Basic concepts and tools for the Toki Pona minimalist and constructed language: Wordnet synsets; analysis of the vocabulary; synthesis and syntax highlighting of texts

Renato Fabbri
renato.fabbri@gmail.com
University of São Paulo,
Institute of Mathematical and Computer Sciences
São Carlos, SP, Brazil

December 16, 2017

Abstract

A minimalist constructed language (conlang) is useful for experiments and comfortable for making tools. The Toki Pona (TP) conlang is minimalist both in the vocabulary (with only 14 letters and 124 words) and in the $\approx$ 10 syntax rules. The language is useful for being a used and somewhat established minimalist conlang with at least hundreds of fluent speakers. In this article, we describe current concepts and resources for TP, and make available Python scripted routines for the analysis of the language, the synthesis of texts, the specification of syntax highlighting schemes, and the achievement of a preliminary TP Wordnet [1]. We focus on the analysis of the basic vocabulary, as corpus analyses were found in [2]. The synthesis is based on sentence templates, relates to context by keeping track of used words, and renders larger texts by using a fixed number of phonemes (e.g. for poems) and number of sentences, words and letters (e.g. for paragraphs). Syntax highlighting reflects morphosyntactic classes given in the official dictionary and different solutions are described and implemented in the well-established Vim text editor [3]. The tentative TP Wordnet is made available in three forms that reflect the choices of the synsets related to each word. In summary, this text holds potentially novel conceptualizations about, and tools and results in analyzing, synthesizing or syntax highlighting the TP language.

keywords: Constructed languages, Natural Language Processing, Syntax highlighting, Wordnet, Toki Pona

1 Introduction

Toki Pona (TP) is a minimalist conlang (constructed language) with only 124 words (120 without the synonyms). Therefore, the concepts are usually very general and different, and, without context, the words are rarely related through meronymy and hyponymy. Such a linguistic setting is desired because of the simplicity which entails easier e.g. learning and tool making. Another reason why the minimalist language design is compelling is the study and harnessing of the strong and weak forms of the Sapir-Whorf hypothesis (linguistic relativity), i.e. that language influences or dictates thought and world experience [4]. Accordingly, one uses a conlang as a thinking tool (or platform) or to make experiments about the influence language has on the thoughts of the 'speaker' (also writer and reader). Toki Pona is often described as a tool to meditate, to simplify the thinking processes, and as a way to modify the mood and impressions about the world. [5, 6, 7, 8] In [9], Sonja Lang (the creator of Toki Pona), describes that she has seen the language been used successfully in the context of management, creation of texts, legal texts, etc.

In this article, we present a conceptual overview of the language, which is fit both to the newcomer and to the expert for being considerably different from what we found in the literature [5, 6, 10], with emphasis on simplicity and flexibility. We also present novel software routines for analysis, synthesis, syntax highlighting, and the achievement of a tentative TP Wordnet [1].

Next subsections hold a description of the general resources available, a historical note, and some words about natural and constructed languages. Section 2 describes TP's phonology and syntax. Section 3 presents the software routines we made available and immediate results, such as listings and statistics of words, poems and short stories, coloring schemes, and preliminary TP Wordnet versions. Conclusions and further work are in Section 4. Appendix A holds considerations about my usage of Toki Pona, with thoughts about rule breaking and potentially new conlangs. Appendix B holds final words in Toki Pona.

1.1 Resources on Toki Pona

One might organize current resources for the Toki Pona language in: references and learning material, corpus, websites, interaction groups (where users talk a post texts and comments), and software gadgets. The main references of the language are: the official book "Toki Pona: The Language of Good" [5], authored by Sonja Lang, the creator of the language; and the online book "o kama sona e toki pona!", from jan Pije [6]. A number of tools and other resources for dealing with Toki Pona were developed by the community. The most complete list is supposedly [11] and includes videos, musical pieces, artistic texts, reference documents (e.g. a Toki Pona - Esperanto dictionary), journalistic articles, and software tools. For a more comprehensive view of the resources available for the user, we suggest following the links in [11, 12].

1.2 Historical note

TP was developed as an internal and personal language by Sonja Lang [9]. It was released as a draft in 2001 and in 2007 some documents reported it to have a few hundred speakers. The English official book [5] was released only in 2014. In 2016, a version of the official book was released in French. Nowadays, one finds a number of texts about TP and written in it (e.g. in social platforms such as Facebook groups, microblogging, Telegram and IRC), and other diverse uses of TP e.g. for artificial intelligence and software tools [11].

1.3 Natural and constructed and artificial languages

A language one uses (or might use) to communicate by speaking and writing is called 'natural language'. A 'constructed language' (also planned or invented language) is a natural language built by someone or a group, such as Esperanto, Toki Pona, and Lojban. An artificial language is a language yield by artificial agents, such as by AI routines, or by humans in controlled experiments, and are considered e.g. within 'cultural evolution' studies. Formal languages are defined by tokens and rules to operate them, they span from computer programming languages to math and formal models for natural languages.

Constructed languages are in some traditions called artificial languages, but creators most often prefer to use the term 'standardized', 'constructed' or 'planned' language with the argument that the conlang is rooted on natural languages, and 'artificial' is misleading. The preferred terms seem to be planned or constructed language or simply conlang. The construction of languages is called glossopoeia. TP is a conlang which might be classified as engineered for experiments, meditation and philosophy; and suitable for use as an auxiliary international language and as an artistic language. [13, 14]

2 Overview of the language

This section describes very succinctly the formation of words and sentences in the Toki Pona language. It should enable a newcomer to grasp the essentials of the language and the experienced to acquire new insights. Furthermore, it is a solid reference of the phonological and syntax rules, and enables one to understand and modify the software presented in Section 3.

2.1 Phonology

Words in Toki Pona are written using only 14 letters:

- Vowels a (open), e (mid front), o (mid back), i (close front), u (close back).
- Consonants j, k, l, m, n, p, s, t, w:
  - Nasal: m (labial), n (coronal).
  - Plosive: p (labial), t (coronal), k (dorsal).

