README

This document provides an overview of the `fsca` package. This package was developed to calculate measures of syntactic complexity in L2 French texts as part of a research project focusing on phraseological complexity in learner language. For more details see Vandeweerd, Housen and Paquot (in press).

Also see how to run a complete analysis in this vignette.

Licence

This package is distributed with an MIT license, which means that anyone is free to use or modify the contents. In such cases, please the following citation:

> Vandeweerd, N. (in press). *fsca: French syntactic complexity analyzer.* International Journal of Learner Corpus Research.

Installation

The `fsca` package can be installed from the github repository:

```
install.packages("devtools")
devtools::install_github("nvandeweerd/fsca")
library(fsca)
```

Overview

The package contains three functions:

- `getParse()`: a function to prepare dependency parsed texts for analysis
- `getUnits()`: a function to extract syntactic units from dependency parsed texts
- `getMeasures()`: a function to calculate measures of syntactic complexity from dependency parsed texts

To see the documentation for each function, call `help()`.

The package also contains a data file (`test.sents`) with the example sentences found below. This can be loaded using `data()`.

```
data(test.sents)
```

Preprocessing

The input of both `getUnits()` and `getMeasures()` is a dependency parsed sentence in CONLL format. This function was written to extract syntactic units from a corpus of L2 French texts in order to calculate measures of syntactic complexity (see Vandeweerd, Housen, & Paquot, n.d.). The texts were POS-tagged with MElt Tagger (Denis & Sagot, 2012) and dependency parsed with Malt Parser (Nivre, Hall, & Nilsson, 2006).[1]

[1] On the basis of the dependency tags generated by Malt Parser, five new dependency labels were created for the purpose of calculating phraseological complexity measures: dobj: objects of verbs with the POS tag "NC"; amod : noun modifiers with the POS tag "ADJ"; advmod_ADJ: modifiers of adjectives with the POS tag "ADV"; advmod_ADV: modifiers of adverbs with the POS tag "ADV"; advmod_VER: modifiers of verbs with the POS tag "ADV"

An example of a sentence in CONLL format is provided below:

```
[1] "C'est un point très important."
```

```
##       TOKEN   POS.TT     LEMMA POS.MELT POSITION   DEP_TYPE DEP_ON
## 1        C'  PRO:DEM        ce      CLS        1        suj      2
## 2       est VER:pres      être        V        2       root      0
## 3        un  DET:ART        un      DET        3        det      4
## 4     point      NOM     point       NC        4        ats      2
## 5      très      ADV      très      ADV        5 advmod_ADJ      6
## 6 important      ADJ important      ADJ        6       amod      4
## 7         .     SENT         .    PONCT        7      ponct      2
```

In this case, the main verb of the sentence is *est* ('is-3SG.PRES') and is labeled as the 'root'. As the top level node of a sentence, it is not dependent on any other word, hence the value of 0 in the DEP_ON column. Because *est* is the second word of the sentence, the position is 2 (indicated in the POSITION column). All words which are directly dependent on *est* have the value of 2 in the DEP_ON column. This includes the subject *C'* ('it') as well as the subject attribute *point* ('point'). The sentence-final period is also dependent on the root. The words which are directly dependent on *point* include the determiner *un* ('a') and the adjective *important* ('important'), which is itself modified by the adverb *très* ('very'). In this way, every word in the sentence is dependent on one and only one word.[2]

Syntactic structures

The `getUnits()` function first searches for a list of node words for a given unit (e.g. nouns for noun phrases) and then extracts all of the dependencies on each of the node words using the using the `induced.subgraph()` function from the `igraph` package (Csardi & Nepusz, 2006). The definitions used for each unit are provided below along with examples extracted from the corpus. Each example provides the original sentence followed by the units which were extracted. They have been simplified for readability by pasting the tokens together.

Sentences

Following Lu (2010: 481) we defined a sentence as "a group of words delimited by one of the following punctuation marks that signal the end of a sentence: period, question mark, exclamation mark, quotation mark or ellipsis". Because the CONLL input to the `getUnits()` function is already segmented into sentences, no additional query is required.

Clauses

Clauses are defined as structures with a subject and a finite verb (Hunt, 1965). After identifying dependent clauses and T-units, clauses include all T-units as well as all subordinate clauses emedded within the T-units in a given sentence.

