TECHNICAL REPORT

Using text-based indices to predict perceptions of the lexical richness of short French, German, and Portuguese texts written by children

Compiled by Jan Vanhove

Project repository: https://doi.org/10.17605/OSF.IO/VW4PC

Made public on November 12, 2019

Last update: November 19, 2021
(deanonymised references and acknowledgements)

Contents

Part I

Context

Chapter 1

Introduction

1.1 Purpose of this technical report

This report documents—in tedious detail—all the steps taken in the *Lexical richness* project so that it may serve as a point of reference common to all researchers working on this project and it can be referred to in research papers to avoid cramming them with all the tedious details involved in collecting, preparing and analysing the data.

1.2 Goal of this project

The primary goal of this project are to find out how well raters' perceptions of the lexical richness of short German, French, and Portuguese texts written by children can be predicted from automatically computable (i.e., algorithmic) measures of lexical diversity and lexical sophistication and some other properties such as text length, and gain insight into how different lexical aspects of a text affect its perceived lexical richness.

1.3 The text length problems

We are accutely aware that most algorithmic measures of lexical diversity (e.g., the type-token ratio) are considered to be inappropriate for use with texts as short as ours. (Most texts contain well fewer than 100 words.) However, we understand this to mean that such measures, *in and by themselves*, may not permit reliable conclusions about the actual lexical diversity/richness in short texts. It may yet be possible that a *combination* of measures permits more reliable conclusions for such texts. If so, our analyses will show this. If, by contrast, a combination of diversity and sophistication measures cannot predict tendencies

in the perceived lexical richness of such short texts, then we consider this an interesting scientific as well as practical conclusion in its own right.

We are also well aware that measures of lexical diversity, by their design, tend to correlate negatively with text length—a property which many researchers deem undesirable. Our stance with regard to this is as follows: Our interest is in modelling *human judgements* of lexical richness, not in finding the 'Holy Grail' of a measure of diversity that is both reliable and invariant with respect to text length. (The latter is a sensible goal to pursue when using these measures for, say, author identification (Tweedie & Baayen, 1998).) If such human judgements are affected by the texts' length, then this is important to find out (also see Jarvis, 2013a). To that end, we include text length among the possible predictors of judged lexical richness. If it turns out that a statistical model that includes text length as a predictor permits more reliable predictions about judged lexical richness than one without, we would be remiss—*given our goals*—not to capitalise on this. Note, furthermore, that by investigating texts of (naturally occurring) different lengths, we will be able to answer such questions as *How does the predictive power of the statistical model vary according to text length?*

A related argument is that differences in text length may affect how raters judge the lexical richness of the texts. However, text length differences occur naturally (e.g., when teachers score pupils' essays). To arbitrarily limit the text sample so as to arrive at a set of texts of comparable length would not do justice to the basic fact that the children who wrote these texts were given the same brief yet produced texts of differing lengths. How such differences affect raters' judgements is a useful thing to know.

1.4 Acknowledgements

For their help at various stages in this project, we thank Judith Berger, Katharina Karges, Meik Michalke, Carlos Pestana, Catia da Silva Parente, Fabio Soares, Carina Steiner, Isabelle Udry, all raters and pilot raters, all participating children as well as their parents and teachers.

This research project was funded by the Research Centre on Multilingualism (Fribourg, Switzerland).

Chapter 2

Text collection

This chapter summarises how the texts for the project were collected. It is based on book chapters by Lambelet et al. (2017), Desgrippes & Lambelet (2017), Desgrippes et al. (2017), and Vanhove & Berthele (2017).

2.1 Overview

A total of 482 children wrote two kinds of (short) texts at three points in time (beginning of third grade, end of third grade, and end of fourth grade). The children were roughly 8 years old at the first data collection and roughly 10 years old at the last. 233 of these children lived in French- or German-speaking Switzerland and had a Portuguese background; the others were either children living in Switzerland but without a Portuguese background or children living in Portugal.

2.2 Children

From Vanhove & Berthele (2017):

> The participants were children with Portuguese as a heritage language living in Switzerland. 114 of these children lived in the French-speaking part of Switzerland and had French as their school language; 119 lived in the German-speaking part of Switzerland and had Standard German as their school language. Additionally, three groups of participants without Portuguese as a heritage language served as comparison groups: 78 in French-speaking Switzerland, 80 in German-speaking Switzerland and 91 in Portugal. The average age of the participants at

the first data collection was 8 years and 8 months. Due to subject unavailability, not all participants were tested at each data collection.

Table 2.1: Number of participants according to Vanhove & Berthele (2017).

Group	Number of children
Portuguese–German	119
Portuguese–French	114
German comparison	80
French comparison	78
Portuguese comparison	91
Total	482

For more information on the children's sociological background, which isn't relevant for our current purposes, see Desgrippes & Lambelet (2017).

2.3 Writing tasks

From Desgrippes et al. (2017):

> Participants were asked to write two short essays at three different times of data collection (beginning of third grade, end of third grade and end of fourth grade). … [T]he exact same instructions were given each time. Portuguese participants wrote essays in their heritage language as well as in the school language (i.e. either in French or German). Comparison groups did the tasks only in the school language.
>
> …
>
> More precisely, participants were asked to:
>
> - write a letter to their aunt (respectively godmother in the heritage language version) choosing between two options for [going on a] holiday[] with her ([to the] se[a] or [to the] mountain[s] in the Portuguese version; travel[ling] by plane or car in the French/German versions);
> - narrate with details a specific day (last school [trip] in the French/German version, a day of [the] last holidays in the Portuguese version).

Table 2.2: Number of texts produced according to Desgrippes et al. (2017).

Language of text	Text type	Group	Time 1	Time 2	Time 3	Total
German	Argumentative	German comparison	79	79	74	232
		Portuguese–German	107	95	92	294
	Narrative	German comparison	77	79	74	230
		Portuguese–German	106	93	92	291
	Total German		369	346	332	1047
French	Argumentative	French comparison	77	76	70	223
		Portuguese–French	102	104	101	307
	Narrative	French comparison	73	76	68	217
		Portuguese–French	105	103	101	309
	Total French		357	359	340	1056
Portuguese	Argumentative	Portuguese comparison	75	73	72	220
		Portuguese–German	104	99	93	296
		Portuguese–French	102	104	102	308
	Narrative	Portuguese comparison	74	74	71	219
		Portuguese–German	106	98	97	301
		Portuguese–French	104	106	103	313
	Total Portuguese		565	554	538	1657
Grand total			1291	1259	1210	**3760**

2.3.1 Literal instructions

2.3.1.1 Portuguese

Preferes passar as tuas férias no mar ou na montanha ?

A tua madrinha convidou-te para passar as férias com ela. Ela ainda não decidiu onde vai passar as férias e quer saber a tua opinião.

Preferes passar as tuas férias no mar ou na montanha ?

Escreve uma carta à tua madrinha na qual tu lhe explicas a tua escolha. Não te esqueças de lhe dar **ao menos três razões** pelas quais tu preferes passar as férias no mar ou na montanha. **Tenta convencê-la !**

2.3.1.2 French

Préfères-tu voyager en voiture ou en avion ?

Ta tante t'a invité pour passer les vacances avec elle. Elle n'a pas encore décidé du moyen de transport et veut savoir ton opinion.

Préfères-tu voyager en voiture ou en avion ?

Ecris une lettre à ta tante où tu lui expliques ce que tu préfères. N'oublie pas de lui donner **au moins trois raisons** pour lesquelles tu préfères l'avion ou la voiture. **Essaie de la convaincre !**

2.3.1.3 German

Reist du lieber mit dem Auto oder mit dem Flugzeug?

Deine Tante hat dich eingeladen, mit ihr Ferien zu verbringen. Sie hat sich noch nicht entschieden, welches Transportmittel ihr benützen werdet. Sie möchte deine Meinung wissen:

Reist du lieber mit dem Auto oder mit dem Flugzeug, wenn du in die Ferien gehst?

Schreibe deiner Tante einen Brief, wo du ihr erzählst, was du lieber hast. Vergiss nicht deiner Tante **mindestens drei Gründe** anzugeben, weshalb du lieber das Auto oder das Flugzeug nimmst. **Versuche sie zu überzeugen.**

2.3.2 Narrative texts

2.3.2.1 Portuguese

Um dia de férias

O próximo número de uma revista para crianças será dedicado às férias. Vais escrever um artigo para esta revista onde falas de um dia de tuas férias.