- Fricative: s (coronal).
- Approximant: w (labial), l (coronal), j (dorsal).

There are standard guidelines for pronunciation, but the language allows for considerable allophonic variation. For example, /p t k s l/ might be pronounced [p t k s l] or [b d g z ɾ]. Especially for poetry, one might consider j and w to be vowels (e.g. j as 'i' and w as 'u').

Syllables are of the form (C)V(n): an optional consonant, a vowel and an optional coronal nasal consonant (n). Non word-initial syllables must follow the pattern CV(N). These sequences are forbidden: ji, wu, wo, ti, nm, nn.

2.2 Syntax

2.2.1 Fundamental notions

As in other natural languages, colloquial TP might have incomplete sentences and deviate from the norm. The basic structure of sentences is: 'subject' (Noun) li 'predicate' (Verb) e 'object' (Noun). The li might be repeated to associate more than one predicate to the subject. The particle li is omitted if the subject is a simple mi (I or us) or sina (you). A discussion about issues (potential problems) yield by this last rule and how I deal with them is in Appendix A. The e might be repeated to associate more than one object to a predicate. Sentences might be related through la, 'sentence' la 'sentence', where the second sentence is the main sentence, and the first sentence is a condition to the second. Multiple la-s are not described in literature, but one might assume that the last sentence is a conditional to the next, except in cases where the context strongly suggests otherwise.

Noun and verb phrases are (usually) built with the non-particle words. The first word is the noun or verb and subsequent words qualify the noun or verb (i.e. are adjectives and adverbs). The pi particle might be used to separate sequences of words to be evaluated before the relation yield by pi. As pi is often ill understood and used, the following structures might be handy for newbies and as a reference:

- No pi, 'word word word': word ← (qualifies 1) word ← (qualifies 2) word.
- One pi, 'word pi word word': word ← (qualifies 2) [word ← (qualifies 1) word].
- Two pi-s: 'word pi word word word pi word word': word ←5 [word 2 word] 3 word ←4 word 1 word; or: word ←5 [word 1 word] 2 word ←4 word 3 word.

Further notes on the usage of pi:

- In a sequence of words, without pi, the second word qualifies the first, the third word qualifies the phrase yield by the first two words, the fourth word qualifies the phrase yield by the first three words and so on.

- It is redundant to use pi before the last word in a noun or verb phrase, reason why it is most often omitted. Its use in this case is considered an error [5, 6], but, as one might notice, it does not add (much) information through syntax because the order of qualifications is conserved. It adds as an emphasis because of greater length of the written segment, as a preparation: 'jan lili pi mama' (mommy's child). Also used in [15].
- The book by jan Pije [6] describes another use for pi: after li to mean possession, e.g. 'soweli li pi sina' (your pet). This employment of pi might be regarded as correct, but is promptly written as a noun phrase (e.g. 'soweli sina') and is not mentioned by the official book [5].

All the words except the particles (li, e, la, pi, a, o, anu, en, seme, mu) and the words classified solely as prepositions (kepeken, lon, tan) are usable after any noun or verb phrase, (i.e. 107 words discarding synonyms). Notice that the phrase expresses a noun (in a subject or object) or a verb (in a predicate). And that the first word of the phrase is the noun or verb, and that subsequent words are adjectives or adverbs. The pi precedes a noun or a verb to be qualified and then qualify the phrase that comes before it.

At this point, the only missing syntax rule is related to the prepositions: kepeken, lon, sama, tan, tawa. They might appear at the end of a phrase, should be followed by another phrase, and require no particle. E.g. 'toki *tan* jan Pije li pana e sona *tawa* mi'. Because prepositions can be used as adjectives or nouns or verbs or adverbs, etc, and might follow any noun or verb phrase, they are one of the three main sources of ambiguity: prepositions; absence of li after sina and mi; incomplete sentences. E.g. '(mi [li]) pana tawa kon e ilo pi suli mute'.

2.2.2 Particles

Beyond the structural particles (li, e, la, pi) presented in last section, other particles are:

- a or kin, emphasis.
- o, vocative or imperative ('jan lukin sitelen o, li wawa')
- taso, means however as sentence or 'only' if adjective.
- anu, en: 'or' and 'and'. Used for nouns in noun phrases. For verbs, repeat li. For object nouns, repeat e. If the noun is complementing a preposition (e.g. tawa, lon), one might repeat the preposition. As TP is a recent language, and is particularly able to cope with variation due to its simplicity, I would advocate for using en and anu/en wherever there is no ambiguity. E.g.: 'mi tawa mute anu moku lon tenpo lili' In the official documentation that is not described and might be regarded as wrong in strict canonical TP.
- nanpa, denotes numbering.

- seme, for questions, used next to the thing being asked for. 'tan seme la sina pana e sike?'.
- mu, for animal noises. For me it is not a particle, as in the official dictionary, but a noun. I also like to use it as a verb: 'mi pakala e luka. mi mu mute.'

The vocabulary specifies these morphosyntactic classes: nouns, adjectives, verbs, pre-verbs, adverbs, prepositions, particles, and numbers. I find that it might help the speaker, specially the newcomer, but it might also suggest a deviation from what I understand and read: the words might be used indistinctly to be a noun (the first word of: a subject, a predicate when there is no object, an object, or a prepositional complement), an adjective (anything that follows an noun and is not a particle or a preposition), a verb (follows mi or sina or li), or an adverbs (follows a verb). The pre-verbs (wile, ken, awen, kama, lukin, sona), might be followed by a verb, but might also be understood as a verb qualified by the next word, which carries a very similar if not identical meaning ('wile moku' might be understood both as a pre-verb followed by a verb and as a verb followed by an adverb). The pre-verb words are all also specified in the official dictionary to have other morphosyntactic classes, such as adjective, noun, and verb. The only exception is wile, which is specified to be only a pre-verb.