Dependent clauses

Dependent clauses are clauses which are semantically and/or structurally dependent on a super-ordinate syntactic structure. They include nominal clauses, adverbial clauses and adjectival clauses (Hunt, 1965; Lu, 2010). They must contain a finite verb and a subject.

Nominal clauses

Nominal clauses are extracted using subordinate conjunctions (e.g. *que*, 'which') as the main node.

```
[1] "Ils prétendent qu'il est impossible de rééduquer un tel jeune criminel."
```

[2] For an overview of dependency parsing see Green (2011).

```
getUnits(test.sents[["b.208.1"]],
         what = "tokens",
         units = c("DEP_CLAUSES"),
         paste.tokens = TRUE)
## $DEP_CLAUSES
## $DEP_CLAUSES[[1]]
## [1] "qu' il est impossible de rééduquer un tel jeune criminel"
```

Adverbial clauses

Adverbial clauses are also extracted using subordinate conjunctions as the main node.

```
[1] "Quand les lois sont contre le droit, il n'y a qu'une héroïque façon de protester contre elles : les
```

```
getUnits(test.sents[["b.347.1"]],
         what = "tokens",
         units = c("DEP_CLAUSES"),
         paste.tokens = TRUE)
## $DEP_CLAUSES
## $DEP_CLAUSES[[1]]
## [1] "Quand les lois sont contre le droit"
```

Adjectival clauses

Adjectival clauses are extracted using pronouns and finite verbs that modify a noun as the main nodes (e.g. *durent* and *arrivent* in the example below).

```
[1] "Des procès qui durent des années, des déclarations d'impôts inhumainement longues et compliquées, u
```

```
getUnits(test.sents[["b.110.1"]],
         what = "tokens",
         units = c("DEP_CLAUSES"),
         paste.tokens = TRUE)
## $DEP_CLAUSES
## $DEP_CLAUSES[[1]]
## [1] "qui durent des années"
##
## $DEP_CLAUSES[[2]]
## [1] "qui n' arrive pas à prendre forme"
```

Special cases

Clauses of the type 'il y a' are considered dependent clauses only if *a* is directly dependent on a finite verb or if there is a finite verb dependent on *a*. This means that adverbial clauses which function like 'ago' in English (e.g. *il y a deux ans...*; 'two years ago...') are captured as dependent clauses but simple declaratives (e.g. *il y a une maison*; 'there is a house') are not.

```
[1] "Il y a quelques siècles, les empereurs pourraient modifier les règles selon leur volonté."
```

```
getUnits(test.sents[["a.65.1"]],
         what = "tokens",
         units = c("DEP_CLAUSES"),
         paste.tokens = TRUE)
## $DEP_CLAUSES
## $DEP_CLAUSES[[1]]
## [1] "Il y a quelques siècles"
```

```
[1] "Ensuite, il y a des délits dus aux problèmes personnels survenus à la maison ou à l'école."
```

```
getUnits(test.sents[["a.57.1"]],
         what = "tokens",
         units = c("DEP_CLAUSES"),
         paste.tokens = TRUE)
## $DEP_CLAUSES
## $DEP_CLAUSES[[1]]
## [1] "NA"
```

Direct interrogatives in the form *est-ce que* are not considered the head of subordinate clauses. Rather, the head of a clause is the finite verb dominated by *est* as in interrogatives formed by inversion.

```
[1] "Est-ce que les règles sont nécessaires?"
```

```
getUnits(test.sents[["a.17.1"]],
         what = "tokens",
         units = c("CLAUSES", "DEP_CLAUSES"),
         paste.tokens = TRUE)
## $CLAUSES
## $CLAUSES[[1]]
## [1] "Est -ce que les règles sont nécessaires"
##
##
## $DEP_CLAUSES
## $DEP_CLAUSES[[1]]
## [1] "NA"
```

