpler representation for phonemes that are hard to read or write. This was useful for debugging and visualization purposes.

4.1.2. Phonotactics Modules

The goal of a Phonotactics Module is to generate the phonotactic rules of the language. The module configures the `PhonotacticData` class, which specifies the rules for syllable structure and phonotactic constraints for word beginnings and endings.

4.1.3. Syllable Builder Module

The Syllable builder modules combines the phonemes generated by the Phonetics Module and the phonotactic rules generated by the Phonotactics Module to generate all the possible syllables in the language.

4.1.4. Grammar Modules

4.1.5. Vocabulary Modules

The goal of the Vocabulary Module is to generate the vocabulary of the language. The module configures the `VocabDictionary` class, which holds the list of `VocabularyEntry` objects. Each `VocabularyEntry` object contains the word in the constructed language,

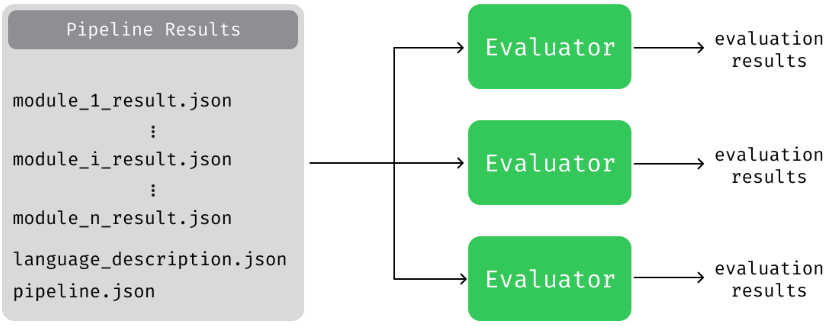


Figure 4.2.: The setup for running evaluations. Each evaluator stores the results in a folder inside the evaluations folder.

its translation, and its definition in english. A source words list is also part of each entry, to facilitate easy search for translation purposes. The class can also hold embeddings for each word, which could be also used for downstream tasks.

4.1.6. Translation Modules

Once the language description is generated, we use a translation module to translate some text corpora represented by `AbstractSourceText`. The module translates the source text paragraph by paragraph, using an LLM model.

4.1.7. Evaluation Modules

Figure 4.2 shows the setup for running evaluations on the generated languages. The Evaluators are a separate class that does not inherit from the abstract `Module` class. They inherit the `Evaluator` class, which creates an evaluations folder inside the results folder, to store the results of the evaluations.

5. Results and Analysis

6. Discussion, Conclusion and Future Work

7. Author Contributions

The codebase used for this thesis was developed by me in collaboration with Sara Visconti. I setup the initial repository and abstract classes for the modules and pipelines as well as the data classes for the language, which was then reviewed and refactored by Sara. Both of us modified or added to these modules and classes as needed based on experiment requirements. We also implemented specfiic modules for the experiments.

1. **Phonemes**: I implemented the RandomPhonemeModule, MostCommonPhonemeModule and LanguagePhonemeModule. I also implemented the BasicAlphabetModule.

2. **Phonotactics**: I implemented the BasicPhonotacticsModule, CustomPhonotacticsModule and SyllableBuilderModule.

3. **Grammar**: I implemented the BasicGrammarModule, BaselineAgglutinativeGrammarModule, and BaselineIsolatingGrammarModule. These modules built on top of Sara's Implementations of grammar features.

4. **Vocabulary**: I implemented the FixedVocabularyModule, FromSourceVocabularyModule, ClusterSimplifyVocabularyModule, ClusterTwoLevelVocabularyModule and ApproximatingVocabularyModule. I also used the Mapping modules implemented by Sara.