Thus, the classes given in the dictionary dictate little in practice: jan kala li lape lon ni. Where kala, lape and ni are in this phrase as adjective, verb and noun, and are in the dictionary as noun, adjective and adjective.

2.2.3 Recognizing POS tags by speaker and machine

One might perform a POS tagging by following these rules:

- Noun: the first word in a noun phrase. Starting a sentence if it is complete, after an 'e', after a pi. The predicate is often a noun if the sentence has no object.
- Adjective: second word on after an e and after second word on in the subject phrase if present.
- Verb: after a li, mi or sina. If there is no object, the predicate might be a noun or a verb.
- After a preposition, one might consider that what follows is 1) always a noun, or that 2) it can be a noun, adjective or verb, or that 3) it is a hybrid of noun, verb and adjective. [16]
- Notice that there is ambiguity in the structure introduced by the omission of li after mi and sina. Also, when there is no object, a noun or a verb or an adjective might be in the verb position. These are sources of syntactic ambiguities in TP. They might be resolved or minimized by using the semantics of the words. An initial effort in this direction might be using the

classes in the official dictionary to resolve ambiguities whenever possible. This solution is not optimal for correct POS tagging, and does not solve all possible ambiguities (there are words classified as noun and adjective, adjective and verb, noun and verb, particle and verb). Another source of ambiguity is the pre-verbs as described in the literature [5, 6]. But I find it reasonable to understand them as verbs, as described in the end of Section 2.2.2.

- If there is no object, the predicate might be parsed as being a hybrid of noun, verb and adjective.

2.2.4 Additional notes

John Clifford (a notable "toki-ponist", a.k.a. jan Kipo) states that there are not noun, verb, adjective (or even prepositional) phrases in TP, but only phrases in a structure yield by particles and content words [16]. The only synonyms on Toki Pona are: a and kin; lukin and oko; sin or namako; ale or ali. In formations such as toki e ni:, wile e ni:, tan ni: etc., '(e) ni' might be omitted and : used alone. Names are by default transliterated, but might not be. These and other issues are further described in Appendix A.

Many other considerations might be made about the language, such as additional (deprecated and new) words, situations where li is used and avoided, canonical and alternative employment of pi, counting systems, associations of TP word sequences to words in English, usage of images to represent TP words and sentences, and best practices in writing TP texts or for translation. For these and other matters, visit [5, 6, 8, 11].

3 Software for analysis, synthesis, and syntax highlighting

In this section, we describe software, statistic, natural language processing, and visualization results: 1) the analysis of the Toki Pona language through statistics of the vocabulary obtained by processing the official dictionary; 2) the synthesis of sentences, poems and paragraphs (e.g. short stories) in Toki Pona; 3) syntax highlighting for Toki Pona, including fine-tuning and theoretical considerations; 4) an initial Wordnet [1] of Toki Pona words. Corpus analysis was already found in [2]. The Python scripts (or the toolbox) for obtaining all the results in this sections, and more, is publicly available in [17], to facilitate both the inspection of the results and the generation of derivatives by other interested parties. Such scripts were made very friendly for one to understand and alter at will.

3.1 Statistics of the vocabulary

In [2] are statistics about a Toki Pona corpus. This section focuses on the statistics of the vocabulary and of the syntactic rules: the letters, phonemes, word

sizes, possible combinations for words and sentences. The script `makeStatistics.py` was used to obtain the all the measurements and tables discussed herein, and hold further measurements and sets of words which were regarded as less suitable for this exposition, but is useful for a deeper investigation.

As described in Section 2, there are only 14 letters, and phonemes are also very restricted. There are 120 different words in the official vocabulary, 4 of them having synonyms. A total of 124 tokens (or lemmas), not counting proper nouns (names) and punctuation. Table 1 shows the number of words related to each POS tag (or morphosyntactic class) specified in the dictionary.

Table 1: POS tags incident and chosen. The official dictionary often relates tokens to more than one POS tag. For the current version of the text highlighting plugin described in Section 3.3, for example, a token has to have an established tag to have a defined color. On the 'Chosen' column, the tokens were regarded only once by choosing the class in the dictionary respecting the precedence order: PRE-VERB, VERB, PREPOSITION, PARTICLE, ADJECTIVE, NOUN, NUMBER.

POS	All	Chosen
NOUN	58	49
ADJECTIVE	40	34
VERB	15	13
PARTICLE	12	12
PRE-VERB	6	6
PREPOSITION	5	5
NUMBER	4	1
total	140	120

From the official 124 words, 26 of them (20.97%) have only one syllable, 85 (68.55%) have two syllables, and 13 (10.48%) have three syllables. No official word has four or more syllables. In all the 124 tokens, there are 235 syllables (68 different), and Table 2 exhibits the 10 most frequent syllables in the first, last, middle and all positions. Middle-word syllables only occur 13 times, and are all different, with the exception of 'la' which occurs twice. A complete list of the syllables and their frequencies (in different positions) is in the `hsyls` variable of the `makeStatistics.py` script. A list of all words, grouped by their size in syllables, and ordered alphabetically and by the number of letters, is in the `hlsyl__` variable.

Vowel and consonant frequencies, and comparisons of vowels against consonants, are as shown in Table 3 for starting, end and internal position. The limited number of consonants favors the naturalness of the language prosody as it resembles babbling.

