2025

Toki Pona – Language Analysis and Modelling

Edith Susanna Andrews

15 January 2025

Contents

1. Introduction

Toki Pona is a minimalistic language that was invented by linguist Sonja Lang [1], so she would be able to simplify her thoughts [2]. The first drafts of the language were published online in 2001, and a revised version was published in the 2014 book “Toki Pona: The Language of Good” (also referred to as *lipu pu* in Toki Pona). She also released the Toki Pona Dictionary in 2021 (called *lipu ku*), which lists 137 essential words, but there are several other words that are in use. Most of these words are polysemous, or have multiple meanings, and can be used as multiple parts of speech. Due to its limited vocabulary, Toki Pona relies heavily on context to convey meaning.

The name “Toki Pona” translates to “simple language” or “the language of good”. According to the Sapir-Whorf hypothesis, language influences how its speakers think [3]. Toki Pona was intended to promote positive thinking and find joy in its simplicity [2]. It encourages users to distill complex concepts into their most essential elements without focusing on the details [4]. Although there have been many new words
suggested by the members of the community, most people adhere to the 137-word core, with which they are able to communicate effectively.

Traditional natural language processing systems are capable of handling large vocabularies and well-defined grammar rules, but the flexible syntactic structure and constrained lexicon of Toki Pona require a different approach. A core feature of the language is polysemy, which creates ambiguity that requires contextual analysis to interpret accurately. The implementation of a language model for Toki Pona can offer valuable insights into handling languages with unconventional linguistic frameworks.

The aim of this project is to create a custom spaCy language model for Toki Pona. The spaCy library provides tools for tasks like tokenization, dependency parsing, and part-of-speech tagging. Adapting these tools to Toki Pona involves addressing several challenges, including defining token boundaries, modeling polysemous words, and handling the language’s flexible grammar.

Commented [1]: https://tokipona.org/

Commented [2]: "Babel's modern architects". Los Angeles Times. 24 August 2007. Archived from the original on 3 January 2013. Retrieved 26 March 2022.

Commented [3]: Fabbri, Renato (July 2018). "Basic concepts and tools for the Toki Pona minimal and constructed language". ACM Transactions on Asian and Low-Resource Language Information Processing. arXiv:1712.09359.

Commented [4]: "Babel's modern architects". Los Angeles Times. 24 August 2007. Archived from the original on 3 January 2013. Retrieved 26 March 2022.

Commented [5]: Morin, Roc (15 July 2015). "How to Say (Almost) Everything in a Hundred-Word Language". The Atlantic. Archived from the original on 12 July 2022. Retrieved 1 August 2019.

2. Linguistic Features of Toki Pona

2.1 Phonology and Phonotactics

2.1.1 Phonemic Inventory

Toki Pona consists of only 14 phonemes. The 9 consonants (/p, t, k, s, m, n, l, j, w/) are pronounced as they are in English, and have the same value as in the International Phonetic Alphabet. The letter 'j' is pronounced like the English 'y', similar to the German or Swedish 'j' [5]. The vowels (/a e i o u/) have only one pronunciation, similar to the vowels in Spanish, Esperanto, and Japanese. Stress is always placed upon the first syllable of a word [6].

2.1.2 Phonotactics

Syllables in Toki Pona follow a (C)V(N) pattern - they begin with an optional consonant, a mandatory vowel, and an optional nasal consonant (N) at the end. Therefore, syllables can take on one of the following forms: V, CV, VN, and CVN [3]. If the syllable is not the initial syllable of a word, the consonant is mandatory. The most common syllable type is CV, making up 75% of syllables [7]. Some sequences of consonants and vowels are forbidden in Toki Pona. The combinations (/wu, wo, ji, ti/) are banned to avoid palatalization and make pronunciation easier for speakers of different linguistic backgrounds.

2.1.3 Allophony

The nasal at the end of a syllable does not have a fixed pronunciation, but changes depending on the consonant that follows it, which is known as nasal assimilation. The nasal sound adapts to match the place of articulation of the following consonant [8]. It occurs as an [n] before consonants produced at the alveolar ridge (/n t s l/), an [m] before bilabial consonants (/m p w/), as an [ŋ] before the velar sound /k/, and as an [ɲ] before the palatal sound /j/. Toki Pona does not allow a syllable's final nasal to occur before /m/ or /n/ in the same root word.

Due to the language's small phoneme inventory, Toki Pona allows for allophonic variation in the sounds. Voiceless stops like (/p t k/) can be pronounced as voiced stops (/b d g/), the voiceless fricative /s/ can be pronounced as /z/ or /ʃ/, and the lateral consonant /l/ can be pronounced as /r/. Vowels can also be pronounced long or short, without altering the meaning of the word [7].

Commented [6]: https://sona.pona.la/wiki/Phonology

Commented [7]: Lang, Sonja (2014). Toki Pona: The Language of Good. Tawhid. ISBN 978-0978292300. OCLC 921253340. pg 13

Commented [8]: Fabbri, Renato (July 2018). "Basic concepts and tools for the Toki Pona minimal and constructed language". ACM Transactions on Asian and Low-Resource Language Information Processing. arXiv:1712.09359.

Commented [9]: Blahuš, Marek (November 2011). Fiedler, Sabine (ed.). "Toki Pona: eine minimalistische Plansprache" [Toki Pona: A Minimalistic Planned Language] (PDF). Interlinguistische Informationen (in German). 18. Berlin: 51–55. ISSN 1432-3567. Archived (PDF) from the original on 27 June 2021. Retrieved 8 January 2019.