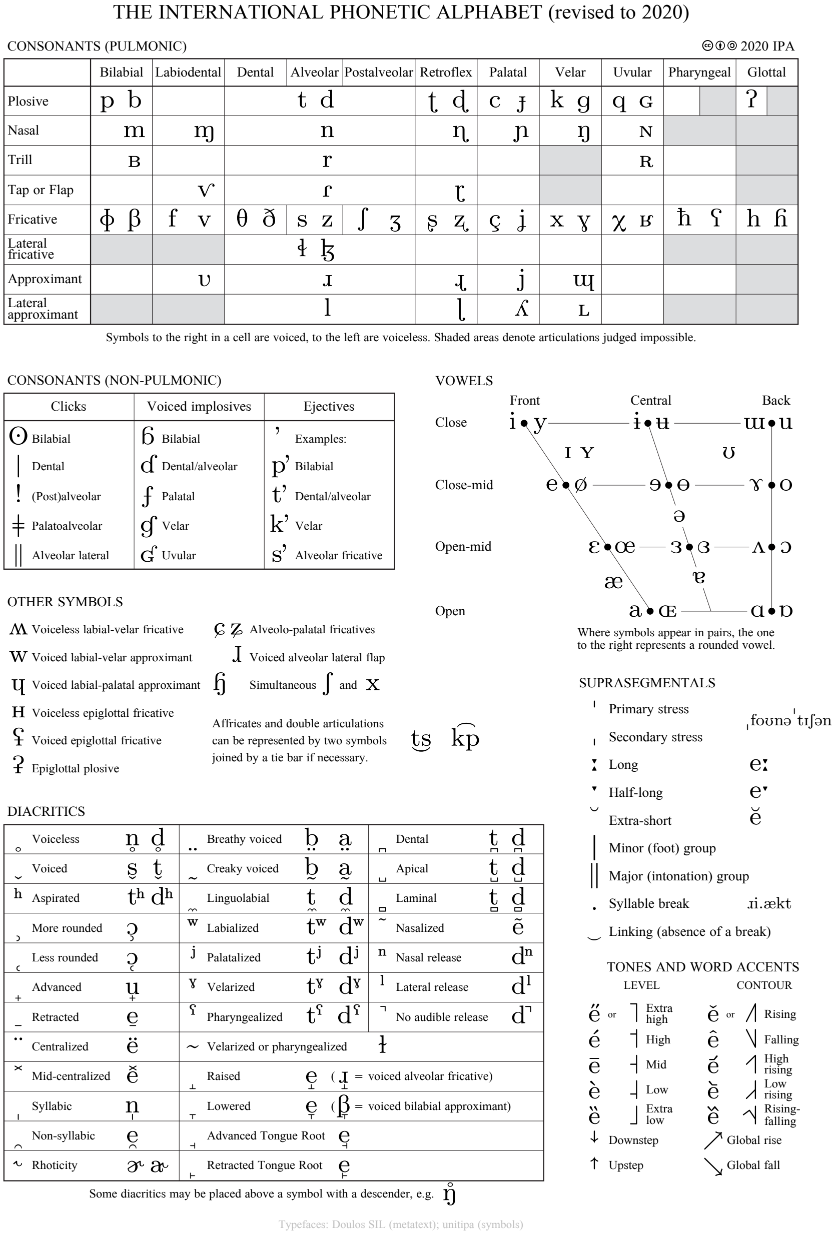
5. **Conlang**: I implemented the LLM based Source Text Translation Module.

6. **Evaluation**: I implemented the CompressionEvaluator, DetranslationEvaluator, HTML Summary Generator and RaceCEvaluator. I used Sara's Implementation of Zipfs Law Evaluator and BertEvaluator. Sara also added the various metrics to the DetranslationEvaluator.

7. **LLMs**: I implemented the LocalLLM and OpenAI implementation of the Remote LLM.

We jointly developed the conceptual design and methodology. Other contributions from Sara Visconti are not reported here since they are not mentioned in this work.

A. Appendix



THE INTERNATIONAL PHONETIC ALPHABET (revised to 2020)

CONSONANTS (PULMONIC) 2020 IPA

	Bilabial	Labiodental	Dental	Alveolar	Postalveolar	Retroflex	Palatal	Velar	Uvular	Pharyngeal	Glottal
Plosive	p b			t d		ʈ ɖ	c ɟ	k ɡ	q ɢ		ʔ
Nasal	m	ɱ		n		ɳ	ɲ	ŋ	ɴ		
Trill	ʙ			r					ʀ		
Tap or Flap		ⱱ		ɾ		ɽ					
Fricative	ɸ β	f v	θ ð	s z	ʃ ʒ	ʂ ʐ	ç ʝ	x ɣ	χ ʁ	ħ ʕ	h ɦ
Lateral fricative				ɬ ɮ							
Approximant		ʋ		ɹ		ɻ	j	ɰ			
Lateral approximant				l		ɭ	ʎ	ʟ			

Symbols to the right in a cell are voiced, to the left are voiceless. Shaded areas denote articulations judged impossible.