Pensa num dia das tuas últimas férias : Onde é que foste ? O que aconteceu ? O que fizeste durante esse dia desde o início até ao fim ?

Conta esse dia aos leitores da revista com o máximo de detalhes possível.

2.3.2.2 French

Ta dernière course d'école

Le prochain numéro d'un magazine pour enfants sera consacré aux courses d'école. Tu vas écrire un article pour ce magazine pour y raconter ta dernière course d'école.

Repense à ta dernière course d'école : Où es-tu allé ? Qu'est-ce qui s'est passé ? Qu'as-tu fait pendant cette sortie de classe depuis le début jusqu'à la fin de la journée ?

Raconte aux lecteurs du magazine le plus possible de détails sur cette journée.

2.3.2.3 German

Deine letzte Schulreise

Die nächste Nummer einer Jugendzeitschrift handelt von Schulreisen. Du sollst nun ein Artikel für diese Zeitschrift schreiben, wo du von deiner letzten Schulreise berichtest.

Denke an deine letzte Schulreise zurück: Wo bist du hingegangen? Was ist dort alles passiert? Was hast du von morgens bis abends gemacht?

Erzähle den Lesern dieser Jugendzeitschrift so viel wie möglich über diese Reise.

Part II

Preparing and tagging the corpus

Chapter 3

Preparing the texts for tagging and rating

This chapter documents how the texts were prepared for part-of-speech (POS) tagging and for getting rated by human judges; the following chapters detail the manual tweaks that had to be effected to ensure that the POS tagging went smoothly.

3.1 Transcribing the texts

The hand-written French, German and Portuguese texts were transcribed to text files by student assistants. The transcribers were instructed to render the text verbatim, i.e., errors and code-switches and all.

3.2 Correcting the texts

Our goal is to investigate the lexical richness of the children's texts. However, grammatical and orthographic errors in the texts would likely negatively affect human raters' judgements and would impact on algorithmic measures of lexical diversity and sophistication, too. For these reasons, the children's texts were corrected grammatically, orthographically, and in terms of their punctuation by student assistants.

The guiding principle when correcting the texts was to adapt the lexical items occurring in the texts as little as possible. For instance, inappropriate or missing suffixes were corrected (e.g., German adjectives in the wrong case or missing French plural -*s*), but inappropriate lexical choices were not changed.

Foreign words (e.g., English words in German texts, or Portuguese words in French texts) were not replaced by their native counterpart. By contrast, typical Swiss-German words such as *Znüni* and *grillieren*, which may be unfamiliar to German raters and were usually not recognised by the tagging software, were replaced by their Northern-German counterparts (e.g., *Pausenbrot*, *grillen*).

3.3 Anonymisation

For both the POS tagging and the human judgements, the texts were rendered anonymous: proper names, brand names, and town names were replaced by placeholders. The anonymised words were all written in all-caps (e.g., PRENOM, NOM, LIEU etc.) for the ratings. For POS tagging, these needed to be replaced by other placeholders lest placeholders such as *LIEU* and *NOM* be taken for the actually occurring words *lieu* and *nom*. Anonymisation placeholders were ignored for POS tagging.

Table 3.1: French anonymisation placeholders (used when rating the texts) and their replacements for POS tagging.

Placeholder	Replacement
ILLISIBLE	illegible
LIEU	Zurich
MARQUE	brandreplacement
NOM	Schmidt
PRENOM	Daniel

Table 3.2: German anonymisation placeholders (used when rating the texts) and their replacements for POS tagging.

Placeholder	Replacement
MARKE	brandreplacement
NAME	Schmidt
ORT	Zürich
UNLESERLICH	illegible
VORNAME	Daniel

Table 3.3: Portuguese anonymisation placeholders (used when rating the texts) and their replacements for POS tagging.

Placeholder	Replacement
APELIDO	Schmidt
ILISIVEL	illegible
LUGAR	Zurich
MARCA	brandreplacement
NOME	Daniel

3.4 Additional automated text cleaning

Even after the texts were manually corrected, we still had to clean them a bit before they could be tagged and rated. This mostly involved changing the file encodings so that the text files were properly read in on a Linux machine, correcting some remaining common misspellings, and changing some correct spellings that the tagging software could not cope with (due to recent spelling reforms) into obsolete spellings that it could cope with. To effect this automated cleaning, we made use of Linux bash commands as well as `R`.

3.4.1 Bash commands

This section is mostly for our own reference.

For each language, we put all texts (`txt` file format) in a directory, then navigated to this directory in a Linux terminal, and then ran the following commands in the terminal. (Some of these commands don't render accurately in the PDF; always double-check with the source code if you run into problems.)

```
## Some file names contain superfluous spaces.
## These need to be removed:
rename 's/ //g' *
```

The following bash commands, and indeed all `R` scripts, are only work if the file encoding of the `txt` files is UTF-8. To automatically convert a directory of `txt` files to UTF-8, create a bash script with the following content:

```
#!/bin/bash

# Find the current encoding of the file
encoding=$(file -i "$2" | sed "s/.*charset=\(.*\)$/\1/")
```

```
# If the encoding is 'unknown-8bit', this
# always corresponds to iso-8859-1 (for us).
if [ "${encoding}" == "unknown-8bit" ]
then
encoding="iso-8859-1"
fi

# Reencode file if encoding differs from target
if [ ! "$1" == "${encoding}" ]
then
echo "recoding from ${encoding} to $1 file : $2"
recode ${encoding}..$1 $2
fi
```

Then run the following command in the terminal:

```
find . -name "*.txt" -exec recodeifneeded utf-8 {} \;
```

When all `txt` files are encoded as UTF-8, run the following commands in the order specified.

3.4.1.1 French

These commands were applied to the French texts only.

```
## The first few commands apply to both the
## texts to be shown to the raters and the
## versions to be tagged.

## The texts were formatted on a Windows system
## but were to be tagged on a Linux system.
## Convert DOS file format to Unix:
find . -type f -exec dos2unix {} \;

## DOS line ends tend to cause difficulties;
## replace them by a space:
sed -i ':a;N;$!ba;s/\n/ /g' *

## Carriage returns ('\r') cause difficulties, too;
## remove them:
sed -i 's/\r//g' *
```

```
## Same for byte-order marks.
## This command needs to be run SEVERAL times:
sed -i 's/\xEF\xBB\xBF//' *

## Replace ’ (stylised apostrophe)
## by ' (straight apostrophe)
sed -i "s/[’]/' /g" *

#-----------------------------------------------------
## The following steps are for the computations only;
## don't apply them to the texts shown to the raters:

## Remove parentheses:
sed -i 's/[()]//g' *

## Replace special symbols occurring in
## some of the texts by a space:
sed -i 's/[½-"„«»+]/ /g' *

## Replace all-caps placeholders by their tagging counterparts
sed -i 's/ILLISIBLE/illegible/g' *
sed -i 's/LIEU/Zurich/g' *
sed -i 's/MARQUE/brandreplacement/g' *
sed -i 's/PRENOM/Daniel/g' *
sed -i 's/NOM/Schmidt/g' *

## Convert to lower case:
sed -i 's/\(.*\)/\L\1/' *

## Misspellings: Replace mini-golf by minigolf:
sed -i 's/mini-golf/minigolf/g' *

## Replace : and - by : followed by a space.
## Rationale: Some dashes and colons were not
## followed by a space so that the tagging software
## thinks they're in the middle of words.
sed -i 's/[:-]/: /g' *

## Remove degree characters:
```

```
sed -i 's/[°]//g' *

## Remove non-breaking spaces
sed -i 's/\xc2\xa0/ /g' *

## Replace multiple successive dots (..., .., ..... etc.)
## by a single dot.
sed -i -r 's/\.{2,}/. /' *

## Replace triple dots, exclamation marks, arrows
## by single dots
sed -i 's/[!…→]/. /g' *

## Then create subfolder 'Fixed'
mkdir Fixed
```