Commented [10]: https://sona.pona.la/wiki/Phonotactics

Commented [11]: Blahuš, Marek (November 2011). Fiedler, Sabine (ed.). "Toki Pona: eine minimalistische Plansprache" [Toki Pona: A Minimalistic Planned Language] (PDF). Interlinguistische Informationen (in German). 18. Berlin: 51–55. ISSN 1432-3567. Archived (PDF) from the original on 27 June 2021. Retrieved 8 January 2019.

2.1.4 Tokiponization

The process of adapting foreign words or names into the phonetic and structural constraints of Toki Pona is called tokiponization. This involves simplifying words to only include the 14 letters in Toki Pona's phoneme inventory, and modifying them to contain simple, open syllables. This process includes replacing sounds that do not exist in the language, eliminating consonant clusters, and truncating long words. The essence of the word should remain, while still being recognizable and pronounceable with Toki Pona's phonological system. Some names of nations, languages and religions have already been established, and it is generally better to use them for comprehensibility [6].

Commented [12]: Lang, Sonja (2014). Toki Pona: The Language of Good. Tawhid. ISBN 978-0978292300. OCLC 921253340.

2.2 Vocabulary

The vocabulary of Toki Pona has roots borrowed from several natural languages such as English, Tok Pisin, Finnish, Esperanto, Serbo-Croatian, Acadian French, Dutch, Georgian, Mandarin, Cantonese, Welsh, Tongan, Akan, Swahili, and Japanese [9]. In the 2014 book "Toki Pona: The Language of Good" (also known as *pu*), Sonja Lang describes 120 words. Before the book was released, the language was still in development, and the words suggested then which were not added to the *pu* were called "pre-*pu*" words. As more people began to learn and speak the language, new words were suggested and incorporated into its lexicon. Words that were added later are called "post-*pu*" words.

Commented [13]: Nimi ale pona

2.2.1 Colors

There are five root words for colors in Toki Pona: *pimeja* (black), *walo* (white), *loje* (red), *jelo* (yellow), and *laso* (blue/green). Other colors can be expressed as a combination of these colors. For example, *laso loje* (a reddish shade of blue) or *loje laso* (a bluish shade of red) could both be used to denote the color purple.

2.2.2 Numbers

The number system is not very elaborate in Toki Pona, as it has only a few number words: *ala* (zero), *wan* (one), *tu* (two), and *luka* (five). Higher numbers can be made by combining these numbers [10].

- 3: *tu wan* (2+1)
- 4: *tu tu* (2+2)
- 7: *luka tu* (5+2)

Commented [14]: Yerrick, Damian (23 October 2002). "Toki Pona li pona ala pona? A review of Sonja Kisa's constructed language Toki Pona". Pin Eight. Archived from the original on 28 September 2007. Retrieved 20 July 2007.

The word *mute* (many) can be used in place of larger numbers, and *ale* (all) can represent an infinite amount.

2.3 Morphology

Toki Pona is an isolating language, which means that words are not conjugated or inflected for number, gender, tense or case. Instead, words are invariant, and can function as different parts of speech depending on the context. For example, *mi moku* can either mean "I eat" or "I am food" [11].

Commented [15]: Tomaszewski, Zach (11 December 2012). "A Formal Grammar for Toki Pona" (PDF). University of Hawai'i. Archived (PDF) from the original on 1 November 2019. Retrieved 21 September 2019.

2.3.1 Modifiers

In Toki Pona, words can be modified by other words that follow it, known as modifiers. For example:

- *jan pona*, which literally translates to "person good", and can mean both "good person" and "friend".

The word *pona* (good) modifies the word *jan* (person). If a second modifier is added, both words before it are modified:

- *jan pona mute* ("many good people")

2.3.2 Negation

To negate a word, the word *ala* ("not") is appended to a verb or statement. For example:

- *mi lape ala* (I'm not sleeping)
- *lape ala* ("no" in response to "Are you sleeping?")

2.4 Syntax

Due to its simple sentence structure, Toki Pona relies heavily on particles to clarify relationships between sentence elements.

2.4.1 Sentence Structure

Sentences in Toki Pona typically follow a Subject-Verb-Object (SVO) structure. There is no verb "to be" in Toki Pona, but a particle *li* is used to separate the subject and the verb. This is only included when the subject is not *mi* (first person) or *sina* (second person). For example:

- *mi moku* ("I am eating")
- *jan li moku* ("The person is eating")

This particle can also be used to say that the subject does more than one thing: *jan li moku li toki* (The person is eating and talking).

Vocative phrases are marked with the particle *o* after the addressee. It can also be used before a verb as a second person imperative, or after the subject *mi* or *sina*, replacing *li* to express wishes [6].

Commented [16]: Lang pg 34

2.4.2 Direct Objects

The particle *e* is a predicate marker, which separates the direct object from the rest of the sentence.

- *jan li moku e kili* (“The person eats the fruit”)

2.4.3 Possession

The particle “pi” can be used similarly to “of” in English, to indicate possessives.

- *tomo pi jan pona* (“house of a good person” or “friend’s house”)
- *jan pi pona mute* (“person of much goodness” or “very good person”)

2.4.4 Coordination

Multiple subjects in a sentence can be separated by *en*:

- *moku en lape li pona* ("Food and sleep are good")

2.4.5 Context

Context can be added to sentences by prepending another sentence followed by *la* [12].