CONSONANTS (NON-PULMONIC)

Clicks	Voiced implosives	Ejectives
ʘ Bilabial	ɓ Bilabial	ʼ Examples:
ǀ Dental	ɗ Dental/alveolar	pʼ Bilabial
ǃ (Post)alveolar	ʄ Palatal	tʼ Dental/alveolar
ǂ Palatoalveolar	ɠ Velar	kʼ Velar
ǁ Alveolar lateral	ʛ Uvular	sʼ Alveolar fricative

VOWELS

Where symbols appear in pairs, the one to the right represents a rounded vowel.

OTHER SYMBOLS

ʍ Voiceless labial-velar fricative
w Voiced labial-velar approximant
ɥ Voiced labial-palatal approximant
ʜ Voiceless epiglottal fricative
ʢ Voiced epiglottal fricative
ʡ Epiglottal plosive

ɕ ʑ Alveolo-palatal fricatives
ɺ Voiced alveolar lateral flap
ɧ Simultaneous ʃ and x

Affricates and double articulations can be represented by two symbols joined by a tie bar if necessary. t͡s k͡p

SUPRASEGMENTALS

ˈ	Primary stress	ˌfoʊnəˈtɪʃən
ˌ	Secondary stress	
ː	Long	eː
ˑ	Half-long	eˑ
˘	Extra-short	ĕ
|	Minor (foot) group	
‖	Major (intonation) group	
.	Syllable break	ɹi.ækt
‿	Linking (absence of a break)	

DIACRITICS

̥	Voiceless	n̥ d̥	̤	Breathy voiced	b̤ a̤	̪	Dental	t̪ d̪
̬	Voiced	s̬ t̬	̰	Creaky voiced	b̰ a̰	̺	Apical	t̺ d̺
ʰ	Aspirated	tʰ dʰ	̼	Linguolabial	t̼ d̼	̻	Laminal	t̻ d̻
̹	More rounded	ɔ̹	ʷ	Labialized	tʷ dʷ	̃	Nasalized	ẽ
̜	Less rounded	ɔ̜	ʲ	Palatalized	tʲ dʲ	ⁿ	Nasal release	dⁿ
̟	Advanced	u̟	ˠ	Velarized	tˠ dˠ	ˡ	Lateral release	dˡ
̠	Retracted	e̠	ˤ	Pharyngealized	tˤ dˤ	̚	No audible release	d̚
̈	Centralized	ë	~	Velarized or pharyngealized	ɫ			
̽	Mid-centralized	e̽	̝	Raised	e̝ (ɹ̝ = voiced alveolar fricative)			
̩	Syllabic	n̩	̞	Lowered	e̞ (β̞ = voiced bilabial approximant)			
̯	Non-syllabic	e̯	̘	Advanced Tongue Root	e̘			
˞	Rhoticity	ɚ a˞	̙	Retracted Tongue Root	e̙			

Some diacritics may be placed above a symbol with a descender, e.g. ŋ̊

TONES AND WORD ACCENTS

LEVEL			CONTOUR		
e̋ or	˥	Extra high	ě or	˩˥	Rising
é	˦	High	ê	˥˩	Falling
ē	˧	Mid	e᷄	˦˥	High rising
è	˨	Low	e᷅	˩˨	Low rising
ȅ	˩	Extra low	e᷈	˧˦˧	Rising-falling
ꜜ	Downstep		↗	Global rise	
ꜛ	Upstep		↘	Global fall	

Typefaces: Doulos SIL (metatext); unitipa (symbols)

Figure A.1.: IPA Chart, available under a Creative Commons Attribution-Sharealike 3.0 Unported License. Copyright © 2018 International Phonetic Association.

Abbreviations

List of Figures

List of Tables

Bibliography

[24] "Ithkuil." In: *Wikipedia* (Aug. 2024).

[99] *Handbook of the International Phonetic Association : A Guide to the Use of the International Phonetic Alphabet*. 1. publ. Cambridge u.a.: Cambridge Univ. Press, 1999.

[BL05] S. Banerjee and A. Lavie. "METEOR: An Automatic Metric for MT Evaluation with Improved Correlation with Human Judgments." In: *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*. Ed. by J. Goldstein, A. Lavie, C.-Y. Lin, and C. Voss. Ann Arbor, Michigan: Association for Computational Linguistics, June 2005, pp. 65–72.