3.4.1.2 German

These commands were applied to the German texts only.

```
## The texts were formatted on a Windows system
## but were to be tagged on a Linux system.
## Convert DOS file format to Unix:
find . -type f -exec dos2unix {} \;

## DOS line ends tend to cause difficulties;
## replace them by a space:
sed -i ':a;N;$!ba;s/\n/ /g' *

## Carriage returns ('\r') cause difficulties, too;
## remove them:
sed -i 's/\r//g' *

## Same for byte-order marks.
## This command needs to be run SEVERAL times:
sed -i 's/\xEF\xBB\xBF//' *

## Replace ’ (stylised apostrophe)
## by ' (straight apostrophe)
sed -i "s/[’]/' /g" *

#-----------------------------------------------------------------------
## The following steps are for the computations only;
## don't apply them to the texts shown to the raters:

## Replace Puren-Fest by Purenfest
sed -i 's/Puren-Fest/Purenfest/g' *

## Remove parentheses:
sed -i 's/[()]//g' *

## Replace special symbols occurring
## in some of the texts by a space:
sed -i 's/[½-"„«»+]/ /g' *

## Replace all-caps placeholders
```

```
## by their tagging counterparts
sed -i 's/MARKE/brandreplacement/g' *
sed -i 's/ORT/Zürich/g' *
sed -i 's/UNLESERLICH/illegible/g' *
sed -i 's/VORNAME/Daniel/g' *
sed -i 's/NAME/Schmidt/g' *

## Replace : and - by : followed by space
sed -i 's/[:-]/: /g' *

## Remove degree characters:
sed -i 's/[°]//g' *

## Remove non-breaking spaces
sed -i 's/\xc2\xa0/ /g' *

## Replace multiple successive dots (..., .., ..... etc.)
## by a single dot.
sed -i -r 's/\.{2,}/. /' *

## Replace triple dots, exclamation marks, arrows
## by single dots
sed -i 's/[!…→]/. /g' *

## Then create subfolder 'Fixed'
mkdir Fixed
```

3.4.1.3 Portuguese

These commands were applied to the Portuguese texts only.

```
## In the Portuguese texts, some invisible
## Unicode characters cause problems on Linux.
## The following commands remove
## these characters from the txt files.
## http://stackoverflow.com/questions/43090168/encoding-008d-character/43094010

## 008D
CHARS=$(python -c 'print u"\u008D".encode("utf8")')
sed -i 's/['"$CHARS"']//g' *

## 0090
CHARS=$(python -c 'print u"\u0090".encode("utf8")')
sed -i 's/['"$CHARS"']//g' *

## 008F
CHARS=$(python -c 'print u"\u008F".encode("utf8")')
sed -i 's/['"$CHARS"']//g' *

## à was miscoded as 0088
CHARS=$(python -c 'print u"\u0088".encode("utf8")')
sed -i 's/['"$CHARS"']/à/g' *

## The texts were formatted on a Windows system but were to be tagged on a Linux system.
## Convert DOS file format to Unix:
find . -type f -exec dos2unix {} \;

## DOS line ends tend to cause difficulties; replace them by a space:
sed -i ':a;N;$!ba;s/\n/ /g' *

## Carriage returns ('\r') cause difficulties, too; remove them:
sed -i 's/\r//g' *

## Same for byte-order marks.
## This command needs to be run SEVERAL times:
sed -i 's/\xEF\xBB\xBF//' *
```

```
## Replace ’ (stylised apostrophe) by ' (straight apostrophe)
sed -i "s/[’]/' /g" *

## Misspellings (computer-related)
## danar and danamos are actually dançar and dançamos
sed -i 's/danar/dançar/g' *
sed -i 's/danamos/dançamos/g' *

## Other remaining spelling errors
## (and some Brazilian spellings)
sed -i 's/ iamos/ íamos/g' *
sed -i 's/Iamos /Íamos /g' *
sed -i 's/conosco/connosco/g' *
sed -i 's/ á / à /g' *
sed -i 's/Á /À /g' *
sed -i 's/para-quedas/paraquedas/g' *
sed -i 's/quilômetros/quilómetros/g' *
sed -i 's/sobremas/sobremesa/g' *
sed -i 's/penáltis/penaltis/g' *
sed -i 's/vá-lá/vá lá/g' *
sed -i 's/Vá-lá/Vá lá/g' *
sed -i 's/ vocs/ vocês/g' *
sed -i 's/Vocs /Vocês /g' *
sed -i 's/ vôlei/ vólei/g' *
sed -i 's/cocô/cocó/g' *
sed -i 's/jacúzi/jacuzzi/g' *
sed -i 's/espetácular/espetacular/g' *

## Some texts use PRENOME instead of NOME
## as the placeholder for first names.
sed -i 's/PRENOME/NOME/g' *

#-----------------------------------------------------------------------
## The following steps are for the computations only;
## don't apply them to the texts shown to the raters:

## Remove parentheses:
sed -i 's/[()]//g' *
```

```
## Replace special symbols occurring in some of the texts by a space:
sed -i 's/[½-"„«»+]/ /g' *

## Replace all-caps placeholders
## by their tagging counterparts
sed -i 's/MARCA/brandreplacement/g' *
sed -i 's/ILISIVEL/illegible/g' *
sed -i 's/NOME/Daniel/g' *
sed -i 's/APELIDO/Schmidt/g' *
sed -i 's/LUGAR/Zurich/g' *

## Convert to lower case
sed -i 's/\(.*\)/\L\1/' *

## For French and German,
## we replaced : or - by : followed by space.
## For Portuguese, this separates clitics from their stems
## (which is what we want).
## But it also split up fim-de-semana into three words.
## We may want to rewrite such words to fim_de_semana.
## For reference:
## The file FrequencyCorpora/hyphenatedPortugueseWords.txt
## contains all hyphenated words in SUBTLEX.
## It's easier to take care of this in R in the cleanTexts script.

## Replace : by : followed by space
sed -i 's/[:]/: /g' *

## Replace o.k. and p.s. by o_k and p_s
## (single words containing dots)
sed -i 's/o\.k\./o_k/g' *
sed -i 's/p\.s\./p_s/g' *

## Remove degree characters:
sed -i 's/[°]//g' *

## Remove non-breaking spaces
sed -i 's/\xc2\xa0/ /g' *
```

```
## Replace multiple successive dots (..., .., ..... etc.)
## by a single dot
sed -i -r 's/\.{2,}/. /' *

## Replace triple dots, exclamation marks, arrows
## by single dots
sed -i 's/[!…→]/. /g' *

## Then create subfolder 'Fixed'
mkdir Fixed
```

Note that French and Portuguese but *not* German texts are converted to lowercase for the computations. For French, converting the texts to lowercase causes fewer problems when tagging; for German, it causes more problems. For Portuguese, it does not really matter one way or the other, but lower-case is slightly easier to deal with afterwards.

3.4.2 Additional pre-tagging cleaning in `R`

3.4.2.1 French

The script `RFunctions/CleanFrenchTexts.R` accomplished the final cleaning steps prior to tagging for French texts. It accomplished the following:

1. Any words containing Arabic numerals *except for* ordinal numbers also containing *ème* were removed. This removed cardinal numbers, numeric dates, times etc. The logic behind retaining words such as *ème* is that the *ème* part suggests that the children know the word in question.

2. Multiword sequences such as *tandis que* were rewritten as single words (*tandis_que*) so that they could be looked up in the Lexique corpus. This concerned the following words: aujourd'hui, tandis que, tandis qu', tandis qu, jusqu'à, jusqu'au, jusqu'aux, c'est-à-dire, quelqu'un, parce que, parce qu', parce qu.

3. The following clitics were detached from their stems: -je, -moi, -tu, -toi, -le, -la, -lui, -elle, -nous, -vous, -ils, -elles, -eux, -les, -ce, -là, -y, -on. For instance, *allons-y* was rewritten as *allons y*.

4. The *î* and *û* graphemes were recently abolished in French spelling (in most but not in all words) but are still used in `koRpus/TreeTagger`. We therefore replaced the *i* and *u* spellings in the texts by the *î* and *û* ones that the tagging software is familiar with to avoid problems with POS tagging down the road.

Table 3.4: French spelling variants.