Given the rules given in Section 2.1 (of 14 letters, 5 vowels, (C)V(n) phonemes, forbidden ji, wu, wo, ti, nn, nm), 96 words are possible with 1 syllable, 8256 with 2 syllables, 710016 with 3 syllables. In the incident words of the official

Table 2: Frequency of syllables in Toki Pona considering all 235 syllables of the 124 tokens, only the first or last syllables or only the middle syllable. In parenthesis are the count and percentage of the corresponding syllable. For more information and a complete list of syllables, see Section 3.1.

rank	all	first	last	middle
1	li (13, 5.53%)	a (8, 6.45%)	li (10, 8.06%)	la (2, 15.38%)
2	la (10, 4.26%)	o (5, 4.03%)	lo (6, 4.84%)	je (1, 7.69%)
3	ka (9, 3.83%)	pi (5, 4.03%)	na (6, 4.84%)	ka (1, 7.69%)
4	na (9, 3.83%)	ka (4, 3.23%)	la (5, 4.03%)	ke (1, 7.69%)
5	pa (9, 3.83%)	la (4, 3.23%)	ma (5, 4.03%)	li (1, 7.69%)
6	a (8, 3.40%)	pa (4, 3.23%)	pa (5, 4.03%)	lu (1, 7.69%)
7	ma (8, 3.40%)	se (4, 3.23%)	ka (4, 3.23%)	ma (1, 7.69%)
8	si (8, 3.40%)	si (4, 3.23%)	sa (4, 3.23%)	me (1, 7.69%)
9	lo (7, 2.98%)	su (4, 3.23%)	si (4, 3.23%)	pe (1, 7.69%)
10	pi (6, 2.55%)	i (3, 2.42%)	te (4, 3.23%)	ta (1, 7.69%)

dictionary, no middle syllable end with the nasal consonant 'n'.

Possible sentence structures are unlimited. For a noun or verb phrase, one might quantify the possibilities by considering all words but the particles that are not classified also as something else (i.e. li, e, la, pi, a, o, anu, en, seme, mu) and the prepositions that are not classified also as something else (i.e. kepeken, lon, tan). This yields $120 - 13 = 107 words$. As discussed in Section 2.2, I advocate that, although such words are classified in the dictionary, they might be used indistinctly as nouns, adjectives, verbs and adverbs. Thus, one has 107 possibilities of noun and verb phrases with one word, $107^2 = 11,449$ possibilities with two words, $107^3 = 1,225,043$ with three words and so on. Be n, v, o, and p the number of words in the subject (noun), predicate (verb), object (noun) and prepositional (see Sections 2.2.3 and 2.2.4) phrases of a sentence, and assume that the sentence has at most one prepositional phrase, one might quantify the possibilities using the formula: $\delta = 107^n \times 107^v \times 10^o \times 5 \times 107^p$, where 5 stands for the possible prepositions. To account for the particles, one possibility is to assume for simplicity that one might use none or one particle (not li, e, la, pi, resulting in 8 particles and 9 possibilities) at each phrase, yielding: $\delta = 107^n \times 107^v \times 107^o \times 5 \times 107^p \times 9^4$. For example, assume $n = v = o = p = 1$, then $\delta = 107^1 \times 107^1 \times 107^1 \times 5 \times 107^1 \times 9^4 = 4.300066 \times 10^+12$, i.e. more then 4 trillion possible sentences with single word phrases (i.e. no adjectives or adverbs), while allowing only one particle per phrase and only one predicate phrase (e.g. 'ona li moku e soweli lon supa'). The general case is implemented as a function in the script `makeStatistics.py` [17].

Table 3: Frequency of letters in Toki Pona. freq, freq_I, freq_L and freq_M are the frequencies of the letters in any, initial, last and middle positions. The columns 'v' and 'c' that follow them are frequencies considering only vowels and consonants. The most frequent vowel is 'a' in any position, although it is more salient among words starting with a vowel and among the last letter of the words. For starting, ending and middle positions, the second most frequent vowel varies. Among the consonants, 'n' is the most frequent because it is the only consonant allowed in the last position and because almost 20% of the words end with 'n'. On the initial position, 's' is the most frequent consonant, while in middle position 'l' is the most frequent consonant. Many other conclusions can be drawn from this table and are useful e.g. for exploring sonorities in poems.

letter	freq	v	c	freq_I	v	c	freq_L	v	c	freq_M	v	c
a	16.35	33.19	-	8.06	40.00	-	29.03	35.64	-	14.22	29.46	-
e	8.60	17.45	-	2.42	12.00	-	11.29	13.86	-	10.78	22.32	-
i	11.53	23.40	-	3.23	16.00	-	20.97	25.74	-	10.78	22.32	-
o	7.55	15.32	-	4.03	20.00	-	14.52	17.82	-	6.03	12.50	-
u	5.24	10.64	-	2.42	12.00	-	5.65	6.93	-	6.47	13.39	-
j	2.10	-	4.13	3.23	-	4.04	0.00	-	0.00	2.59	-	5.00
k	6.29	-	12.40	11.29	-	14.14	0.00	-	0.00	6.90	-	13.33
l	9.22	-	18.18	12.10	-	15.15	0.00	-	0.00	12.50	-	24.17
m	4.61	-	9.09	10.48	-	13.13	0.00	-	0.00	3.88	-	7.50
n	10.48	-	20.66	6.45	-	8.08	18.55	-	100.00	8.19	-	15.83
p	5.66	-	11.16	11.29	-	14.14	0.00	-	0.00	5.60	-	10.83
s	6.29	-	12.40	13.71	-	17.17	0.00	-	0.00	5.60	-	10.83
t	3.14	-	6.20	6.45	-	8.08	0.00	-	0.00	3.02	-	5.83
w	2.94	-	5.79	4.84	-	6.06	0.00	-	0.00	3.45	-	6.67

3.2 Synthesis of text

Such counting exercises are also useful for facilitating (semi-)automated writing through scripting. The syntax organizes the words in larger structures. The rhymes are very restricted and sonorities are bounded by the small vocabulary and simple syntax. There are some specific tasks for achieving texts, such as finding the number of syllables considering the elisions, or handling interaction of the writer with the script to choose sentences or verses or stanzas. The same package [17] has capabilities for synthesizing TP text. On its most basic level, these routines yield noun, verb and prepositional phrases, and sentences. They also aim at making larger scale texts by assuring the use of specific words (to entail context), e.g. to assist the creation of short narratives, and by employing stylistic outlines for poems.

Figure 1 holds some synthesized texts with different color schemes for the syntax highlighting, as presented in the next section. The textual synthesis described here, and implemented in `makeTexts.py`, might be enhanced in countless ways, e.g. as described in Section 4.