Commented [17]: https://blinry.org/toki-pona-cheat-sheet/toki-pona-cheat-sheet.pdf

- *mi lape la ali li pona* ("When I’m asleep, everything is okay").

This is also useful to denote time.

- *tenpo ni la mi lape* ("I am sleeping now")
- *tenpo kama la mi lape* ("I will sleep in the future")
- *tenpo pini la mi lape* (“I slept in the past”)

Since Toki Pona does not have a way to explicitly specify tenses, these temporal references are helpful context to understand when the events take place.

2.4.6 Questions

- Questions can be framed by replacing the verb in a sentence with the “(verb) ala (verb)” format. For example, *sina moku ala moku?* translates to “Are you eating?” This can be responded to with *moku* (yes) or *moku ala* (no).
- Open-ended questions use *seme* as a placeholder: *sina toki e seme?* ("What are you saying?").

2.5 Semantics

Words in Toki Pona are often polysemous. A single word can function as a noun, verb, or adjective based solely on its placement in a sentence. For example, the word *toki* has multiple definitions:

- As a noun: *toki* means “language”, “speech”, or “communication”. It is used before the tokiponized name of a language to denote the language itself, such as *toki Inli* (English).
- As an adjective, it can mean “verbal” or “speaking”.
- As a verb: “to say”, “to speak”, “to communicate”, and even “to think”.
- As an interjection: *toki!* is a greeting, like “hello”.

The exact meaning of each word can be inferred through context and word combinations.

2.5.1 Proper Nouns

In Toki Pona, proper nouns are adapted and classified using generic descriptors to convey meaning while also adhering to the language’s minimalist principles. The tokiponized names of people, places and entities are preceded by a classifier like *ma* (land/country/place), *ma tomo* (town/village/city), *jan* (person), *toki* (language), *soweli* (animal), and so on. For example:

- *jan Lisa* (a person named Lisa)
- *ma Mewika* (America)
- *ma tomo Pelin* (Berlin)

Most words in Toki Pona have large semantic spaces [13], which is the range of possible meanings of a word. Rather than having a large number of specialized terms, Toki Pona uses a small set of words that can be expanded upon through context and compounding with modifiers. Being constrained to this set of words forces the speaker to be creative and think outside the box when trying to describe something complex.

Commented [18]: https://lipamanka.gay/essays/dictionary

2.6 Pragmatics and Sociolinguistics

Pragmatics is the study of how context influences meaning, and it plays a significant role in Toki Pona. Sentences can be disambiguated by taking context into account. Context can include anything that is not present in the sentence, like where the sentence was said, who said it, how it was said, and so on. If something remains ambiguous, follow-up questions can be asked to confirm what the speaker means to say. A lot of details tend to be unimportant, which encourages speakers to think carefully about what they really want to convey.

Toki Pona also doesn't have a direct way to express politeness, but it is considered to be the default state. There are no words for "thank you" or "please", as it is always assumed that the speaker has positive intentions. Phatic phrases are also generally avoided, as they don't convey deep meaning. Instead, importance is placed on language that encourages communication, rather than simply making small talk. There is also no distinction between formal and informal styles of speech. However, the usage of *nimi sin* (new words) that are not part of the core Toki Pona vocabulary are not preferred in formal settings. This is due to the desire to maintain the simplicity of Toki Pona and ensure that it is widely understood.

3. Writing Systems

The 14 Latin letters (a e i j k l m n o p s t u w) are used to write the language. Capital letters are used for proper nouns, but all other words are written with lowercase letters, even when beginning a sentence. Apart from the Latin alphabet (*sitelen lasina*), two major logographic writing systems were included in "Toki Pona: The Language of Good", namely *sitelen pona* and *sitelen sitelen* [6].

Commented [19]: Lang 2014, p. 96.

3.1 Sitelen Pona

In *sitelen pona* ("good/simple writing"), every word is represented by a single grapheme. Most of the characters are derived from universal symbols, and are easy to recognize and remember. A number of these glyphs contain a combination of graphical components known as radicals [14]. This radical based system was created as an input method for entering toki pona words with the Wakalito keyboard layout [15]. The radicals are as follows:

Commented [20]: https://sona.pona.la/wiki/Radicals_in_sitelen_pona

Commented [21]: https://sona.pona.la/wiki/Wakalito

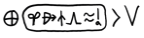
Table 1 Radical Based System in Sitelen Pona

Description	Sitelen Pona glyph	Toki Pona word	English translation
Arrow	↓	*ni*	this, these
Arrowhead	>	*li*	particle to separate subject and verb
Box (closed)	□	*lipu*	book
Box (open)	⊔	*poki*	container

Circle (thing)	○	*ijo*	thing
Circle (head)	◌	*jan*	person, people, humanlike
Cross	×	*ala*	no, not
Dot (disambiguator)	ᛁ	*seli*	hot, heat
Dot (mouth)	ᵔᵔ	*suwi*	cute, soft, sweet
Dots (eyes)	⁜	*pipi*	Insect, bug
Emitters	☌	*toki*	speak, say, talk, communicate
Hammer	⊞	*ilo*	device, tool, mechanical, instrument
Hand (flat)	∩	*luka*	arm, hand, five
Hand (pointing)	ᓚ	*mi*	I, us, me, we, my, our
Heart	♡	*pilin*	feel, feeling, heart, emotion
Mouth (closed)	◡	*pona*	good, safe, help, fix, well
Mouth (open)	ᗑ	*moku*	eat, consume, drink
Legs	⅄	*tawa*	to, go (to), towards, for, walk, moving, leave, depart, move, transport
Punctuation stem	ʔ	*seme*	what, which
Tally mark	|||	*mute*	many, very, a lot
Triangle (color)	△	*kule*	color, paint, colorful
Wavy line	ᔓ	*linja*	rope, hair, stalk (of plant)

There are also multiple glyphs that do not have any accepted radical, which require various different combinations of radicals on the Wakalito layout.