Spelling in texts	Replaced by
boite	boîte
boites	boîtes
connait	connaît
connaitre	connaître
diné	dîné
diner	dîner
disparaitre	disparaître
entrainer	entraîner
ile	île
maitre	maître
maitres	maîtres
maitresse	maîtresse
maitresses	maîtresses
paraitre	paraître
plaire	plaîre
plait	plaît
reconnaitre	reconnaître
trainer	traîner

5. Numerically written ordinals were rewritten, i.e., *1er* becomes *premier*, *9ème* becomes *neuvième*.

6. Whitespace was removed when immediately followed by punctuation. Sequences of dots were replaced by a single dot.

The cleaned texts were stored in the directory `FrenchTexts/Fixed`.

3.4.2.2 German

The script `RFunctions/CleanGermanTexts.R` accomplished the final cleaning steps prior to tagging for German texts. Specifically, any words containing Arabic numerals were removed, sequences of whitespace and punctuation were replaced by punctuation only, and sequences of multiple dots were replaced by single dots. The cleaned texts were stored in the directory `GermanTexts/Fixed`.

3.4.2.3 Portuguese

The script `RFunctions/CleanPortugueseTexts.R` accomplishes the final cleaning steps prior to tagging for Portuguese texts. It accomplishes the following:

1. Any words containing Arabic numerals were removed.
2. A recent Portuguese spelling reform abolished some silent letters. For instance, *actriz* was rewritten as *atriz*. The tagging software only recognised the old spelling, however, so we needed to change the spellings in the corrected texts without the silent letters to the ones with the silent letters.

Table 3.5: Portuguese spelling variants.

Spelling in texts	Replaced by
adotar	adoptar
atividade	actividade
atividades	actividades
atriz	actriz
batizado	baptizado
diretamente	directamente
elétrica	eléctrica
espetacular	espectacular
espetaculares	espectaculares
espetáculo	espectáculo
espetáculos	espectáculos
esquí	esqui
insetos	insectos
noturnos	nocturnos
objetos	objectos
ótima	óptima
ótimas	óptimas
ótimo	óptimo
ótimos	óptimos
protetor	protector
receção	recepção
reflete	reflecte
trator	tractor
tratores	tractores

3. Whitespace followed by punctuation was changed into punctuation only, and multi-dot sequences were replaced by a single dot. For some reason, though, the tagging software doesn't recognise lowercase words that follow a dot—the Portuguese texts were converted to lowercase—so that we then *added* another whitespace to any dots.

4. The tagging software automatically splits up hyphenated words into their constituent parts. Most of these words are verb–clitics combinations that were split up in their constituent parts for the frequency corpus, too. For these words, the tagging software's behaviour is beneficial to us. For some other words, however, the hyphenated word forms actually do occur in the frequency corpus (e.g., *fim-de-semana* should not be split up into *fim*, *de* and *semana*). To circumvent the software's default behaviour for hyphenated words that occur in the frequency corpus (SUBTLEX-PT, Soares et al., 2015), we extracted all hyphenated words from the SUBTLEX-PT corpus (`FrequencyCorpora/hyphenatedPortugueseWords.txt`) and replaced the hyphens by underscores (i.e., *fim-de-semana* became *fim_de_semana*). Then we replaced all hyphenated words occurring in both the texts and SUBTLEX-PT by their underscored versions.

The cleaned texts were stored in `PortugueseTexts/Fixed`.

3.5 Texts ill-suited for rating

A handful of texts were manually identified—some by the researchers, some by the pilot raters—as being nonsensical, e.g., because they didn't contain a single finite verb. In one case, the pupil copied entire sentences from the instruction prompt; this text was also considered ill-suited for rating.

This is the list of texts that were considered nonsensical and hence weren't included in the rating procedure.

- French:

```
## AA_PLF_12_arg_F_C_2
## AD_PLF_17_narr_F_C_2
## Y_PNF_22_narr_F_C_1
## Z_PLF_4_narr_F_C_1
```

- German:

```
## M_PLD_10_arg_D_C_1
## N_CD_13_narr_D_C_2
```

For Portuguese, our research assistant suggested the following texts would not be usable in the rating procedure as they are grammatically nonsensical or chock-full of French or German loans that would not be understandable to Portuguese speakers.

- Portuguese:

```
## AC_PLD_4_narr_P_C_3
## AD_PLF_13_narr_PC1
## AD_PLF_17_narr_P_C_1
## AE_PLF_1_arg_P_C_1
## AE_PLF_6_narr_P_C_1
## AE_PLF_6_narr_P_C_3
## AJ_CP_4_narr_P_C_1
## F_PLD_8_arg_P_C_2
## H_PLD_11_arg_PC1
## O_PLD_11_narr_P_C_2
## O_PLD_13_narr_P_C_1
## O_PLD_13_narr_PC1
## O_PLD_5_arg_P_C_1
## Y_PNF_27_narr_PC2
## Y_PNF_29_arg_P_C_1
## Y_PNF_29_arg_PC1
## Y_PNF_29_narr_P_C_1
## Y_PNF_8_arg_P_C_3
## Y_PNF_8_narr_P_C_3
```

Chapter 4

Part-of-speech tagging

4.1 Software

The cleaned texts (see Chapter 3.4) were tagged using the `koRpus` package (Michalke, 2017) for `R`. This package, in turn, is based on the TreeTagger software (Schmid, 1994).

While `koRpus` supports French, German and Portuguese in principle, some patching is needed to get the Portuguese tagging run smoothly. The patches are defined in the file `lang.support-pt_jv.R` (based on the `koRpus.lang.pt` package; see https://reaktanz.de/R/pckg/koRpus.lang.pt/), which defines the tags used by the Portuguese `TreeTagger` to make them interpretable for `koRpus`.

4.2 Stopwords

The following tokens are first names (all converted to *Daniel* for tagging), family names (all converted to *Schmidt*), organisations (*NASA*), places, planets, brands (all converted to *brandreplacement*), particular days, the Portuguese names for the months (which were not included in SUBTLEX-PT), or indicators for illegible words. They were defined as 'stopwords', i.e., words that are disregarded in the analyses. (Capitalisation is disregarded for Portuguese and French, but not for German.)

Table 4.1: Stopwords.

Stopword
Aare
abril

açoriana
agosto
atlántico

atlántida
brandreplacement
brandreplacements
Daniel
dezembro

Dufourspitze
États-Unis
fevereiro
halloween
illegible

israel
Italie
janeiro
julho
junho

maio
maiorca
março
marte
Matterhorn

nasa
novembro
Olten
outubro
páscoa

Portugal
Rigi
saturno
Schmidt
Schweiz

septembro
setembro
Suisse

swissair
Zurich

Zürich

4.3 Manual changes to the TreeTagger software

We effected the following minor changes to the `TreeTagger` software.

French: The file `french-chunker.par` (in the `treetagger/lib/` directory) was renamed to `french-chunker-utf8.par` to solve a "file not found" issue. The file itself wasn't changed.

German: The string `Berg.` was removed from the `german-abbreviations-utf8` file (also in the `treetagger/lib/` directory) since this string is never an abbreviation (for *Bergisch*) but always a noun ('mountain') followed by a dot.

Portuguese: `TreeTagger` splits up a set of contracted forms (e.g., *do* into *de* and *o*) but not all of them (e.g., not *deste* into *de* and *este*). This inconsistency causes problems down the road so that it was easiest to not split up contracted forms at all. Additionally, the software splits up Portuguese stem–clitic combinations into their stems and clitics, which is desirable for our purposes. However, some clitics were missing from the software's specification file (more specifically *-lo*, *-la*, *-los* and *-las*). To solve both problems, the file `portuguese-splitter.perl` (in the `treetagger/cmd/` directory) was changed by commenting out all lines pertaining to contracted forms and by adding the four missing elements to the list of clitics.

4.4 Manual tweaks to the tags

While `TreeTagger/koRpus` does an admirable job on the whole, it assigned some tokens in our texts to the wrong lemma, and others to the wrong part-of-speech. Some rarer words were tagged as `<unknown>` and needed manual lemmatisation. Such errors can be rectified using the `correct.tag()` function in the `koRpus` package. This function was elaborated in the `RFunctions/fun_correct.tags.R` script (courtesy of Katharina Karges), which requires as its input vectors of tokens and their associated lemmata. These are defined below.