Automatic and randomized synthesis of texts in TP is particularly useful because of the reduced vocabulary where each word is related to a broad semantic field. Ideas often make sense in unexpected ways, and thus the synthesis yields a procedure to explore the semantic possibilities within TP. One might object

```
ilo soweli pi ma mi
```

(a) Synthesized phrase.

```
laso li ko moku unpa olin e pipi waso kepeken kili pi kalama
kala.
```

(b) Synthesized sentence.

```
kute akesi pan li olin kala e suwi sama esun loje. kiwen ike li
mije nanpa sinpin e pakala lon alasa suli sina. sinpin mama li
suno lawa e sike soweli pi suwi utala lon moku. poka li kiwen
luka esun open e wawa kiwen kulupu. ma poka ilo li pimeja e
esun sewi noka tan pipi.
```

(c) Synthesized paragraph.

```
# == sitelen nanpa wan tan ilo nanpa ==
# == First poem synthesized ==

kasi jelo ante li anpa tenpo mun
e walo selo tan unpa kama
ona sona awen li lawa lili
e mani laso lon sewi pan mi

mije weka sin li walo lipu
e len kon ken ko kepeken kin mi
luka mama open li kama kin ken
e ko mun moku kepeken sinpin

kule kin lupa li akesi sin ni
e unpa utala uta lon uta akesi
ma pali akesi li wan musi kin
e selo tomo tan sitelen ken

jan suno luka li awen kin ali
e poka pilin lon lili pu uta
tawa sama wan li ante len tu unpa
e toki mije kepeken kon ni

ko jo ilo meli li sin pu tawa
e lawa ale anpa sama namako
moli moli ala li len telo ona
e ken ni wan pan sama pana unpa

anpa nanpa ko li kama taso
e sewi meli tan kulupu ilo
pimeja pini li kiwen ona
e taso esun sin tan nena anpa ko
```

(d) Synthesized poem.

Figure 1: Toki Pona texts synthesized by the `makeTexts.py` script in [17]: (a) a phrase, (b) a sentence, (c) a paragraph, and (d) a poem. The unusual word combinations are convenient for exploring semantic possibilities. One might obtain many of such texts to select the excerpts he/she finds fit for the intended use. For more information, see Section 3.2. The words are colored in accordance to the considerations in Section 3.3.

that the resulting texts are unusual and even consider that they often hold insubstantial or unsound meaning. I advocate that these unexpected formations are desirable for exploring the possibilities of the language and of thought, and for artistic endeavors. Also, to obtain texts which one finds usual or satisfactory in more strict or personal terms, such a person might just write them normally.

3.3 Syntax highlighting

The same package [17] has a Vim [3] syntax highlighting plugin for Toki Pona, routines to arrange the coloring schemes (i.e. to yield the text files with the specifications for which tokens to relate to which coloring groups), and instructions for installing and using the resulting syntax highlighting for Toki Pona texts [17]. An online syntax highlighting gadget is found at [18], and the solution here described presents a number of enhancements: it is capable of coloring all morphosyntactic classes; it might be fine-tuned in the colors and their relations to sets of tokens; it is designed to be used within Vim and might be exported as HTML; the highlighting scheme is promptly rendered by a script; the resulting script might be used with many color schemes.

Basically, the resulting syntax highlighting distinguishes the words among the morphosyntactic classes according to the official dictionary as given in Table 1. A word often belongs to more than one class, thus the precedence of them might be set by the user. Also, some classes might be further refined (e.g. words beginning with vowels) or joined (e.g. distinguishing only particles and the rest, or particles and prepositions and the rest). The colors are also promptly changed according to [3] and exemplified in the package documentation.

Currently, the Python package synthesizes the syntax file through the `makeVimSyntax.py` script. The user has control of class precedence and merging and further details through tweaking such routine. The choice of precise coloring schemes involves fine tuning the color scheme being used in Vim (such as 'blue', 'elflord' and 'gruvbox'), and Vim's highlighting schemes as described in [3]. In summary, the usage of the package and plugin might be performed through the following actions:

- Installation of the plugin.
- Tweaking of the syntax file by hand.
- Running the `makeVimSyntax.py` Python script to generate a new `tokipona.vim` syntax file according to other settings.
- Write a file with the `.tokipona` or `.tp` extension (inside Vim) using Toki Pona words. Reload the highlighting scheme using `:e` whenever you change the syntax file by hand or through the Python script.
- Access the used highlighted groups with `:syntax`. Access all the highlighting groups with `:so $VIMRUNTIME/syntax/hitest.vim` or `:hi`. Change

the coloring of a set of terms by associating a used group (e.g. tokiponaADJECTIVE) to an existing group (e.g. Visual) such as in `:highlight link tokiponaADJECTIVE Visual`.

The syntax file has an association of TP words with highlighting groups. It also holds associations of these groups to other groups to use their color settings. Therefore, one is able to use various color schemes in syntax-highlighted TP texts, such as in Figure 1. A Vim user might run e.g. `:colorscheme blue`, `:colorscheme solarize`, `:colorscheme gruvbox` or `:colorscheme elflord` to see the same text colored with different color schemes (association of colors to sets of tokens). To interfere directly on the colors chosen, `:highlight Normal guifg=#00000 guibg=#0000ff` will change the standard foreground (text) color to pure black and background to pure blue. See `:h gui-colors` and `:h highlight` for the way you might edit colors directly. Useful commands are given as commented lines in the end of the syntax file. One might obtain an HTML file with the colored TP text by using the `:TOhtml` command.