Citations or reported speech enclosed with French guillmets («»), single or double quotation marks are also considered subordinate clauses.

```
[1] "Il est possible que la langue disparaisse après quelques générations, car, comme dit le professeur
```

```
getUnits(test.sents[["b.274.1"]],
         what = "tokens",
         units = c("DEP_CLAUSES"),
         paste.tokens = TRUE)
## $DEP_CLAUSES
## $DEP_CLAUSES[[1]]
## [1] "que la langue disparaisse après quelques générations"
##
## $DEP_CLAUSES[[2]]
## [1] "comme dit le professeur Roegiest professeur des langues romanes et linguiste de l' espagnol à l
##
## $DEP_CLAUSES[[3]]
## [1] "Une langue qui a perdu son prestige est souvent condamnée à mort"
##
## $DEP_CLAUSES[[4]]
## [1] "qui a perdu son prestige"
```

Coordinated clauses

Coordinated clauses are clauses which are not semantically and/or structurally dependent on a super-ordinate syntactic structure but are conjoined to one or more clauses of syntactically equal status. They may be joined by a coordinating conjunction (e.g. *et*; 'and'), punctuation (e.g. semi-colon, colon, comma) or by juxtaposition and must contain both a subject and a finite verb. They are extracted from the root nodes of finite verbs which are not themselves dependent on subordinate conjunctions or pronouns (to exclude dependent clauses).

```
[1] "Dans la prison, il n'ont plus d'éducation, il ne voient que des criminels et ils doivent devenir d
```

```
getUnits(test.sents[["a.90.1"]],
         what = "tokens",
         units = c("CO_CLAUSES"),
         paste.tokens = TRUE)
## $CO_CLAUSES
## $CO_CLAUSES[[1]]
## [1] "Dans la prison il n' ont plus d' éducation"
##
## $CO_CLAUSES[[2]]
## [1] "il ne voient que des criminels"
##
## $CO_CLAUSES[[3]]
## [1] "et ils doivent devenir des adultes en nulle temps"
```

T-units

We use Hunt's (1970: 199) definition of a t-unit as "one main clause plus any subordinate clause or non-clausal structure that is attached to or embedded in it". Identifying t-units therefore depends on the identification of coordinated clauses since a sentence can only contain multiple t-units if it contains multiple coordinated clauses. A sentence that does not contain any coordinated clauses simply has one t-unit, provided it has at least one finite verb. Consistent with Hunt (1965), we do not classify sentence fragments (clauses without a finite verb) as t-units. Therefore, if a sentence has no coordinated clauses (and one finite verb) it has one t-unit. All additional coordinated clauses within a sentence are separate t-units. The three coordinated clauses in the sentence above therefore account for three t-units.

```
[1] "Dans la prison, il n'ont plus d'éducation, il ne voient que des criminels et ils doivent devenir d
```

```
getUnits(test.sents[["a.90.1"]],
         what = "number",
         units = c("CO_CLAUSES","T_UNITS"))
## CO_CLAUSES    T_UNITS
##          3          3
```

Noun phrases

We use Lu's (2010) definition of a noun phrase as a *complex nominal* (see Cooper, 1976) which includes: nouns plus adjective(s), possessive(s), prepositional phrase(s), relative clause(s), participle(s), or appositive(s), nominal clause(s). We also include words (nouns, adverbs and pronouns) that are have a determiner (e.g. *une maison* 'a house'; *cet autre* 'this other') in our definition. Following Lu (2010) and Cooper (1976), we also include gerunds and infinitives in subject position. The root nodes of noun phrases therefore include (within each t-unit) common nouns and proper nouns as well as gerunds and non-finite verbs when they are dependent on a finite verb and have a subject dependency label.

```
[1] "La question sur une nouvelle réforme de l'État est un grand problème qui se pose aujourd'hui en Be
```

```
getUnits(test.sents[["b.46.1"]],
         what = "tokens",
         units = c("NOUN_PHRASES"),
         paste.tokens = TRUE)
## $NOUN_PHRASES
## $NOUN_PHRASES[[1]]
## [1] "La question sur une nouvelle réforme de l' État"
##
## $NOUN_PHRASES[[2]]
```

```
## [1] "une nouvelle réforme de l' État"
##
## $NOUN_PHRASES[[3]]
## [1] "l' État"
##
## $NOUN_PHRASES[[4]]
## [1] "un grand problème qui se pose aujourd' hui en Belgique"
##
## $NOUN_PHRASES[[5]]
## [1] "La question"
##
## $NOUN_PHRASES[[6]]
## [1] "une nouvelle réforme"
##
## $NOUN_PHRASES[[7]]
## [1] "un grand problème"
```