The tweaks outlined below are the result of an iterative process of tagging the texts, outputting all tokens tagged as `<unknown>`, double-checking a random subset of tagged texts, fixing the problems, and retagging the texts. If new texts are to be tagged, they may well yield different problems from the ones solved here (e.g., new German compound nouns).

That said, the most common problems, such as those involving contracted preposition–article combinations and personal pronouns, should already have been taken care of.

4.4.1 French

4.4.1.1 Articles

The apostrophed articles *l'* and *d'* should be assigned to their unapostrophed counterparts. Additionally, *des* should be assigned to *une* (which is the default indefinite article in `TreeTagger/koRpus`) rather than to *du* (which is a contraction of *de* and *les*). Overriding the `TreeTagger/koRpus` defaults in these cases will be wrong in some cases but right in most.

Note that *d'* can sometimes be a preposition; here we assign *d'* to the article category across the board.

Table 4.2: French tokens manually labelled as articles and the lemma they were assigned to.

Token	Lemma
d	de
des	une
l	le

4.4.1.2 Conjunctions

Apostrophed conjunctions should be assigned to their unapostrophed counterparts. Note that multiword conjunctions (*tandis que*, *parce que*) were rewritten as single words when cleaning the texts.

Table 4.3: French tokens manually labelled as conjunctions and the lemma they were assigned to.

Token	Lemma
parce_qu	parce_que
parce_que	parce_que
qu	que
tandis_qu	tandis_que
tandis_que	tandis_que

4.4.1.3 Prepositions

Complex prepositions, specifically *jusqu'à, jusqu'au, jusqu'aux*, should all be assigned to the same lemma, whereas *jusqu'* should be assigned to *jusque*.

Note: Lexique 3 has separate lemmas for *jusqu'* and *jusque*.

Table 4.4: French tokens manually labelled as prepositions and the lemma they were assigned to.

Token	Lemma
jusqu	jusque
jusqu_à	jusqu_à
jusqu_au	jusqu_à
jusqu_aux	jusqu_à

4.4.1.4 Interjections

Table 4.5: French tokens manually labelled as interjections and the lemma they were assigned to.

Token	Lemma
bonjour	bonjour
merci	merci
oups	oups
salut	salut

4.4.1.5 Nouns

There are several nouns that `TreeTagger/koRpus` doesn't seem to pick up on or doesn't recognise as nouns. Their lemmas are identical to the tokens. Then there were a couple of words that weren't assigned to the correct lemma due to problems with plural formation or alternative spellings. Some of these concern spelling mistakes which should be fixed by now, but we leave them in here for completeness or in case some slipped through the maze.

Table 4.6: French tokens manually labelled as nouns and the lemma they were assigned to.

Token	Lemma
accrobranche	accrobranche

acrostiche	acrostiche
acrostiches	acrostiche
ami	ami
amie	ami
animaux	animal
aperçu	aperçu
arrivée	arrivée
artiste	artiste
aubergine	aubergine
automobile	automobile
baby-foots	baby-foot
barbe-à-papa	barbe-à-papa
bêtise	bêtise
bob-luge	bob luge
boucher	boucher
cachecache	cache-cache
cd	CD
chatte	chat
chips	chips
copain	copain
copine	copain
copines	copain
cours	cours
course	course
cruz	cruz
écolier	écolier
écolière	écolier
electroménager	électroménager
esquimau	esquimau
étoile	étoile
evaluation	évaluation
fildeféristе	fildeféristе
fils	fils
fois	fois
fondue	fondue
forces	force

fun	fun
grenadine	grenadine
hobbies	hobby
hôtesse	hôtesse
hôtesses	hôtesse
iles	île
internet	internet
jour-là	jour là
journaux	journal
karaoké	karaoké
locomotive	locomotive
maîtresse	maîtresse
malaises	malaise
mardi	mardi
marins	marin
matelos	matelot
matériel	matériel
ménagères	ménagère
mesdames	madame
messieurs	monsieur
mini-golf	minigolf
minigolf	minigolf
mois	mois
monoluge	monoluge
monstre	monstre
objectifs	objectif
orchestre	orchestre
ordinateur	ordinateur
ovomaltine	Ovomaltine
parcours	parcours
parents	parent
pédibus	pédibus
phrase-histoire	phrase-histoire
phrases-histoires	phrase-histoire
piquenique	pique-nique
plénum	plénum

portemonnaie	porte-monnaie
porters	porté
portés	porté
portrait	portrait
pourcent	pourcent
pourcents	pourcent
préalpes	préalpes
princesse	prince
public	public
quetchua	quetchua
règles	règle
revue	revue
roller	roller
sandwiches	sandwich
school-rickshaw	school-rickshaw
secondes	seconde
sens	sens
slides	slide
snow-parc	snow-parc
tâche	tâche
tamale	tamale
télécabine	télécabine
télécabines	télécabine
torche	torche
vase	vase
véhicule	véhicule
vidéo	vidéo
virelangues	virelangue
volleyball	volleyball
voyage	voyage
vtt	vtt
wc	wc
yéti	yéti
yoyo	yoyo
zèbre	zèbre

4.4.1.6 Adjectives

Table 4.7: French tokens manually labelled as adjectives and the lemma they were assigned to.

Token	Lemma
2ème	2ème
2èmes	2ème
3èmes	3ème
bangladais	bangladais
bleu	bleu
chère	cher
cools	cool
crus	cru
déconfortable	déconfortable
féérique	féérique
folle	fou
multi-fruits	multi-fruits
parallèlles	parallèle
petits	petit
prochaine	prochain
rayé	rayé
sapiens	sapiens
suisse-allemand	suisse-allemand
tout-terrain	tout-terrain
tyrolienne	tyrolien
vita	vita

4.4.1.7 Adverbs

Note that *aujourd'hui* and *c'est-à-dire* were rewritten when cleaning the texts.

Table 4.8: French tokens manually labelled as adverbs and the lemma they were assigned to.

Token	Lemma
aujourd_hui	aujourd_hui
c_est_à_dire	c_est_à_dire

hyper	hyper
n	ne
ok	ok
par-derrière	par-derrière

4.4.1.8 Verbs

Some imperative forms should be assigned to the correct infinitive. This list also includes some superfluous entries, namely some misspellings that should've been fixed already.

Table 4.9: French tokens manually labelled as verb imperatives and the lemma they were assigned to.

Token	Lemma
ecoute	écouter
ecoutez	écouter
ecris	écrire
ecrivez	écrire
essayez	essayer
ferme	fermer
lisez	lire

Some present tense forms, including misspellings that should've already been fixed, should be assigned to the correct infinitive.

Table 4.10: French tokens manually labelled as present tenses and the lemma they were assigned to.

Token	Lemma
achètes	acheter
chatouille	chatouiller
dépens	dépendre
espère	espérer
fini	finir
ouvrent	ouvrir
plait	plaîre
plaît	plaîre
pleure	pleurer

ranger	ranger
sommes	être
suis	être

Infinitives.

Table 4.11: French tokens manually labelled as verb infinitives and the lemma they were assigned to.

Token	Lemma
bruler	brûler
essayer	essayer
payer	payer

Imperfect tense (*imparfait*).

Table 4.12: French tokens manually labelled as imperfect tenses and the lemma they were assigned to.

Token	Lemma
étais	être
trainait	traîner

Past participles.

Table 4.13: French tokens manually labelled as past participles and the lemma they were assigned to.

Token	Lemma
arrivées	arriver
essayé	essayer
plu	plaîre
resonné	resonner
soupé	souper

4.4.1.9 Pronouns

Table 4.14: French tokens manually labelled as pronouns and the lemma they were assigned to.

Token	Lemma
c	ce
celui-là	celui-là
j	je
m	me
quelqu_un	quelqu_un
s	se
soi-même	soi-même
t	te

4.4.1.10 Abbreviations

Table 4.15: French tokens manually labelled as abbreviations and the lemma they were assigned to.

Token	Lemma
cm	cm
fr	fr
km	km

4.4.1.11 Foreign words

Table 4.16: French tokens manually labelled as foreign words and the lemma they were assigned to.

Token	Lemma
do	do

4.4.1.12 Symbols

The stand-alone symbols =, |, ©, ! were disregarded during tagging.

