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

RENATO FABBRI*, Institute of Mathematical and Computer Sciences, University of São Paulo, BR

A minimal constructed language (conlang) is useful for experiments and comfortable for making tools. The Toki Pona (TP) conlang is minimal both in the vocabulary (with only 14 letters and 124 lemmas) and in the (about) 10 syntax rules. The language is useful for being a used and somewhat established minimal conlang with at least hundreds of fluent speakers. This article exposes current concepts and resources for TP, and makes available Python (and Vim) scripted routines for the analysis of the language, synthesis of texts, syntax highlighting schemes, and the achievement of a preliminary TP Wordnet. Focus is on the analysis of the basic vocabulary, as corpus analyses were found. 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. The tentative TP Wordnet is made available in three patterns of relations between synsets and word lemmas. In summary, this text holds potentially novel conceptualizations about, and tools and results in analyzing, synthesizing and syntax highlighting the TP language.

CCS Concepts: • **Applied computing** → Arts and humanities; **Document management and text processing**; • **Social and professional topics** → *Cultural characteristics*;

Additional Key Words and Phrases: Constructed languages, natural language processing, syntax highlighting, wordnet, Toki Pona

1 INTRODUCTION

Toki Pona (TP) is a minimal 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 not frequently related through meronymy and hyponymy. Such a linguistic setting is desired because of the simplicity which entails facilitated e.g. learning and tool making. Another reason why the minimal language design is compelling is the study and harnessing of the strong

---

*This is the corresponding author

Author's address: Renato Fabbri, Institute of Mathematical and Computer Sciences, University of São Paulo, Avenida Trabalhador São-carlense, 400 - Centro, São Carlos, SP, 668, BR, renato.fabbri@gmail.com.

and weak forms of the Sapir-Whorf hypothesis (linguistic relativity), i.e. that language dictates or at least influences thought and world experience [O'Neill 2015]. 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. [Blahuš 2011; Knight 2017; Lang 2014; Schneider 2018] In [Lang and Broholm 2014], Sonja Lang (the creator of TP), describes that she has seen the language been used successfully in the context of management, creation of texts, legal texts, etc.

This article provides a conceptual overview of the language, which is fit both to the newcomer and to the expert for being considerably different from what was found in the literature [community 2018b; Knight 2017; Lang 2014], here with emphasis on simplicity and flexibility. Novel software routines for analysis, synthesis, syntax highlighting, and the achievement of a tentative TP Wordnet [Miller 1998] are also herein presented.

Next subsections hold a description of the general resources available, a historical note, and some words about natural and constructed languages. Section 2 describes the TP 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 the author's 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 categorize 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" [Lang 2014], authored by Sonja Lang, the creator of the language; and the online book "o kama sona e toki pona!", from jan Pije [Knight 2017]. A number of tools and other resources for dealing with Toki Pona were developed by the community. The most complete list is supposedly [community 2018a] 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 [community 2018a; Wikipedia contributors 2018c].

1.2 Historical note

TP was developed as an internal and personal language by Sonja Lang in late 1990s [Lang and Broholm 2014]. A description was released as a draft in 2001, and in 2007 some documents reported it to have a few hundred speakers. The English official book [Lang 2014] was published only in 2014. In 2016, a version of the official book was published in French. Nowadays, one finds a number of texts about TP and written in TP (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 [community 2018a].