3.3.1 Advanced syntax highlighting considerations

Standard guidelines for syntax highlighting depend heavily on cultural and use factors and have scarce scientific studies [19]. There are informed projects such as Solarized [20], which present solutions for some contexts. Strikingly, standard guidelines for syntax highlighting were not found. Therefore, we considered current data visualization theory [21, 22, 23, 24] to glimpse at the potential guidelines:

- The use of blue and other high frequency colors (such as violet) for the background.
- The avoidance of blue and other high frequency colors to fill small objects, such as letters and words.
- Explore simplicity and elegance through the use of discrete coloring schemes. Most of the tokens should be preferably of the same color. One might use only a small number of colors to achieve a clean visualization of texts. Also, a power-law distribution of tokens among colors might be well-suited to mimic natural occurring phenomena. (Unfortunately, the TP dictionary classification of words are not in a power-law distribution and might be better described by a half-normal or by overlapping normal distributions over the three peaks: $\sim 35-58, 12-18, 1-5 occurrences$).
- Physical stimuli might be related to perceptual stimuli both through a power-law or an exponential law (respectively known as Steven's law and Weber-Fechner's law). This might be useful e.g. for coloring sets of tokens in a way that their similarity is considered.
- One usually wishes to maximize contrast, although taste and less wearing combinations might also dictate the coloring choices.

- Stipulate axes of parameters with which to set colors. Wavelength or frequency is an obvious axis given the considerations above. Other evident parameters are hue, transparency etc.

- One might consider two types of color schemes: those with a dark background, which are more comfortable at first; and those with a light background, which are usually impressive (and even annoying) at first, but the eye gets used to it and it keeps you more stimulated.

- The blue color is specially related to physiological stimulation of the body [25, 26], and programmers often report that a blue background keeps them awake and more concentrated (I also notice such effect, and I have programmed daily for more than 10 years).

- Colors have been associated to enhancements in specific tasks, e.g. blue and red are respectively associated to enhanced creativity and detail-oriented tasks in [25].

- It is important to consider that the user might stay many hours at a text editor (and we very often do), and that the colors and formats involved are prone to entail a considerable effect in the body and mental activity and thus in the quality of life and work of a writer (e.g. a programmer).

- Ideally, one should have facilities for tuning the syntax highlighting (e.g. through keyboard shortcuts) as envisioned in [3].

The considerable irrelevance of the morphosyntactic classes as described in Section 2.2 suggests that an appropriate coloring of words should either distinguish only between particles, prepositions and the rest, or consider the syntax to identify the nouns, verbs, adjectives, adverbs, prepositions and particles. Also, the coloring for poems might be more appealing if considered e.g. the counting of syllables, the repeated letters or syllables, and the ending syllables. Insights might be obtained through the observation of the choices made and advocated in software such as text editors (e.g. Vim, Emacs, Sublime), packages dedicated to syntax highlighting (e.g. pigments[1] and highlight.js[2], and other software (e.g. Linux terminals such as Xterm and gnome-terminal). Finally, the careful choice of fonts is known to have a relevant impact in comfort and productivity [27], e.g. sans-sheriff fonts are more promptly read and yield a cleaner text than a serif font, and the same applies to a monospaced font which is more likely to yield a cleaner text, at least for programming.

One might also consider the mapping the textual structures to sound [28] (i.e. parametrize the synthesis of sounds e.g. by the counting of specific tokens inside a script, a function or class or around a variable of conditional or loop), which might be called "syntax sonification".

[1] http://pygments.org/docs/
[2] https://highlightjs.org/

3.4 Toki Pona Wordnet

For the achievement of a first Toki Pona Wordnet, each of the TP words in the dictionary was related to Wordnet synsets [1] through the English lemmas. The most canonical (i.e. Princeton) Wordnet only contains nouns, adjectives, verbs, and adverbs. Thus, particles and prepositions were not considered. Numbers were considered adjectives. Words presented as adjectives in the dictionary were considered both as adjectives and adverbs. [1]

The `makeWordnet.py` script, available in [17], makes available such tentative TP Wordnets in the simplest form: the TP words are keys in a dictionary that returns the corresponding synsets. Three of such dictionaries are implemented: one where all synsets related to the lemma is returned (the version most consistent with the expositions in Section 2), with 4,052 synsets; another as such but excluding the prepositions, with 3,961 synsets; a last one that returns only the synsets registered in Wordnet with the same POS tag as the lemma is in the official dictionary, with 2,493 synsets. Another immediate possibility, not implemented, would be to relate each Toki Pona word to the synsets simultaneously associated to all the English words bounded by a semicolon in the dictionary. Such collection of synsets, and its relation to the whole Wordnet (the total number of synsets in Wordnet is $117,659$), might guide creation of other conlangs (e.g. one might seek to use lemmas related to synsets that are very far apart, or that has the most complete neighborhood possible).

It is worth mentioning that TP words are very general and each of them might mean many things. Thus, the words are not very easily related e.g. by hyponymy or meronymy. Exceptions: jan (people) is a hyponym of soweli (mammal); kili is an hyponym of kasi; walo, pimeja, jelo, loje, laso are hyponyms of kule. There is at least one caveat: if a particular meaning of each word is chosen, then there might be many other of such relations, e.g. luka (as hand) is a meronym of sijelo (as body).

4 Conclusions and further work

This document presented a potentially novel description of Toki Pona (TP) language in Section 2, and innovative and useful software routines for dealing with the language in Section 3. As a minimalist conlang, TP is very convenient for cognitive experiments through linguistic semantics and for devising tools for analysis, creation and visualization. Other conlangs, and even non-planned languages might benefit from TP and the content presented here in many ways, e.g. one might synthesize TP texts and translate them as convenient, adapt the routines to obtain statistics of the vocabulary, or take advantage of the language description to devise new conlangs or stilistic guidelines for any language.

Possible next steps are numerous:

- Further develop the text synthesis facilities described in Section 3.2 might be enhanced by further employing e.g. rhythm, meter, rhyme and form techniques [29].