4.4.2 German

4.4.2.1 Nouns

Most tweaks concern rare or newly constructed (but perfectly fine) compound nouns, Swiss spelling variants of common German words, some typical Swiss words, a couple of nouns that were assigned to a verb lemma, and a couple of plurals that were not recognised.

Table 4.17: German tokens manually labelled as nouns and the lemma they were assigned to.

Token	Lemma
Abendgruppe	Abendgruppe
Bachwanderung	Bachwanderung
Badekleider	Badekleider
Bahnen	Bahn
Beinschmerzen	Beinschmerzen
Berg	Berg
Berge	Berg
Bergen	Berg
Bienennest	Bienennest
Boxenstopps	Boxenstopp
Brückenreise	Brückenreise
Car	Car
Chihuahua	Chihuahua
Dinoskelett	Dinoskelett
Doofmann	Doofmann
Drache	Drache
Drachen	Drache
Egli	Egli
Erfahrungskugeln	Erfahrungskugel
Ersatzkleider	Ersatzkleider
Essensservice	Essensservice
Fantasiegeschichten	Fantasiegeschichte
Felsrutsch	Felsrutsch
Feuerplatz	Feuerplatz
Fische	Fisch
Fischen	Fisch

Franken	Franken
Frühlingsferien	Frühlingsferien
Fünfsternehotel	Fünfsternehotel
Glace	Glace
Gletschergarten	Gletschergarten
Gletscherhaus	Gletscherhaus
Grillwettbewerb	Grillwettbewerb
Grund	Grund
Gründe	Grund
Gründen	Grund
Gruppenfarben	Gruppenfarbe
Gruselgeschichte	Gruselgeschichte
Gruselgeschichten	Gruselgeschichte
Grüssen	Gruß
Gummibärchen	Gummibär
Helvetiens	Helvetien
Hexenweg	Hexenweg
Hosenladen	Hosenladen
Hypokaustenheizung	Hypokaustenheizung
Jungs	Jung
Katzenschwanz	Katzenschwanz
Katzenschwänze	Katzenschwanz
Kerosin	Kerosin
Kinderweltkrieg	Kinderweltkrieg
Klassenlager	Klassenlager
Kletterpark	Kletterpark
Kussrunde	Kussrunde
Lachanfall	Lachanfall
Lieblingstransport	Lieblingstransport
Massageplatz	Massageplatz
Metallhake	Metallhake
Mittagessen	Mittagessen
Morgen	Morgen
Morgengruppe	Morgengruppe
Morgensport	Morgensport
Paketzentrum	Paketzentrum

Pfeilbogen	Pfeilbogen
Pferdezahn	Pferdezahn
Pflegesalbe	Pflegesalbe
Pingpongschläger	Pingpongschläger
Planetenweg	Planetenweg
Plastikschlange	Plastikschlange
Popcorn	Popcorn
Purenfest	Purenfest
Rangverkündigung	Rangverkündigung
Räucherwurst	Räucherwurst
Reisegerät	Reisegerät
Rettungsarten	Rettungsart
Rettungsrutsche	Rettungsrutsch
Rochen	Rochen
Schaukeln	Schaukel
Schlangenbrot	Schlangenbrot
Schlangengeräusche	Schlangengeräusche
Schokofabrik	Schokofabrik
Schülerreise	Schülerreise
Schullager	Schullager
Schulreise	Schulreise
Sechstklässler	Sechstklässler
Spiegellabyrinth	Spiegellabyrinth
Stunden	Stunde
Tunnelgoal	Tunnelgoal
Übernachtungsparty	Übernachtungsparty
Umwelteinsatz	Umwelteinsatz
Verkehrshaus	Verkehrshaus
Waldsofa	Waldsofa
Waldwoche	Waldwoche
Wanderkleidung	Wanderkleidung
Wasserbar	Wasserbar
Wasserrutschbahne	Wasserrutschbahn
Wasserrutschbahnen	Wasserrutschbahn
WC	WC
West	West

Wurzelweg	Wurzelweg
Zuckerstück	Zuckerstück
Zugbahnhof	Zugbahnhof

4.4.2.2 Adjectives

Some inflected adjectives were not assigned to the correct adjective lemma. Other tweaks concern spelling variants.

Capitalised versions of these tokens were assigned to the same lemma, e.g. *Grosse* was also assigned to *groß*.

Table 4.18: German tokens manually labelled as adjectives and the lemma they were assigned to.

Token	Lemma
Beste	gut
coole	cool
doppelte	doppelt
erste	erst
früher	früh
gross	groß
grosse	groß
kuschliger	kuschlig
meiste	meist
meisten	meist
Mögliche	möglich
neues	neu
platschnass	platschnass
riesen	riesig
spannender	spannend
süsses	süß
ungewöhnliche	ungewöhnlich
weit	weit
Wilderem	wild
wohler	wohl

4.4.2.3 Adverbs

Capitalised versions of these tokens were assigned to the same lemma.

Table 4.19: German tokens manually labelled as adverbs and the lemma they were assigned to.

Token	Lemma
anschliessend	anschließend
drittens	drittens
Drittens	drittens
erstens	erstens
Erstens	erstens
mega	mega
ok	okay
OK	okay
okay	okay
spät	spät
usw.	usw
vorgestern	vorgestern
weg	weg
weit	weit
zweitens	zweitens
Zweitens	zweitens

4.4.2.4 Verbs

4.4.2.4.1 Infinitives Capitalised versions of these tokens were assigned to the same lemma.

Table 4.20: German tokens manually labelled as verb infinitives and the lemma they were assigned to.

Token	Lemma
abliegen	abliegen
ankommen	ankommen
fangen	fangen
fliegen	fliegen
Fliegen	fliegen

fortfliegen	fortfliegen
herumlaufen	herumlaufen
Herumlaufen	Herumlaufen
hinabfallen	hinabfallen
rumzufliegen	rumfliegen
runterlaufen	runterlaufen
Runterlaufen	runterlaufen
sitzen	sitzen
spielen	spielen
trinken	trinken
verstecken	verstecken

4.4.2.4.2 Finite verbs Capitalised versions of these tokens were assigned to the same lemma.

Table 4.21: German tokens manually labelled as finite verbs and the lemma they were assigned to.

Token	Lemma
bitte	bitten
Bitte	bitten
danke	danken
erbreche	erbrechen
fällt	fallen
gondelten	gondeln
herausschaue	herausschauen
hinfährst	hinfahren
hintun	hintun
hinunterschaut	hinunterschauen
mitfahre	mitfahren
picknickten	picknicken
rüberkomme	rüberkommen
runterfalle	runterfallen
runterstürzen	herunterstürzen
schwänze	schwänzen
weisst	wissen

4.4.2.4.3 Participles

Table 4.22: German tokens manually labelled as verb participles and the lemma they were assigned to.

Token	Lemma
abgestürtzt	abstürtzen
ausgeruscht	ausrutschen
gebraucht	brauchen
gedacht	denken
gefallen	gefallen
gehört	hören
gepicknickt	picknicken
geraten	geraten
gerodelt	rodeln
geschnätzelt	schätzeln
gestanden	stehen
getraut	trauen
getroffen	treffen
gewandert	wandern
heruntergeladen	herunterladen
raufgeklettert	raufklettern
reingefüllt	einfüllen
runtergerutscht	runterrutschen
runtergesprungen	runterspringen
umgeflogen	umfliegen

4.4.2.5 Pronouns

`TreeTagger/koRpus` assigns the 2SG pronouns *du* (nominative) and *dich* (accusative) to the lemma *du*, but *dir* (dative) to its own lemma. We consequently manually assigned all nominative, accusative, and dative forms of German personal pronouns to the nominative lemma. Moreover, `TreeTagger/koRpus` assigns the tokens *meine, meinen* to the verb *meinen*, but these tokens are usually forms of the pronoun *mein*. We therefore overrode this default behaviour and assigned all *meine, meinen* tokens to the pronoun *mein*.

Note that no distinction is made between *sie* (3SG.fem), *sie* (3PL) and *Sie* (formal second person). Further note that `TreeTagger/koRpus` assigns the pronoun *ihr* to the lemma *ihr* (2PL), though it may also be the 3SG.fem dative form.

These manual tweaks may introduce some tokens to be assigned to an incorrect lemma, e.g., when *meine, meinen* actually do belong to the verb lemma *meinen*, but on the whole they do more good than harm.