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), e.g. Esperanto, Toki Pona, and Lojban, is a natural language built by someone or a group. An 'artificial language' is most usually a term used to designate 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 (aka. conlangs) are in some contexts called artificial languages, but creators most often prefer to use the term 'standardized', 'constructed' or 'planned' language with the argument that the conlangs are rooted on *natural* languages, and 'artificial' is thus misleading. The preferred terms seem to be *planned* or *constructed* language or simply *conlang*. The construction of languages is called glossopoeia. TP is engineered for experiments, meditation and philosophy; and useful at least as an auxiliary international language and as an artistic language. [Wikipedia contributors 2017, 2018a]

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 the author deals with them is in Appendix A. The particle 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 sentences are conditions to the next, except in cases where the context 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. yield 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 [Knight 2017; Lang 2014], 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 segment, as a preparation: 'jan lili pi mama' (mommy's child). Also used in [Fabbri 2017b].
- The book by jan Pije [Knight 2017] describes another use for pi: after li to mean possession, e.g. 'soweli li pi sina' (your pet). This employment of pi might thus be regarded as correct, but is promptly written as a noun phrase (e.g. 'soweli sina') and is not mentioned by the official book [Lang 2014].

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 (used as) adjectives or adverbs. The pi precedes a noun or a verb to be qualified and then qualify the phrase that comes before it. These are reasonable expectations and not strict conventions.

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, might also 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 four main sources of ambiguity: small vocabulary; 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 when subject or after pi. 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, one might 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, might otherwise 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. It is not only a particle, as in the official dictionary, but also a noun and 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. Such classification helps the speaker, specially the

newcomer, but it also suggests a deviation from TP usage: the words might be used indistinctly as 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 have other morphosyntactic classes, in the official dictionary, such as adjective, noun, and verb. The only exception is wile, which is specified to be only a pre-verb, but is trivially used as a verb: 'mi wile e moku e telo'.

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 'li' as the verb 'to be'. The particle li very often relates phrases as the ver 'to be'. E.g. 'ona li pona' (that is beautiful) and 'jan nasin li jan wawa' (someone with techniques is someone strong).

2.2.4 Recognizing POS tags by speaker and machine. One might perform a reasonable 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. [et al. 2017]
- If there is no object, the predicate might be parsed as being a hybrid of noun, verb and adjective.
- 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 considering 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 [Knight 2017; Lang 2014] and in the end of Section 2.2.2.

2.2.5 Additional notes: synonyms, e ni, names, structure, other. 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. John Clifford (a notable "toki-ponist", aka. 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 [et al. 2017].

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 [community 2018a; Knight 2017; Lang 2014; Schneider 2018].

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 (TP) 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 TP; 3) syntax highlighting for TP, including fine-tuning and theoretical considerations; 4) an initial Wordnet [Miller 1998] of TP words. Corpus analysis was already found in [III 2018]. The Python scripts (i.e. the toolbox) for obtaining all the results in this section, and more, is publicly available in [Fabbri 2018b], to facilitate both the inspection of the results and the generation of derivatives by other interested parties. Such scripts are friendly to understand and alter.

3.1 Statistics of the vocabulary

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 `TPTabFig` Python class was used to obtain all the measurements and tables discussed herein, and hold further measurements and sets of words regarded as less suitable for this exposition than for a deeper investigation.[Fabbri 2018a]

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 as preferential e.g. in text synthesis. The official dictionary often relates tokens to more than one POS tag. For the text highlighting Plugin, 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 first occurrence of ['PRE', 'VERB', 'PREPOSITION', 'PARTICLE', 'ADJECTIVE', 'NOUN', 'NUMBER'] in the official dictionary.

POS	All	Chosen
NOUN	58	49
ADJECTIVE	40	34
VERB	15	13
PARTICLE	12	12
PRE	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 `tf.hsyls` variable after executing the `basic/tbBasic.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 `tf.hlsyl__` variable.

Vowel and consonant frequencies, and comparisons of vowels against consonants, are as shown in Table 3 for start, end and internal positions. The limited number of consonants favors the language prosody to appear natural as it resembles babbling.

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 syllables. 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%)

Table 3. Frequency of letters in Toki Pona. Columns 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 may 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

Within the rules exposed in Section 2.1 (of 14 letters, 5 vowels, (C)V(n) phonemes, forbidden ji, wu, wo, ti, nn, nm syllables), 96 words are possible with 1 syllable, 8256 with 2 syllables, 710016 with 3 syllables. In the incident words of the official 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, the author advocates that, although words have specific classes 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 2.2.4 and 2.2.5) 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 the use of at most 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'). [Fabbri 2018b].