Proper names are written by enclosing multiple characters in a rounded rectangle-shaped cartouche. Each character represents the first letter of its word. The words inside the cartouche may be chosen to creatively convey some meaning about the subject. For example, *ma Kanata li suli* (“Canada is big”) can

be written in Sitelen Pona with the name *Kanata* using the words *kasi alasa nasin awen telo a* as follows [16]: 

Glyphs for adjectives may be written inside or above the symbol for the word that they modify. For example, *pilin ike* ("feeling bad") can be written like  or like  .The logo for Toki Pona is also written in this way: 

Since the punctuation is not standardized, sentences in Sitelen Pona are usually separated with interpuncts, periods, line breaks, or wide spaces. Exclamatory marks and question marks are avoided due to their similarity with the glyphs for *o* and *seme* [17].

3.2 Sitelen Sitelen

Jonathan Gabel later created a non-linear writing system using hieroglyphic blocks with components that correspond to syllables called Sitelen Sitelen or Sitelen Suwi. This system was not designed to be efficient or practical to type on a computer. The writer is encouraged to slow down and play with their thoughts while writing [18]. The aesthetics of the characters of Sitelen Sitelen were inspired by characters of different writing systems, including Egyptian hieroglyphs, Linear B, Chinese characters, Maya Script

There are two methods to form words in Sitelen Sitelen. The first method is creating images that represent syllables (alphasyllabary). Each syllable has an initial consonant container, a middle vowel infix, and the optional n subfix. If there is no consonant, a simple circle is drawn.

									
( )	j	k	l	m	n	p	s	t	w

					
a	e	i	o	u	n (subfix)

Commented [22]: https://en.wikipedia.org/wiki/Sitelen_Pona

Commented [23]: https://sona.pona.la/wiki/sitelen_pona

Commented [24]: https://jonathangabel.com/toki-pona/kama-pona/

Using these glyphs, we can arrive at all of the possible (C)V(N) combinations. Multi-syllable words can be created by simply adding syllables together. However, since this is a non-linear writing system, we can use a capsule glyph to group syllables into words without the use of spaces [19][20].

						
to	*ki*	*po*	*na*	*toki*	*pona*	*toki pona*

The second method involves using images that represent whole words (logograms). The word glyphs are used in various combinations to create sentences [21].

		
toki	*pona*	*toki pona*

3.3 Other Writing Systems

Some members of the Toki Pona community have created other writing systems [22], for creative or aesthetic expression, ease of use, or simply for fun. Some of these systems use the scripts of other natural languages, like Hiragana (*sitelen Ilakana*), Hanzi (*sitelen Kansi*), Cyrillic (*sitelen Kililisa*), Hangul (*sitelen Anku)*, Thai (*sitelen Tawi*), and Devanagari (*sitelen Tewanakali*). Some of them modify or improve upon the *sitelen pona*, like *sitelen Antowi*, *sitelen kalama*, *sitelen lili*, *sitelen luka*, and *sitelen pona pona*. There have been attempts to represent the *sitelen pona* with unicode characters and emoticons, like *sitelen Enli / sitelen pi toki pona*, *sitelen Juniko*, *nasin Unikote*, *sitelen Aki*, *sitelen pilin*, and *nasin pi sitelen jelo*. Some take inspiration from fictional languages and styles, such as *sitelen mun* (based on Gallifreyan), *sitelen pipi* (inspired by Hollow Knight), and *sitelen Tenwa* (the Tengwar script from the Lord of the Rings) [22].

Commented [25]: https://jonathangabel.com/toki-pona/syllables-1/

Commented [26]: https://jonathangabel.com/toki-pona/syllables-2/

Commented [27]: https://jonathangabel.com/toki-pona/basic-sentences/

Commented [28]: https://sona.pona.la/wiki/Writing_systems

4. Modelling Toki Pona in SpaCy

spaCy [23] is an open-source library for advanced natural language processing (NLP) in Python. It is designed to be fast, efficient, and easy to use, and it offers a wide range of NLP tools and functionalities for processing and analyzing large volumes of text data. spaCy is widely used in both research and production environments for tasks such as text processing, information extraction, and machine learning.

The current implementation uses this library as the base framework to model the Toki Pona language based on the following linguistic feature extraction techniques:

- Tokenization
- Word Vectors
- Part-of-Speech (POS) Tagging
- Dependency Parsing
- Named Entity Recognition

4.1 Tokenization

Tokenization is the process of converting text into smaller units, called "tokens," which can be individual words, sub words, or characters. In natural language processing (NLP), these tokens are used as the basic building blocks for analyzing and understanding the structure of language. Tokenization allows algorithms to process text data more efficiently.