Verb phrases

As in Lu (2010) we count both finite and non-finite verb phrases. These are extracted by taking all words which are dependent on a verb within a t-unit. Auxiliary verbs do not constitute their own verb phrase but are considered part of the main verb they modify. However, verb phrases with modal verbs are considered separate verb phrases. After extracting the dependents of each verb node, the units are cleaned by removing subjects and pre-verbal modifiers. When two verbs are coordinated they are also considered a singular verb phrase (e.g. *ne sont pas d'accord et présentent de solutions différents*; 'do not agree and present different solutions').

```
[1] "Dotée d'un éventail de mots et d'expressions, la parole constitue le moyen de communication par ex
```

```
getUnits(test.sents[["b.124.1"]],
         what = "tokens",
         units = c("VERB_PHRASES"),
         paste.tokens = TRUE)
## $VERB_PHRASES
## $VERB_PHRASES[[1]]
## [1] "Dotée d' un éventail de mots et d' expressions"
##
## $VERB_PHRASES[[2]]
## [1] "constitue le moyen de communication par excellence"
```

Calculating mesures

The function `getMeasures()` can be used to get several measures of syntactic complexity from a list of sentences. The example below calculates these measures for the sentences used above.

```
 [1] "C'est un point très important."
 [2] "Ils prétendent qu'il est impossible de rééduquer un tel jeune criminel."
 [3] "Quand les lois sont contre le droit, il n'y a qu'une héroïque façon de protester contre elles : l
 [4] "Des procès qui durent des années, des déclarations d'impôts inhumainement longues et compliquées,
 [5] "Il y a quelques siècles, les empereurs pourraient modifier les règles selon leur volonté."
 [6] "Ensuite, il y a des délits dus aux problèmes personnels survenus à la maison ou à l'école."
 [7] "Est-ce que les règles sont nécessaires?"
 [8] "Il est possible que la langue disparaisse après quelques générations, car, comme dit le professeu
 [9] "Dans la prison, il n'ont plus d'éducation, il ne voient que des criminels et ils doivent devenir
```

```
[10] "La question sur une nouvelle réforme de l'État est un grand problème qui se pose aujourd'hui en B
[11] "Dotée d'un éventail de mots et d'expressions, la parole constitue le moyen de communication par e
```

```
getMeasures(examples)
## $MLS
## [1] 18.73
##
## $DIVS
## [1] 9.84
##
## $T_S
## [1] 1.18
##
## $MLT
## [1] 15.85
##
## $DIVT
## [1] 9.93
##
## $C_T
## [1] 1.77
##
## $MLC
## [1] 12.87
##
## $DIVC
## [1] 9.1
##
## $MLNP
## [1] 4.23
##
## $DIVNP
## [1] 4.01
##
## $NP_C
## [1] 2.26
```

References

Cooper, T. C. (1976). Measuring written syntactic patterns of second language learners of German. *Journal of Educational Research*, *69*(5), 176–183. https://doi.org/10.1080/00220671.1976.10884868

Csardi, G., & Nepusz, T. (2006). The igraph software package for complex network research. *InterJournal (Complex Systems)*.

Denis, P., & Sagot, B. (2012). Coupling an annotated corpus and a lexicon for state-of-the-art POS tagging. *Language Resources and Evaluation*, *46*(4), 721–736. https://doi.org/10.1007/s10579-012-9193-0

Green, N. (2011). Dependency Parsing. *WDS'11 proceedings of contributed papers, part i.* Prague.

Hunt, K. (1965). *Grammatical structures written at three grade levels.* Champaign, IL: NCTE.

Hunt, K. (1970). Do sentences in the second language grow like those in the first? *TESOL Quarterly1*, *4*(3), 195–202.

Lu, X. (2010). Automatic analysis of syntactic complexity in second language writing. *International Journal of Corpus Linguistics*, *15*(4), 474–496. https://doi.org/10.1075/ijcl.15.4.02lu

Nivre, J., Hall, J., & Nilsson, J. (2006). MaltParser : A data-driven parser-generator for dependency parsing. *LREC 2006*, 2216–2219.

Vandeweerd, N., Housen, A., & Paquot, M. (n.d.). Applying phraseological complexity measures to L2 French: A partial replication study. *International Journal of Learner Corpus Research*.