3.2   Synthesis of text

Such counting exercises are also useful for (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 package [Fabbri 2018b], herein presented, also 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 the TPSynth class, 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 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 or use the synthesized texts as raw material.

3.3   Syntax highlighting

The same package [Fabbri 2018b] has a Vim [Fabbri 2017a] syntax highlighting plugin for Toki Pona. The Python scripts hold routines to arrange the coloring schemes (i.e. to specify which tokens are related to which coloring groups) which are stored in the syntax/tokipona.vim file. An online syntax highlighting gadget is found at [Knight 2018], 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 in combination with virtually any color scheme.

Basically, the resulting syntax highlighting distinguishes the words among the morphosyntactic classes according to the official dictionary as given in Table 1. Also, some classes might be further defined (e.g. words beginning with vowels) or joined (e.g. distinguishing only particles against rest, or

(a) Synthesized phrase.

(b) Synthesized sentence.

(c) Synthesized paragraph.

(d) Synthesized poem.

Fig. 1. Toki Pona texts synthesized by an instantiated TPSynth Python class [Fabbri 2018b]: (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 texts to select the excerpts she finds fit for the intended use. For more information, see Section 3.2. The words are colored in accordance with the considerations in Section 3.3.

particles and prepositions against the rest). The colors are also promptly changed according to [Fabbri 2017a] and exemplified in the package documentation, specially the `syntax/tokipona.vim` syntax file

Currently, the Python package synthesizes the syntax file through the TPSynHigh class. The user has control of class precedence, 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 [Fabbri 2017a]. The usage of the package and plugin might be performed, in summary, through the following actions:

- Installation of the plugin, so that syntax highlighting of TP texts will be performed in any `*.tp` or `*.tokipona` file.
- Tweaking of the syntax file by hand. Or
- Instantiating the TPSynHigh class and then executing the `mkSynHighFile()` method of the resulting object. This generates a new `tokipona.vim` syntax file, at the right location in Vim folders, according to object 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 routines.
- 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. tpADJECTIVE) to an existing group (e.g. Visual) such as in `:highlight link tpADJECTIVE Visual`. This may also be achieved changing `TPSynHigh().colors` attribute before executing `mkSynHighFile()`.

Such procedures are facilitated by the `basic/tpBasic.py`. One may change: associations of tokens to HIGs; associations of regex patterns to HIGs; associations of HIGs to other HIGs; whole color schemes or just sections of it. The last of these tasks is promptly performed using the `space-c-s` and `space-c-c` normal Vim commands supplied by the color plugin of the PRV framework [Fabbri 2017a]. The other tasks are more easily performed by scripting Vim, Python import and package source code, depending on the user ability and intended tasks. The use of different color schemes within the same syntax file is exemplified in Figure 1. By copying only the `syntax/tokipona.vim` and `ftdetect/tokipona.vim` files, a user might benefit from the standard Vim commands. 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. One might obtain an HTML file with the colored TP text 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 [authors 2016]. There are informed projects such as Solarized [Schoonover 2011], which present solutions for some contexts. Strikingly, standard guidelines for syntax highlighting were not found. Therefore, we considered current data visualization theory [Munzner 2014; Telea 2014; Ward et al. 2015; Ware 2012] to glimpse at the potential guidelines:

- The use of blue and other high frequency colors (such as violet) for the background and the avoidance of their use to fill small objects, such as letters and words.
- Explore simplicity and elegance in the coloring schemes. Most tokens should be preferably of the same color, i.e. a small number of colors might be explored to achieve a clean visualization of texts.