There are several types of tokenization:

- Word Tokenization: Splitting a text into individual words. For example, the sentence "ona li toki pona" would be tokenized into [“ona”, “li”, “toki”, “pona”].
- Sub word Tokenization: Breaking text down into smaller meaningful parts like prefixes, suffixes, or roots. This is useful for handling unknown or rare words.
- Character Tokenization: Splitting the text into individual characters. For instance, "soweli" would become [“s”, “o”, “o”, “w”, “e”, “l”, “I”].
- Sentence Tokenization: Dividing text into sentences rather than words. For example, "ni li jan Pesi. ona li jan pali pona mi." would be tokenized into [“ni li jan Pesi.”, “ona li jan pali pona mi.”].

Tokenization is a critical first step in many NLP tasks, such as text classification, machine translation, and sentiment analysis.

4.1.1 Toki Pona Tokenizer

The process of tokenization in Toki Pona is fairly straightforward because of the language's minimalistic and isolating structure. Each word usually corresponds to a single token without much further breakdown into smaller sub word components. It involves identifying the core words or morphemes (smallest units of meaning), which generally correspond to its simple vocabulary. For example:

- "mi en sina" → ["mi", "en", "sina"]
    - "mi" = I
    - "en" = and
    - "sina" = you

The sentence "mi en sina" is tokenized into the individual words: "mi", "en", and "sina".

4.2 Word Vectors

Word2Vec [24] is a popular technique in natural language processing (NLP) for learning vector representations of words. It was developed by a team led by Tomas Mikolov at Google in 2013 and is designed to capture the semantic meaning of words based on their context in large text corpora. Word2Vec represents words as continuous-valued vectors in a high-dimensional space, where similar words have similar vector representations.

Word2Vec embeddings are used in various NLP tasks like sentiment analysis, text classification, machine translation, and information retrieval. The vectors allow for calculating the similarity between words using metrics like cosine similarity, which is useful for tasks like finding synonyms or clustering words with similar meanings.

Word2Vec is trained using a large corpus of text. The model adjusts the word vectors based on how words are used in similar contexts across the text. It uses techniques like negative sampling or hierarchical softmax to efficiently compute the vectors. The resulting word vectors are in a continuous vector space, where each word is represented by a dense, fixed-length vector (e.g., 100, 200, or 300 dimensions). These vectors are typically more efficient and meaningful than traditional one-hot encoding, which is sparse and high-dimensional.

4.2.1 Toki Pona Word Vectors

The current implementation uses a pretrained set [25] of Word2Vec vectors trained on corpora consisting of transliterated works by the community and examples from The Language of Good by Sonja Lang. This was converted to a SpaCy compatible format using the custom converter provided by the library.

4.3 Part of Speech Tagging

Part of Speech (POS) Tagging is the process of assigning a grammatical category or label to each word in a sentence. Each word is classified into a particular part of speech based on its role or function in the sentence. Common parts of speech include:

- Noun (NOUN): A person, place, thing, or idea (e.g., cat, dog, city).
- Verb (VERB): An action or state of being (e.g., run, eat, is).
- Adjective (ADJ): A word that describes or modifies a noun (e.g., happy, red).
- Adverb (ADV): A word that modifies a verb, adjective, or another adverb (e.g., quickly, very).
- Pronoun (PRON): A word that takes the place of a noun (e.g., he, they, it).
- Preposition (ADP): A word that shows relationships between other words (e.g., in, on, at).
- Conjunction (CCONJ): A word that connects words, phrases, or clauses (e.g., and, but, or).
- Interjection (INTJ): A word or phrase that expresses strong emotion or surprise (e.g., wow, oops).
- Determiner (DET): A word that introduces a noun and specifies it (e.g., the, a, some).

For illustrating this concept, we can consider the following sentence:

"mi toki pona."

This means "I speak good" or "I speak well" in English.

A POS tagger would assign labels to each word:

- mi → Pronoun (PRON): "mi" is the first-person pronoun in Toki Pona, meaning "I" or "me."
- toki → Verb (VERB): "toki" is a verb in Toki Pona meaning "to speak" or "talk."
- pona → Adjective (ADJ): "pona" is an adjective in Toki Pona meaning "good" or "simple," and in this case, it modifies the verb "toki," giving the meaning of speaking well.

POS tagging is a foundational step in many natural language processing (NLP) tasks. It helps us to understand the structure of a sentence by analyzing the grammatical relationships between words. It can also be used to identify and categorize proper names and entities (like people, locations, etc.) often requires POS tagging to distinguish nouns and other specific word types.

Knowing the role of each word helps in generating grammatically coherent summaries or translations. Some words can have multiple meanings depending on their part of speech. For instance, "moku" can be a verb (to eat) or a noun (food). This is especially important for polysemous languages like Toki Pona where the context of the entire sentence is crucial in generating a correct translation.

POS Tagging can be performed using several methods:

- Rule-based Tagging: Involves applying a set of predefined rules to assign POS tags. For instance, a rule might state that if a word follows a determiner (like "the"), it's likely to be a noun.
- Statistical Tagging: Uses machine learning algorithms to predict POS tags based on probabilistic models. These models are trained on large annotated corpora.
- Deep Learning-based Tagging: More recent methods use neural networks and deep learning to automatically learn patterns in language data, improving accuracy.