- Make a syntax highlighting that colors the tokens with respect to the syntatic position and function, and not in a fixed manner as is implemented and exposed in Section 3.3. This might be implemented by using n-grams and further techniques from Natural Language Processing
- Enhance and study the TP Wordnet described in Section 3.4: implement a NLTK-like Wordnet interface; check if each synset is correctly associated to the TP words; seek further synsets that were not found by the English lemmas in the official dictionary; implement the most restricted version described in Section 3.4, which relates each TP word only to the synsets that are related to all the English words not separated by a semicolon; find out the neighborhood of the synsets of the TP Wordnet in the English Wordnet; explore the possibilities for devising new conlangs by means of criteria related to Wordnet.
- Conlanging: use Wordnet for making new conlangs; use insights derived from TP. Maybe create conlangs for specific uses: describing data, programs, scientific writings, creative writing (exploring the thinking process). A conlang might have different modes that might be specified by a section header or tag.
- Understand how the corpus was gathered in `tokipona.net`.
- Know about previously existing words that were used for TP (e.g. suno and suwi might come from sun and sweet), and about the reasons that led Sonja (and maybe other people) to choose the 14 letters and the syllable structure. This might require a dedicated communication with the speaker community and the documentation keepers.
- Corpus-based analysis.
- Publication of original TP texts and translations.
- Make an article written in TP (extending Section B which might be the first section of a scientific document written in TP). The outline might be: a summary in both English and Toki Pona, for facilitating the acquisition of context, and a article in a canonical structure (introduction and related work, materials and methods, results and discussion, conclusions and further work), with an Appendix explaining the content and context in English. I first conceived something around complexity, statistics, physics, or computer science. But one possibility is to write about linguistics, philosophy, literature or psychology with the partners I write in English. I can start a draft, they might learn the language in a few hours or weeks (with or without my help), to contribute, and we can write a short paper.
- Enhance the synthesis of text to yield better contextualized text and stylistic traces. Give the user the ability to choose the sentences: generate a sentence randomly according to previously written or given text, to some

rules input by the user, to the package and the language guidelines. Should output to the screen and asks to keep and discard.

- Make TP variations where each TP word is replaced by a word in a language with a large speaker population. For example, suno might be sun in an English variant, and sol in a Portuguese variant. A naive version might be obtained through choosing the first word in the description of an official vocabulary, as they already exist in English and French.

- We are particularly interested in software tools, and one possibility is to use a conlang to describe software routines. In this sense, it seems profitable to understand what 'non-ambiguous' means in Lojban, and in what sense one is able to compile and parse Lojban. Such study will probably entail considerations about parsing TP.

Acknowledgments

FAPESP (project 2017/05838-3); Toki Pona, Python and Vim authors, developers and literature maintainers; Toki Pona user and speaker communities. I thank Mario Alzate for introducing me to Toki Pona, and Silverio Guazzelli Donatti for the stimulus that resulted from the usage of Toki Pona for chatting.

A My usage of Toki Pona

I use the standard sounds, but often use [z] for s. I often translate texts to Toki Pona (e.g. biblical excepts) and create new texts as poems and short stories. Most of them are in [30] and/or published in the TP Facebook groups. I omit the li particle after subjects sina and mi, in accordance with the norm, but sometimes I use them when there are many predicates. E.g. sina li wawa li pimeja li lukin pona li moku e kasi mute. In such cases, the first li is sometimes omitted. Also, sometimes I use li after mi and sina where I find that there is unwanted ambiguity, e.g. sina moku pona which might be 'sina li moku pona' or 'sina moku pona (li jo e sike)'.

Names are by default transliterated, but I advocate that, as in other languages, names might be used as they are in the correspondent mother tongue. E.g. the name Erdös is used in English and Portuguese although the standard alphabet does not contain ö in such languages. I also tend to legitimate the use of English (or German) words in TP texts if it is the case, as happens often in scientific writing (kernel is a German word used in English, webpage and software are English words used in Portuguese).

Proposed notations for numbers seem numerous. I tend to think that one might indicate if two numbers are being multiplied (pi) or are in different scales (such as in decimal or binary notations). For example, luka (pi) two might mean 52. Most importantly, 52 is a reasonable notation for a simple language. E.g. 'mi jo e jan sama nanpa 12', or 'ona li lon e soweli 27'.

Avoiding needless words... I've been avoiding e ni: and using only :. I've been omitting the subject (sometimes also the li) if it is the same as in the last sentence. I tend to group these words: noka and anpa; luka, suno, sike and lawa; pali and pana. These clusters of meaning hint me that an even simpler language is possible and maybe more optimized to the core goals of TP.

Also, ambiguities introduced by omission of li and absence of a token to denote preposition, suggest to me the possibility of a syntax that is always uniquely parsed (or at least less ambiguous).

I have been inclined to conceive a language with the same syntax of TP: 'sentence' la 'sentence = subject + predicate + object (each term with a possible preposition complement). but always using particles to bound the sentence sections, repeating them when the function is performed in the smallest scale: 'toki pi toki pona pi jan sona' meaning language in toki pona and of intellectuals vs 'toki pi toki pona pi pi jan sona' meaning language in [toki pona of intelectuals]. About the keyword for initiating a preposition complement might be 'a', but it is already taken in TP. Maybe use 'a' before a preposition and use 'ha' or 'he' instead of TP a. In such a setting, one might enable pi li e a (concept-qualifier, subject-predicate, predicate-object, concept-preposition) in any slot of the template: 'sentence' la 'sentence = subject + predicate + object (each term with a possible preposition complement)' Between sentences one would use: la pi, la li, la e, la a, and la la. This is a very fractal conlang proposal. Maybe also have a way to discern between a noun, a qualifier (adjectives/adverbs) and a verb, and accept any of them for the subject, predicate, object, preposition, slots. Or assume a part of speech as default for a slot or for the first word in a slot.