- A power-law distribution of tokens among colors might be well-suited to mimic natural occurring phenomena. The TP dictionary morphosyntactic classification of words resemble a long-tailed distribution: as shown in Table 1, few classes hold the vast majority of words.
- 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 considering their similarity, or correctly choosing values for e.g. the hue or luminosity channels.
- One usually wishes to maximize contrast, although taste and less wearing combinations might also dictate the coloring choices.
- Stipulate axes of parameters to set colors. Wavelength or frequency is an obvious axis given the considerations previously exposed, but they are not perceptually uniform. Considering perceptually uniform color spaces in making choices may be relevant, such as CIElab and CIEluv.
- Consider two types of color schemes: of those with a dark background, which are more comfortable at first; and of those with a light background, which are usually impressive (and even annoying) at first, but the eye adapts in a few minutes, keeps you more stimulated, and is suitable for bright contexts (e.g. in daylight) .
- The blue color is specially related to physiological stimulation of the body [Mehta and Zhu 2009; Viola et al. 2008], and programmers often report that a blue background keeps them awake and more concentrated (the author also notices such effect).
- 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 [Mehta and Zhu 2009].
- It is important to consider that *a text editor user* might stay many hours using the tool (and they very often do), and that the colors and formats involved are thus prone to entail a considerable effect in the body and mental activity, the quality of life, and work results of a writer (e.g. a programmer).
- Ideally, one should have facilities for tuning the syntax highlighting (e.g. through keyboard shortcuts) as envisioned and implemented in [Fabbri 2017a].

The considerable irrelevance of the morphosyntactic classes, 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 of poems might be more appealing if considered e.g. the counting of syllables, the repeated letters or syllables, and the ending syllables. Further insights might be obtained through 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 [Spolsky 2008], 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 [Fabbri et al. 2014] (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

A first Toki Pona Wordnet was constructed relating each of the TP words in the dictionary to English Wordnet synsets [Miller 1998] through the English lemmas. The canonical (i.e. Princeton) Wordnet only contains nouns, adjectives, verbs, and adverbs. Thus, particles were not considered. Numbers were considered adjectives. Words presented as adjectives in the dictionary were considered both as adjectives and adverbs. Prepositions were considered in all classes. [Miller 1998]

The `TPWordnet` class [Fabbri 2018b], provides such tentative TP Wordnets in their 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 4,027 synsets; another as such but excluding the prepositions, with 3,929 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 2,462 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).

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. lawa, luka, noka, sinpin, monsi, lukin, mute, kute, palisa, and lupa might be considered meronyms of sijelo. Antonyms are also specially useful in TP: suno and mun, pona and jaki or ike, sinpin and munsi, lawa and noka, mije and meli, sike and palisa, pana and kama jo, pimeja and walo, weka and poka, sama and ante, ali and ala, anu and e, selo and insa.

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 minimal conlang, TP is very convenient for cognitive experiments through linguistic semantics and for devising tools for analysis, creation and visualization, a fact to which this document is evidence. Other conlangs, and even non-planned languages might benefit from TP and this content 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 stylistic guidelines for any language.

At the moment, the framework in [Fabbri 2018a] is being used, by the author, for TP chatter bots and for semantic explorations of domain knowledge. Some of the many potential next steps should be highlighted:

- The text synthesis facilities described in Section 3.2 might be enhanced by further employing e.g. rhythm, meter, rhyme and form techniques [Wikipedia contributors 2018b].
- Syntax highlighting that colors the tokens with respect to the syntactic positions and functions, and not in a fixed manner as is implemented and exposed in Section 3.3, may be implemented by using n-grams and other techniques from Natural Language Processing.
- The TP Wordnet(s) described in Section 3.4 should be may be enhanced: implement a NLTK-like Wordnet interface; double-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; scrutinize the neighborhood of the synsets of the TP Wordnet in the English Wordnet.