4.3.1 Training POS Tagger

The current implementation uses an existing parser [26] based on SWI-Prolog and Definite Clause Grammars (DCG) to parse the sentences based on rule matching. The presented implementation is written in Python and the parser is written in Prolog. Moreover, the output generated by the parser does not follow a standard convention. Hence, the parsed output of a selected set of sentences have been generated and saved in text format. ChatGPT 4o has been used to convert the output of the parser to the standard CONNL-U format.

The CONLL-U format is commonly used for representing syntactic annotation (such as POS tagging and dependency parsing) in a structured way. It typically consists of several columns, including the word form, its lemma, its part of speech (POS), and its syntactic dependencies. This is an example sentence and its generated output:

Sentence: "mi toki pona."

Parser Output:

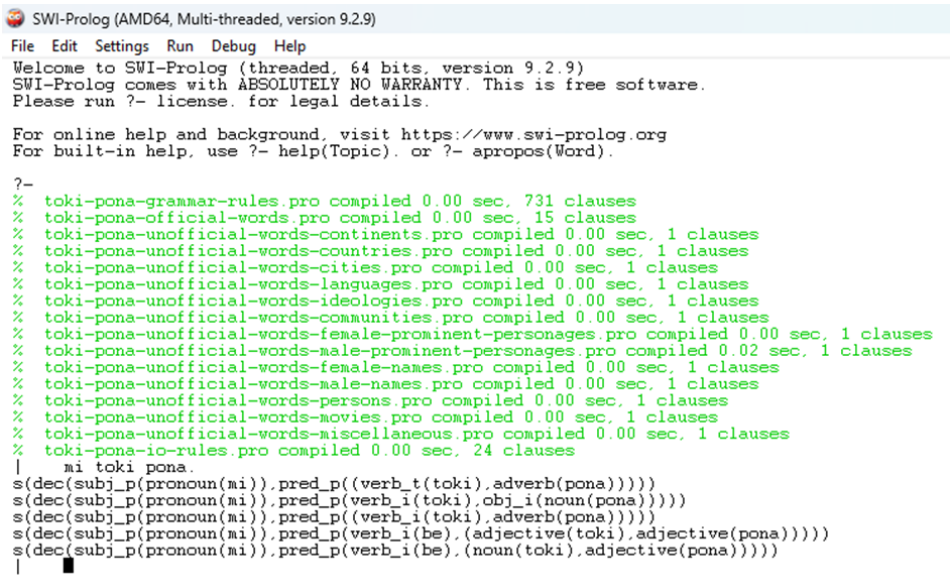


Figure 1 Toki Pona Parser Output in SWI-Prolog Environment

As observed in the output generated by the parser, a single sentence has generated five possible variations for the same sentence. This reflects the polysemous nature of the language and the difficulty in disambiguating a sentence. Helper scripts have been used to save this output to a text file and record the first parsed output that is generated.

ChatGPT 4o CONNL-U Output:

This is a comment line

1 mi mi PRON _ _ 0 nsubj _ _

2 toki toki VERB _ _ 1 root _ _

3 pona pona ADJ _ _ 2 amod _ _

Here is an explanation of the CONLL-U columns:

- ID: The position of the word in the sentence.
    - "mi" is the 1st word, "toki" is the 2nd, and "pona" is the 3rd.
- Form: The actual word as it appears in the sentence.
    - "mi" (I), "toki" (speak), and "pona" (good/well).
- Lemma: The base or dictionary form of the word.
    - "mi", "toki", and "pona" are the same as their lemma forms in this case. This is due to the simplistic nature of the language itself and will be the case for all sentences.
- POS (Part of Speech): The grammatical category or part of speech.
    - "mi" is a pronoun (PRON), "toki" is a verb (VERB), and "pona" is an adjective (ADJ).
- Feats: Features (usually left as "_" when not specified).
    - The current implementation does not consider any features since Toki Pona hardly has any.
- Head: The ID of the word's syntactic head. This shows the word to which the current word is syntactically attached.
    - "mi" (the subject) has no head and is the starting point, so its head is marked as 0 (indicating the root).
    - "toki" is the main verb (root), so it points to "mi" (the subject) with 1.
    - "pona" is an adjectival modifier of the verb "toki", so it attaches to "toki" (the verb) with 2.
- DepRel: The syntactic relationship between the word and its head.
    - "mi" is the subject, so the relation is subject.
    - "toki" is the root of the sentence (main verb), so the relation is root.
    - "pona" is an adjectival modifier of the verb "toki", so the relation is amod (adjectival modifier).
- Deps: Dependencies, often left as "_" unless there are specific sub-relations to describe. The current implementation does not consider these.
- Misc: Miscellaneous information, often left as "_".

These tables have been generated for all sentences in the dataset and saved in the CoNNL-U format. The CoNNL-U files have been used for training the POS tagger component of the implementation's NLP pipeline.

4.4 Dependency Parser

A dependency parser is a tool used in Natural Language Processing (NLP) to analyze the grammatical structure of a sentence. It identifies the syntactic dependencies between words in a sentence, showing how each word is related to others in terms of grammatical functions (such as subject, object, modifier, etc.). The model can be trained for this along with the POS tagging training as it can be described the CONNL-U table format.