B Final words in Toki Pona

toki li nasin e lawa. li nasin tawa (pi) toki insa.

toki pona li pona e nasin tan ni: ona li jo e nimi nanpa lili. ona li pona. pona kepeken weka 'p' li ona, a.

o taso la toki pona li kalama li lukin pona. sitelen en nasin li open e sitelen suli [5, 6, 15, 11, 30, 31]. li sona. li nasin e toki e sona e lawa e lon.

toki ni li wawa tawa jan mute nasin en toki. wawa tawa toki pi jan sona. pi ilo nanpa en nanpa nasin. taso tawa toki sona a. sitelen sona, sitelen musi.

ilo lon Poki 3 li pana e sitelen e sona tan sitelen, en nasin. Poki 2 li pana e nasin pi toki pona. e sitelen tawa kama sona.

mi wile pali e sitelen lon toki pona. sitelen lon nasin, sitelen sona, sitelen musi. ante e toki pona la ona li nasa. taso nasa li pona mute. li pona mute tawa lawa, tawa kama sona, tawa sitelen e toki. ante la toki pona o. toki e toki pona tawa sina. kama sona e toki pona sina.

o pona tawa jan pi toki pona. tawa jan Sonja, Birns-Sprage, Kipo, Pije, Siwejo, Malija, Tepan, jan kulupu mute.

References

[1] Fellbaum, C. (1998). WordNet. John Wiley & Sons, Inc..

[2] Henry, J. lipu pi toki pona pi jan Jakopo. Available at http://jimhenry.conlang.org/conlang/tokipona/tokipona.htm

[3] Fabbri, R. An anthropological account of the Vim editor: features and tweaks after 10 years of usage. Available at https://github.com/ttm/vim/raw/master/article/article.pdf

[4] O'Neill, S. P. (2015). Sapir–Whorf Hypothesis. The International Encyclopedia of Language and Social Interaction.

[5] Lang, S. (2014). Toki Pona: the language of good. Tawhid Publishing. ISBN-10: 0978292308, ISBN-13: 978-0978292300.

[6] Knight, B. (2017) o kama sona e toki pona! Available at: http://tokipona.net/tp/janpije/okamasona.php

[7] Blahuš, M. (2011). Toki pona–eine minimalistische Plansprache. Spracherfindung und ihre Ziele. Beiträge der, 20, 51-56.

[8] Schneider, S. (2017). nasin toki pi toki pona. Available at https://github.com/stefichjo/toki-pona

[9] Lang, S. and Broholm, K. AFP 20 – Sonja Lang: Toki Pona, Conlanging and the meaning of life (Sonja Lang interview). Available at https://actualfluency.com/afp-20-sonja-lang-toki-pona-conlanging-meaning-life/

[10] Toki Pona Community. (2017). Various Memrise courses on Toki Pona. https://www.memrise.com/courses/english/?q=toki+pona

[11] Toki Pona Community (2017). toki pona central hub preparation. Available at https://docs.google.com/document/d/1Dzs-imNeZ8TMgdHUiiungJ4Yf97CJk9ylhQPXjWLsJU

[12] Toki Pona. (2017, December 11). In Wikipedia, The Free Encyclopedia. Retrieved 22:48, December 14, 2017, from https://en.wikipedia.org/w/index.php?title=Toki_Pona&oldid=814813420

[13] Constructed language. (2017, November 1). In Wikipedia, The Free Encyclopedia. Retrieved 10:22, November 18, 2017, from https://en.wikipedia.org/w/index.php?title=Constructed_language&oldid=808264974

[14] Artificial language. (2017, November 26). In Wikipedia, The Free Encyclopedia. Retrieved 23:00, December 14, 2017, from https://en.wikipedia.org/w/index.php?title=Artificial_language&oldid=812127746

[15] Fabbri, R. (2017, November 6). o awen, akesi wawa. In the group toki pona taso. Available at https://www.facebook.com/groups/tokiponataso/permalink/1895942817391318/

[16] Clifford, J. et al. (2017, December 15). Considerations about Toki Pona phrases. In the Facebook group 'toki pona'. Available at https://www.facebook.com/groups/sitelen/permalink/1597077093680003/

[17] Fabbri, R. (2017). A Toki Pona Python Package and Vim Syntax Highlighting. Available at https://github.com/ttm/tokipona

[18] Knight, B. (2017). Online gadget for syntax highlighting Toki Pona texts. . Available at http://tokipona.net/tp/DisplayText.aspx

[19] Various authors, 2011-2016. Syntax-highlighting color scheme studies. Software Engineering in Stack Exchange. Available at https://softwareengineering.stackexchange.com/questions/89936/syntax-highlighting-color-scheme-studies

[20] Schoonover, E. Solarized color scheme. Available at http://ethanschoonover.com/solarized

[21] Munzner, T. (2014). Visualization analysis and design. CRC press.

[22] Telea, A. C. (2014). Data visualization: principles and practice. CRC Press.

[23] Ward, M. O., Grinstein, G., & Keim, D. (2010). Interactive data visualization: foundations, techniques, and applications. CRC Press.

[24] Ware, C. (2012). Information visualization: perception for design. Elsevier.

[25] Mehta, R., & Zhu, R. J. (2009). Blue or red? Exploring the effect of color on cognitive task performances. Science, 323(5918), 1226-1229.

[26] Viola, A. U., James, L. M., Schlangen, L. J., & Dijk, D. J. (2008). Blue-enriched white light in the workplace improves self-reported alertness, performance and sleep quality. Scandinavian journal of work, environment & health, 297-306.

[27] Spolsky, A. J. (2008). User interface design for programmers. Apress.

[28] Fabbri, R., Vieira, V. , Pessotti, A. C. S., Corrêa, D. C., Oliveira Jr, O. N. O. Musical elements in the discrete-time representation of sound (2017). arXiv preprint arXiv:1412.6853. Available at https://github.com/ttm/mass/raw/master/doc/article.pdf

[29] Poetry. (2017, November 19). In Wikipedia, The Free Encyclopedia. Retrieved 19:46, December 3, 2017, from https://en.wikipedia.org/w/index.php?title=Poetry&oldid=811036139

[30] Fabbri, R. Toki Sona (a blog dedicated to Toki Pona) (2015-7). Available at http://tokisona.github.io/

[31] Toki Pona community (2017). Wikipesija. Available at: http://tokipona.wikia.com