- Conlanging: use Wordnet for making new conlangs; use insights derived from TP Wordnet. Create conlangs for specific uses: describing data, programs, scientific writings, creative writing (exploring the thinking process). A conlang might have different modes that are specified by section headers or tags.
- Explore TP relation to other languages (e.g. suno and suwi might come from sun and sweet), and 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.
- Understand how the corpus was gathered in tokipona.net.
- Corpus-based analysis.
- Publication of original TP texts and translations.
- Write academic articles 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 facilitated acquisition of context, and an article in a canonical structure (e.g. introduction and related work, materials and methods, results and discussion, conclusions and further work), with an Appendix explaining the content and context in English. Maybe something about complexity, statistics, physics, or computer science; or linguistics, philosophy, literature or psychology. Partners may learn the language in a few hours or weeks (with or without help). This is a very challenging step.
- 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.
- Explore a minimal conlang to describe software routines. In this sense, it seems advantageous to understand what 'non-ambiguous' means in Lojban, and in what sense one is able to compile and parse Lojban. Consider the possibilities of and reasons for parsing TP.

A CURRENT USAGE OF TOKI PONA BY THE AUTHOR

Disclaimer: this whole section holds personal views and usages of the Toki Pona (TP) language. It is reasonable to use the standard sounds, but often [z] instead of s is more comfortable. Translation into Toki Pona (e.g. of biblical excepts), and the creation of new texts as poems and short stories, is both challenging and pleasant. Texts are shared e.g. in [Fabbri 2018c] and/or published in the TP Facebook groups. The li particle is most often not used after subjects sina and mi, in accordance with the norm, but, even so, they are useful 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, li is found after mi and sina where one finds 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 one should notice 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. The use of e.g. English (or German) words in TP texts is found when needed, 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. E.g. one might indicate if two numbers are being multiplied (pi) or are in different scales (such as in decimal or binary notations): luka (pi) two

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... 'e ni:' may most often be written as ':'. Subject phrase may be omitted (sometimes also the li) if it is the same as in the last sentence. One may vary TP by grouping e.g. these words: noka and anpa; luka, suno, sike and lawa; pali and pana.

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

It is tempting to conceive a language with the same syntax of TP: 'sentence' la 'sentence = subject + predicate + object' (each term with a possible prepositional 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]. The keyword for initiating a preposition complement might be 'a', but it is already taken in TP. One possibility is to 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 syntax template. Between sentences one may 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 [community 2018a,c; Fabbri 2017b, 2018c; Knight 2017; Lang 2014].
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.

ACKNOWLEDGMENTS

REFERENCES

Various authors. 2016. Syntax-highlighting color scheme studies. *Software Engineering at Stack Exchange*. Retrieved July 01, 2018 from https://softwareengineering.stackexchange.com/questions/89936/syntax-highlighting-color-scheme-studies

Marek Blahuš. 2011. Toki pona–eine minimalistische Plansprache. *Spracherfindung und ihre Ziele. Beiträge der* 20 (2011), 51–56.

Toki Pona community. 2018a. toki pona central hub preparation. Retrieved July 01, 2018 from https://docs.google.com/document/d/1Dzs-imNeZ8TMgdHUiiungJ4Yf97CJk9ylhQPXjWLsJU

Toki Pona community. 2018b. Various Memrise courses on Toki Pona. Retrieved July 01, 2018 from https://www.memrise.com/courses/english/?q=toki+pona

Toki Pona community. 2018c. Wikipesija. Retrieved July 01, 2018 from http://tokipona.wikia.com

John Clifford et al. 2017. o awen, akesi wawa. Retrieved July 01, 2018 from https://www.facebook.com/groups/sitelen/permalink/1597077093680003/

Renato Fabbri. 2017a. An anthropological account of the Vim text editor: features and tweaks after 10 years of usage. *arXiv preprint arXiv:1712.06933* (2017).