- Dependency Relations: These are the grammatical relationships between words. For example, in the sentence "I eat pizza," a dependency parser would recognize that "I" is the subject of the verb "eat," and "pizza" is the object of "eat."
    - This is described by the DepRel column of the CoNNL-U format.
- Head Word: In a syntactic structure, the head is the word that governs other words. For example, in the sentence "She sings beautifully," "sings" is the head word because it is the main action, and "She" (subject) and "beautifully" (adverbial modifier) depend on it.
    - This is described by the Head column of the CoNNL-U format.
- Dependent Words: These are words that depend on the head word. For instance, in "She sings beautifully," both "She" and "beautifully" are dependent on "sings."

A dependency parser works by taking a sentence as input, typically in the form of a sequence of words. It processes the sentence by analyzing its syntax and determining the grammatical relationships between words based on their roles and positions. The output is a tree or graph structure where each word is connected to its head word, with labeled edges that describe the grammatical relationships, such as subject, object, or modifier. This structure helps in understanding the sentence's syntactic dependencies and overall meaning.

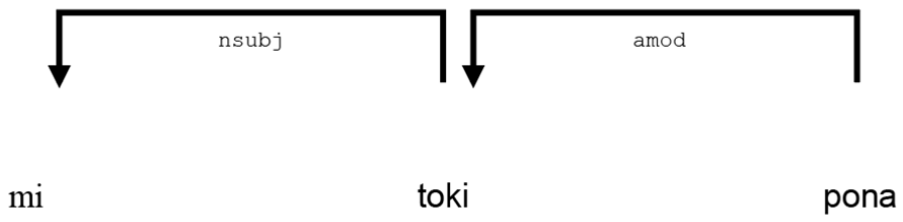


Figure 2 Output generated by Dependency Parser

4.5 Named Entity Recognition

Named Entity Recognition (NER) is a key task in Natural Language Processing (NLP) that involves identifying and classifying named entities in text into predefined categories. These categories typically include people, organizations, locations, dates, times, quantities, and other specific entities that hold particular meaning within a given context. NER helps machines understand the structure of text by identifying important information that can be extracted or analyzed further.

The process involves inputting a sentence or text into an NER system, which then scans and identifies words or phrases that correspond to named entities, classifying them using linguistic rules, statistical models, or deep learning techniques. The output is a labeled result where entities are categorized.

The current implementation uses rule-based matching since the official Toki Pona words (i.e., the 120 words published in Sonja Lang's book, Toki Pona: The Language of Good) are never capitalized. They are lowercase even at the beginning of a sentence. Hence, it is easy to categorize words with the first letter capitalized as named entities and differentiating them based on the preceding tokens. The following rules have been set to identify named entities:

- People’s names: Often borrowed from other languages (e.g., "jan Li" for "Jan Li").
- Places: Locations can be mentioned using context or external names (e.g., "ma tomo" for a particular place).
- Organizations or things: These could be referenced with combinations of words (e.g., "kulupu sona" for an educational group or "kili palisa" for a specific fruit or object).

5. Future Work

5.1 Word Sense Disambiguation

Toki Pona has a small vocabulary, but many words have multiple meanings depending on the context. This creates the potential for Sense2Vec to be highly useful in distinguishing different senses of these words. For example:

- "jan" can mean "person," "someone," or refer to a role, depending on the context.
- "moku" can mean "food," "to eat," or "meal," depending on its use in a sentence.
- "kule" can mean "color," but can also refer to something related to a specific context, like a specific color or an object associated with color.

Sense2Vec[EA5] is an extension of the popular Word2Vec model that addresses the problem of word ambiguity in natural language processing (NLP). While Word2Vec represents words as vectors (embeddings) based on their context in large text corpora, it does not distinguish between different meanings (senses) of the same word. Sense2Vec, on the other hand, disambiguates word senses by treating words in context differently.

5.2 Hybrid Approach for POS Tagging and Dependency Parsing

One powerful method for analyzing languages like Toki Pona is Context-Free Grammar (CFG), which provides a way to model syntactic structures using a set of production rules. This is similar to the Definite Clause Grammar system that was used in the Toki Pona parser used to generate the training dataset for the current implementation.

While CFGs are effective for capturing grammatical relationships, they have limitations when it comes to handling the complexities of Toki Pona's context-sensitive word usage. This is where a hybrid approach[EA6] —combining both rule-based (formal grammar) and model-based techniques (such as machine learning) for tasks like POS tagging and dependency parsing—can provide a more robust solution. A hybrid system can take advantage of both the structure and precision of formal grammar and the adaptability of data-driven models, such as machine learning algorithms, to account for linguistic nuances. This type of hybrid model is especially important for low-resource languages like Toki Pona which lack well-annotated datasets and pre-training methods.

6. Future Work

[1] “Toki Pona (official site).” https://tokipona.org (accessed Jan. 10, 2025).

[2] Dance, “Babel’s modern architects - Los Angeles Times,” *Los Angeles Times*, Mar. 02, 2019. Accessed: Jan. 10, 2025. [Online]. Available: https://www.latimes.com/archives/la-xpm-2007-aug-24-sci-conlang24-story.html

[3] R. Fabbri, “Basic concepts and tools for the Toki Pona minimal and constructed language: description of the language and main issues; analysis of the vocabulary; text synthesis and syntax highlighting; Wordnet synsets,” *arXiv (Cornell University)*, Jan. 2017, doi: 10.48550/arxiv.1712.09359.