Table 4.23: German tokens manually labelled as pronouns and the lemma they were assigned to.

Token	Lemma
dir	du
Dir	du
ihn	er
ihm	er
uns	wir
euch	ihr
Sie	sie
sie	sie
ihnen	sie
Ihnen	sie
mein	mein
meine	mein
meinen	mein

4.4.2.6 Prepositions

Some German prepositions can be combined with an article. For instance, *zum* is the contraction of the preposition *zu* and the article *dem*. `TreeTagger/koRpus` recognises a couple of these contractions but needs some help with some others that occurred in the texts. Contracted prepositions + articles are assigned to the semantically more important preposition lemma, i.e., *übers* is assigned to *über*, not to *das*.

Additionally, some prepositions were not assigned to the correct lemma when they occurred at the start of a sentence (i.e., when they began with a capital letter). These manual assignments take care of this problem.

Table 4.24: German tokens manually labelled as prepositions and the lemma they were assigned to.

Token	Lemma
am	an

Am	an
aufm	auf
Aufm	auf
aufs	auf
Aufs	auf
im	in
Im	in
In	in
übers	über
Übers	über

4.4.2.7 Interjections

Table 4.25: German tokens manually labelled as interjections and the lemma they were assigned to.

Token	Lemma
ciao	ciao
Ciao	ciao
hi	hi
Hi	hi
tschüss	tschüss
Tschüss	tschüss

4.4.2.8 Abbreviations

This should be fixed – some of these words were labelled as adverbs.

Table 4.26: German tokens manually labelled as abbreviations and the lemma they were assigned to.

Token	Lemma
BFF	bff
ok	ok
Ok	ok
OK	ok
UFOS	ufo
usw	usw

Usw	usw
WCs	WC

4.4.2.9 Foreign words

Table 4.27: German tokens manually labelled as foreign words and the lemma they were assigned to.

Token	Lemma
Aéroport	Aéroport
Alien	Alien
Aliens	Alien
Bike	Bike
Bolinhos	Bolinhos
Bolognese	Bolognese
bye	bye
Centavos	Centavo
Check-In	check-in
Class	Class
End	End
first	first
First	first
gamen	gamen
gejumpt	jumpen
Goal	Goal
I	I
jumpen	jumpen
love	love
Merci	merci
mini	mini
Nero	Nero
Nuggets	Nugget
Rafting	Rafting
shoppen	shoppen
Skateboard	Skateboard
sliden	sliden
Snack	Snack

Surf	Surf
Wasabi	Wasabi
you	you

4.4.3 Portuguese

Compared to French and German, Portuguese boasts a wider range of contracted and inflected forms. Unfortunately, `TreeTagger`'s default lemmatisations of these forms render them unusable for the computation of lemmatised lexical sophistication measures.

4.4.3.1 Prepositions

The following forms are contractions of prepositions with articles or pronouns. They were assigned to an appropriate lemma and are considered to be prepositions.

Table 4.28: Portuguese tokens manually labelled as prepositions and the lemma they were assigned to.

Token	Lemma
à	ao
ao	ao
aos	ao
às	ao
àquela	àquele
àquelas	àquele
àquele	àquele
àqueles	àquele
aqueloutra	aquelotro
aqueloutras	aquelotro
aqueloutro	aquelotro
aqueloutros	aquelotro
cha	cho
chas	cho
cho	cho
chos	cho
daquela	daquele
daquelas	daquele
daquele	daquele
daqueles	daquele
dela	dela
delas	dela
deles	dela
dessa	desse

dessas	desse
desse	desse
desses	desse
desta	deste
destas	deste
deste	deste
destes	deste
da	do
das	do
do	do
dos	do
doutra	doutro
doutras	doutro
doutro	doutro
doutros	doutro
dum	dum
duma	dum
dumas	dum
duns	dum
essoutra	essoutro
essoutras	essoutro
essoutro	essoutro
essoutros	essoutro
estoutra	estoutro
estoutra	estoutro
estoutro	estoutro
estoutros	estoutro
naquela	naquele
naquelas	naquele
naquele	naquele
naqueles	naquele
nele	nele
neles	nele
nessa	nesse
nessas	nesse
nesse	nesse

nesses	nesse
nesta	neste
nestas	neste
neste	neste
nestes	neste
na	no
nas	no
no	no
nos	no
num	num
numa	num
numas	num
nuns	num
pela	pelo
pelas	pelo
pelo	pelo
pelos	pelo
pola	polo
polas	polo
polo	polo
polos	polo

4.4.3.2 Articles

The word *a* is typically a feminine definite article and is tagged correctly as such. Occasionally, however, it functions as a preposition, and `TreeTagger` correctly picks it up as such. Unfortunately, *a* isn't represented as a preposition in the SUBTLEX-PT corpus and is thus considered a rare lemma. To circumvent this, all instances of *a* are assigned to the article category.

Table 4.29: Portuguese tokens manually labelled as definite articles and the lemma they were assigned to.

Token	Lemma
a	o

4.4.3.3 Personal pronouns

The contracted forms *comigo*, *contigo* etc. ('with me', 'with you' etc.) were assigned to the personal pronoun category. The lemmata were identical to the base words.

Table 4.30: Portuguese tokens manually labelled as personal pronouns and the lemma they were assigned to.

Token	Lemma
comigo	comigo
connosco	connosco
consigo	consigo
contigo	contigo
convosco	convosco

4.4.3.4 Nouns

Table 4.31: Portuguese tokens manually labelled as nouns and the lemma they were assigned to.

Token	Lemma
beira_mar	beira-mar
bikini	bikini
bikinis	bikini
bodyboard	bodyboard
bowling	bowling
chalet	chalet
chalets	chalet
cor_de_laranja	cor-de-laranja
cor_de_rosa	cor-de-rosa
fato_de_banho	fato-de-banho
fatos_de_banho	fato-de-banho
fim_de_semana	fim-de-semana
fins_de_semana	fim-de-semana
guarda_sol	guarda-sol
hamster	hamster
hamsters	hamster
hip_hop	hip-hop
internet	internet

jacuzzi	jacuzzi
karaoke	karaoke
karaokes	karaoke
meia_noite	meia-noite
meio_dia	meio-dia
montanha_russa	montanha-russa
montanhas_russas	montanha-russa
pára_quedismo	pára-quedismo
pára_sol	para-sol
paraquedas	pára-quedas
pequeno_almoço	pequeno-almoço
pequenos_almoços	pequeno-almoço
ping_pong	ping-pong
pôr_do_sol	pôr-do-sol
pré_história	pré-história
segunda_feira	segunda-feira
segundas_feiras	segunda-feira
sexta_feira	sexta-feira
sextas_feiras	sextas-feiras
snowboard	snowboard
snowboards	snowboard
t_shirt	t-shirt
t_shirts	t-shirt
terça_feira	terça-feira
terças_feiras	terça-feira

4.4.3.5 Adjectives

The feminine form for 'thank you', *obrigada*, needs to be assigned manually to the masculine form.

Table 4.32: Portuguese tokens manually labelled as adjectives and the lemma they were assigned to.

Token	Lemma
obrigada	obrigado

4.4.3.6 Interjections

Table 4.33: Portuguese tokens manually labelled as interjections and the lemma they were assigned to.

Token	Lemma
hey	hey
hó	hó
ó	ó

4.4.3.7 Foreign words

Table 4.34: Portuguese tokens manually labelled as foreign words and the lemma they were assigned to.

Token	Lemma
alien	alien
bacon	bacon
dog	dog
et	et
frisbee	frisbee
h	h
hot	hot
i	i
ice-tea	ice-tea
iochaminansa	iochaminansa
is	is
love	love
name	name
o_k	o_k
ok	ok
p_s	p_s
raclette	raclette
ratatouille	ratatouille
snorkeling	snorkeling
spa	spa
tea	tea
tuc-tuc	tuc-tuc

vs	vs
what	what
your	your

Part III

Algorithmic measures of lexical richness

Chapter 5

Lexical diversity

Lexical diversity measures attempt to express the extent to which different words are used in the same text (Jarvis: variability). To compute such measures, we first tagged the texts (see Chapter 4). The result of this tagging is an `.Rda` object that contains a data frame that in turn contains one `kRp` object for each tagged text. To compute the lexical diversity measures, these `kRp` objects are read out of the data frame and a number of functions are applied to them. Most of these functions are part of the `koRpus` package.