Renato Fabbri. 2017b. o awen, akesi wawa. Retrieved July 01, 2018 from https://www.facebook.com/groups/tokiponataso/permalink/1895942817391318/

Renato Fabbri. 2018a. PRV: Python, RDF and Vim. Retrieved July 03, 2018 from https://github.com/ttm/prv/

Renato Fabbri. 2018b. Toki Pona tools and article. Retrieved July 01, 2018 from https://github.com/ttm/tokipona

Renato Fabbri. 2018c. Toki Sona: a blog dedicated to Toki Pona. Retrieved July 01, 2018 from http://tokisona.github.io/

Renato Fabbri, Vilson Vieira da Silva Junior, Antônio Carlos Silvano Pessotti, Débora Cristina Corrêa, and Osvaldo N Oliveira Jr. 2014. Musical elements in the discrete-time representation of sound. *arXiv preprint arXiv:1412.6853* (2014).

Jim Henry III. 2018. lipu pi toki pona pi jan Jakopo. Retrieved July 01, 2018 from http://jimhenry.conlang.org/conlang/tokipona/tokipona.htm

Bryant Knight. 2017. o kama sona e toki pona! Retrieved July 01, 2018 from http://tokipona.net/tp/janpije/okamasona.php

Bryant Knight. 2018. Online gadget for syntax highlighting Toki Pona text. Retrieved July 01, 2018 from http://tokipona.net/tp/DisplayText.aspx

Sonja Lang. 2014. *Toki Pona: The language of good*.

Sonja Lang and Kris Broholm. 2014. AFP 20 – Sonja Lang: Toki Pona, Conlanging and the meaning of life. Retrieved July 01, 2018 from https://actualfluency.com/afp-20-sonja-lang-toki-pona-conlanging-meaning-life/

Ravi Mehta and Rui Juliet Zhu. 2009. Blue or red? Exploring the effect of color on cognitive task performances. *Science* 323, 5918 (2009), 1226–1229.

George Miller. 1998. *WordNet: An electronic lexical database*. MIT press.

Tamara Munzner. 2014. *Visualization analysis and design*. AK Peters/CRC Press.

Sean P O'Neill. 2015. Sapir–Whorf Hypothesis. *The International Encyclopedia of Language and Social Interaction* (2015), 1–10.

Stephan Schneider. 2018. nasin pi toki pona. Retrieved July 01, 2018 from https://github.com/stefichjo/toki-pona

Ethan Schoonover. 2011. Solarized: precision colors for machines and people. Retrieved July 01, 2018 from http://ethanschoonover.com/solarized

Avram Joel Spolsky. 2008. *User interface design for programmers*. Apress.

Alexandru C Telea. 2014. *Data visualization: principles and practice* (2nd. ed.). AK Peters/CRC Press.

Antoine U Viola, Lynette M James, Luc JM Schlangen, and Derk-Jan Dijk. 2008. Blue-enriched white light in the workplace improves self-reported alertness, performance and sleep quality. *Scandinavian journal of work, environment & health* (2008), 297–306.

Matthew O Ward, Georges Grinstein, and Daniel Keim. 2015. *Interactive data visualization: foundations, techniques, and applications.* AK Peters/CRC Press.

Colin Ware. 2012. *Information visualization: perception for design.* Elsevier.

Wikipedia contributors. 2017. Artistic language — Wikipedia, The Free Encyclopedia. https://en.wikipedia.org/w/index. php?title=Artistic_language&oldid=808260717 [Online; accessed 3-July-2018].

Wikipedia contributors. 2018a. Constructed language — Wikipedia, The Free Encyclopedia. https://en.wikipedia.org/w/index.php?title=Constructed_language&oldid=846512368 [Online; accessed 1-July-2018].

Wikipedia contributors. 2018b. Poetry — Wikipedia, The Free Encyclopedia. https://en.wikipedia.org/w/index.php?title=Poetry&oldid=847801129 [Online; accessed 1-July-2018].

Wikipedia contributors. 2018c. Toki Pona — Wikipedia, The Free Encyclopedia. https://en.wikipedia.org/w/index.php?title=Toki_Pona&oldid=846623837 [Online; accessed 1-July-2018].