[4] R. Morin, “Toki Pona: a language with a hundred words,” *The Atlantic*, Jul. 15, 2015. Accessed: Jan. 10, 2025. [Online]. Available: https://www.theatlantic.com/technology/archive/2015/07/toki-pona-smallest-language/398363/

[5] “Phonology - sona pona,” *Sona Pona*, May 13, 2024. https://sona.pona.la/wiki/Phonology (accessed Jan. 10, 2025).

[6] S. Lang, *Toki pona: The Language of Good*. 2014.

[7] M. Blahuš, “Toki Pona: eine minimalistische Plansprache: Toki Pona: A Minimalistic Planned Language,” *Interlinguistische Informationen (in German)*, vol. 20, pp. 51–56, 2011, [Online]. Available: https://media.interlinguistik-gil.de/beihefte/18/beiheft18.pdf#page=51

[8] “Phonotactics - sona pona,” *Sona Pona*, Oct. 20, 2024. https://sona.pona.la/wiki/Phonotactics (accessed Jan. 10, 2025).

[9] C. R. Moniz, S. H. Van Der Muelen, and Jan Lipamanka, “Nimi ale pona (2nd ed.),” Oct. 08, 2020. https://docs.google.com/spreadsheets/d/1t-pjAgZDyKPXcCRnEdATFQOxGbQFMjZm-8EvXiQd2Po/edit?usp=sharing (accessed Jan. 10, 2025).

[10] D. Yerrick, “Toki Pona li pona ala pona? A review of Sonja Kisa’s constructed language Toki Pona,” *Pin Eight*, Oct. 23, 2022. https://pineight.com/tokipona/tpreview.html (accessed Jan. 15, 2025).

[11] Z. Tomaszewski, “A Formal Grammar for Toki Pona,” University of Hawai‘i, Nov. 2012. Accessed: Jan. 09, 2025. [Online]. Available: http://www2.hawaii.edu/~chin/661F12/Projects/ztomaszewski.pdf

[12] “Toki pona cheat sheet,” *Blinry*, 2001. Accessed: Jan. 15, 2025. [Online]. Available: https://blinry.org/toki-pona-cheat-sheet/toki-pona-cheat-sheet.pdf

[13] Lipamanka, “The Semantic Spaces Dictionary.” https://lipamanka.gay/essays/dictionary (accessed Jan. 15, 2025).

[14] “Radicals in sitelen pona - sona pona,” *Sona Pona*, Jan. 05, 2025. https://sona.pona.la/wiki/Radicals_in_sitelen_pona (accessed Jan. 15, 2025).

[15] “Wakalito - sona pona,” *Sona Pona*, Sep. 30, 2024. https://sona.pona.la/wiki/Wakalito

[16] Wikipedia contributors, “Sitelen pona,” *Wikipedia*, Dec. 31, 2024. https://en.wikipedia.org/wiki/Sitelen_Pona (accessed Jan. 15, 2025).

[17] “sitelen pona - sona pona,” *Sona Pona*, Oct. 05, 2024. https://sona.pona.la/wiki/sitelen_pona (accessed Jan. 15, 2025).

[18] J. Gabel, “Welcome - kama pona,” 2024. https://jonathangabel.com/toki-pona/kama-pona/ (accessed Jan. 15, 2025).

[19] J. Gabel, “Syllables Part 1: Introducing syllable glyphs,” 2024. https://jonathangabel.com/toki-pona/syllables-1/ (accessed Jan. 15, 2025).

[20] J. Gabel, “Syllables Part 2: Combining syllables and writing names,” 2024. https://jonathangabel.com/toki-pona/syllables-2/ (accessed Jan. 15, 2025).

[21] J. Gabel, “Basic sentences,” 2024. https://jonathangabel.com/toki-pona/basic-sentences/ (accessed Jan. 15, 2025).

[22] “Writing systems - sona pona,” *Sona Pona*, Jan. 10, 2025. https://sona.pona.la/wiki/Writing_systems (accessed Jan. 15, 2025).

[23] A. Trask, P. Michalak, and J. Liu, “sense2vec - A Fast and Accurate Method for Word Sense Disambiguation In Neural Word Embeddings,” *arXiv.org*, Nov. 19, 2015. https://arxiv.org/abs/1511.06388.

[24] Explosion, “GitHub - explosion/spaCy: 💫 Industrial-strength Natural Language Processing (NLP) in Python,” *GitHub*. https://github.com/explosion/spaCy (accessed Jan. 15, 2025).

[25] T. Mikolov, K. Chen, G. Corrado, and J. Dean, “Efficient estimation of word representations in vector space,” *arXiv.org*, Jan. 16, 2013. https://arxiv.org/abs/1301.3781

[26] C. Martens, “GitHub - chrisamaphone/nimi2vec: Toki Pona Word Vector Embeddings,” *GitHub*. https://github.com/chrisamaphone/nimi2vec (accessed Jan. 15, 2025).

[27] A. Trask, P. Michalak, and J. Liu, “sense2vec - A Fast and Accurate Method for Word Sense Disambiguation In Neural Word Embeddings,” *arXiv.org*, Nov. 19, 2015. https://arxiv.org/abs/1511.06388.

[28] S. B. Ozates, A. Ozgur, T. Gungor, and B. O. Basaran, “A hybrid deep dependency parsing approach enhanced with rules and morphology: a case study for Turkish,” *IEEE Access*, vol. 10, pp. 93867–93886, Jan. 2022, doi: 10.1109/access.2022.3202947.