[DB20] Ł. Dębowski and C. Bentz. "Information Theory and Language." In: *Entropy (Basel, Switzerland)* 22.4 (Apr. 2020). ISSN: 1099-4300. DOI: 10.3390/e22040435.

[FH22] R. Futrell and M. Hahn. "Information Theory as a Bridge between Language Function and Language Form." In: *Frontiers in Communication* 7 (2022). ISSN: 2297-900X. DOI: 10.3389/fcomm.2022.657725.

[Gib+19] E. Gibson, R. Futrell, S. P. Piantadosi, I. Dautriche, K. Mahowald, L. Bergen, and R. Levy. "How Efficiency Shapes Human Language." In: *Trends in Cognitive Sciences* 23.5 (May 2019), pp. 389–407. ISSN: 1364-6613. DOI: 10.1016/j.tics.2019.02.003.

[Ha10] R. Ha. "Cost-Benefit Analysis in Animal Behavior." In: Jan. 2010, pp. 402–405.

[JM25] D. Jurafsky and J. H. Martin. *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition with Language Models*. 3rd. 2025.

[Lai+17] G. Lai, Q. Xie, H. Liu, Y. Yang, and E. Hovy. "RACE: Large-scale ReAding Comprehension Dataset From Examinations." In: *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. Ed. by M. Palmer, R. Hwa, and S. Riedel. Copenhagen, Denmark: Association for

Computational Linguistics, Sept. 2017, pp. 785–794. DOI: `10.18653/v1/D17-1082`.

[Lev22] N. Levshina. *Communicative Efficiency: Language Structure and Use*. Cambridge: Cambridge University Press, 2022.

[Lin04] C.-Y. Lin. "ROUGE: A Package for Automatic Evaluation of Summaries." In: *Text Summarization Branches Out*. Barcelona, Spain: Association for Computational Linguistics, July 2004, pp. 74–81.

[LLY19] Y. Liang, J. Li, and J. Yin. "A New Multi-choice Reading Comprehension Dataset for Curriculum Learning." In: *Proceedings of The Eleventh Asian Conference on Machine Learning*. PMLR, Oct. 2019, pp. 742–757.

[MM19] S. Moran and D. McCloy, eds. *Phoible 2.0*. Jena: Max Planck Institute for the Science of Human History, 2019.

[Pap+02] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu. "BLEU: A Method for Automatic Evaluation of Machine Translation." In: *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*. ACL '02. USA: Association for Computational Linguistics, July 2002, pp. 311–318. DOI: `10.3115/1073083.1073135`.

[Pet15] D. J. Peterson. *The Art of Language Invention : From Horse-Lords to Dark Elves, the Words behind World-Building*. New York, New York: Penguin Books, 2015. Chap. ix, 292 pages ; 22 cm. ISBN: 978-0-14-312646-1 0-14-312646-6.

[Ros10] M. Rosenfelder. *The Language Construction Kit*. Yonagu Books, 2010. ISBN: 978-0-9844700-0-6.

[Sch21] C. Schreyer. "Constructed Languages." In: *Annual Review of Anthropology* 50.Volume 50, 2021 (2021), pp. 327–344. ISSN: 1545-4290. DOI: `10.1146/annurev-anthro-101819-110152`.

[Sha51] C. E. Shannon. "Prediction and Entropy of Printed English." In: *The Bell System Technical Journal* 30.1 (Jan. 1951), pp. 50–64. ISSN: 0005-8580. DOI: `10.1002/j.1538-7305.1951.tb01366.x`.

[TS07] R. Trask and P. Stockwell. *Language and Linguistics: The Key Concepts*. Key Concepts Series. Routledge, 2007. ISBN: 978-0-415-41359-6.

[Yul20] G. Yule. *The Study of Language*. 7th ed. Cambridge: Cambridge University Press, 2020.

[Zen+20] C. Zeng, S. Li, Q. Li, J. Hu, and J. Hu. *A Survey on Machine Reading Comprehension: Tasks, Evaluation Metrics and Benchmark Datasets*. Oct. 2020. DOI: `10.48550/arXiv.2006.11880`. arXiv: `2006.11880 [cs]`.