When computing the lexical diversity measures, stopwords (Section 4.2) and foreign words were disregarded (`RFunctions/filterClasses.R`).

Some of the measures described in this chapter aren't strictly speaking lexical diversity measures, but since it was easiest to compute while computing the real lexical diversity measures, they're discussed here anyway.

5.1 Text length

`nTokens`: Number of tokens.

`nTypes`: Number of types, i.e., different tokens.

`nLemmas`: Number of different lemmata. For instance, French *sont* and *est* are different types (word forms) but belong to the same lemma (*être*).

`nFullStops`: Number of full stops (i.e., *.*, *?*, *!*). This serves as the number of orthographic sentences in the text.

5.2 Syntactic complexity

`conjunctionNumber`: The number of tokens belonging to the POS category conjunction.

`conjunctionRatio`: The ratio of conjunctions to the number of tokens.

`meanSentenceLength`: The number of tokens divided by the number of full stops. This isn't currently provided in the output, but it's easy to compute.

5.3 Lexical complexity

`meanWordLength`: The average (mean) length of the words as they occur in the text (i.e., not lemmatised) in letters.

`lexWordNumber`: The number of lexical tokens, i.e., tokens belonging to the POS categories adjective, adverb, noun and verb.

`lexWordRatio`: The ratio of lexical tokens to the number of tokens.

`ornWordNumber`: The number of 'ornamental' tokens, i.e., tokens belonging to the POS categories adjective and adverb.

`ornWordRatio`: The ratio of ornamental tokens to the number of tokens. (Not included in output but easily computed.)

`RepeatedConjunctions`: The number of times conjunctions were repeated, i.e., the total number of conjunctions minus the number of unique conjunction lemmata. (Following up on a suggestion that repeating conjunctions and adverbs in particular may negatively affect human ratings.)

`PropRepeatedConjunctions`: `RepeatedConjunctions` divided by `nTokens`.

`RepeatedAdverbs`: The total number of adverbs minus the number of unique adverb lemmata.

`PropRepeatedAdverbs`: `RepeatedAdverbs` divided by `nTokens`.

5.4 Measures based solely on the number of types, tokens, and lemmata

All of the following measures can directly be computed on the basis of the `nTokens`, `nTypes` and `nLemmas` variables.

`TTR`: The type–token ratio:

$$TTR = \frac{\text{number of types}}{\text{number of tokens}} \tag{5.1}$$

`Guiraud`: Guiraud's (1954) R:

$$R = \frac{\text{number of types}}{\sqrt{\text{number of tokens}}} \tag{5.2}$$

These two basic measures can be computed from `nTokens` and `nTypes`, but they are nonetheless provided in the default output for convenience.

`LTR`: The lemma–token ratio:

$$LTR = \frac{\text{number of lemmata}}{\text{number of tokens}} \tag{5.3}$$

`GuiraudLemma`: Guiraud's R using lemmata instead of types:

$$G_L = \frac{\text{number of lemmata}}{\sqrt{\text{number of tokens}}} \tag{5.4}$$

The script doesn't currently compute `LTR` and `GuiraudLemma`, but both measures can straightforwardly computed from the present output using `nLemmas` and `nTokens`.

`Herdan`: Herdan's C (see Tweedie & Baayen, 1998):

$$C = \frac{\log(\text{number of types})}{\log(\text{number of tokens})} \tag{5.5}$$

`Rubet`: Rubet's k (see Tweedie & Baayen, 1998):

$$k = \frac{\log(\text{number of types})}{\log\left(\log(\text{number of tokens})\right)} \tag{5.6}$$

`Carroll`: Carroll's Corrected TTR (see Tweedie & Baayen, 1998):

$$CTTR = \frac{\text{number of types}}{\sqrt{2 \times \text{number of tokens}}} \tag{5.7}$$

Note that `Carroll` is just `Guiraud`$^{-0.5}$.

`Dugast`: Dugast's Uber Index (see Tweedie & Baayen, 1998):

$$U = \frac{\log(\text{number of tokens}^2)}{\log(\text{number of tokens}) - \log(\text{number of types})} \tag{5.8}$$

Dugast's Uber Index can't be computed for texts with a `TTR` of 1 as the denominator equals zero for such texts. For those texts, we used the Dugast's Uber Index for the text with the highest `TTR` other than 1.

`Summer`: Summer's index (see Tweedie & Baayen, 1998):

$$S = \frac{\log\left(\log(\text{number of types})\right)}{\log\left(\log(\text{number of tokens})\right)} \tag{5.9}$$

`Maas`: One of Maas' indices (see Tweedie & Baayen, 1998):

$$a^2 = \frac{\log(\text{number of tokens}) - \log(\text{number of types})}{\log^2(\text{number of tokens})} \tag{5.10}$$

`LN`: Lukajenkov and Nesitoj's index (see Tweedie & Baayen, 1998):

$$LN = \frac{1 - (\text{number of types})^2}{(\text{number of types})^2 \times \log(\text{number of tokens})} \tag{5.11}$$

`Brunet`: Brunet's index (see Tweedie & Baayen, 1998):

$$W = (\text{number of tokens})^{(\text{number of types})^{-0.172}} \tag{5.12}$$

None of these measures are currently shown in the output, but they are easy to compute since they are functions of `nTokens` and `nTypes` only. Lemma-based versions of these measures are similarly straightforward to compute.

5.5 POS-specific word counts and ratios

Kim (2014) included word counts and ratios specific to the lexical parts of speech. The following predictors are provided in the output:

`nNounTokens`: The number of nouns.

`nVerbTokens`: The number of verbs.

`nAdjTokens`: The number of adjectives.

`nAdvTokens`: The number of adverbs.

`nNounTypes`: The number of unique nouns.

`nVerbTypes`: The number of unique verbs.

`nAdjTypes`: The number of unique adjectives.

`nAdvTypes`: The number of unique adverbs.

`nNounLemmas`: The number of unique noun lemmata.

`nVerbLemmas`: The number of unique verb lemmata.

`nAdjLemmas`: The number of unique adjective lemmata.

`nAdvLemmas`: The number of unique adverb lemmata.

On the basis of these predictors, it is easy to compute the following or any of a number of other variations:

`TTR.Noun`: `nNounTypes` divided by `nNounTokens`.

`TTR.Verb`: `nVerbTypes` divided by `nVerbTokens`.

`TTR.Adj`: `nAdjTypes` divided by `nAdjTokens`.

`TTR.Adv`: `nAdvTypes` divided by `nAdvTokens`.

Additionally:

`conjunctionNumber`: The number of conjunctions.

`conjunctionRatio`: `conjunctionNumber` divided by `nTokens`.

`nConjunctionLemmas`: The number of unique conjunction lemmata.

5.6 Mean segmental type–token ratio (MSTTR)

For the MSTTR (Johnson, 1944), the text is split into equal-sized consecutive segments whose size is defined by the user. The type–token ratio (TTR, see above) is computed for each segment, and the mean TTR is the text's MSTTR. Tokens at the end of a text that don't form a full segment are ignored.

By default, the text is split into consecutive segments of 100 tokens. Given that most of our texts are considerably shorter than that, it does not make much sense to stick to this default. For now, three MSTTR values are computed for each text, namely with segment lengths of 10, 30 and 50 tokens.

`MSTTR10`: MSTTR with a segment length of 10 tokens.

`MSTTR30`: MSTTR with a segment length of 30 tokens.

`MSTTR50`: MSTTR with a segment length of 50 tokens.

The numbers 10, 30 and 50 were chosen arbitrarily, but can be fine-tuned during the exploratory phase of the analysis (see Section 10.2): If any of these measures show promise as predictors of human judgements of lexical richness, we can try to optimise the segment length (e.g., segment length of 43 instead of a round number).

When a text has fewer than 10, 30 or 50 tokens, the respective MSTTR value is currently set to `NaN`. For modelling purposes, the text's `TTR` is used as a replacement value in such cases.

5.7 HD-D

From `?koRpus::lex.div`:

> The HD-D value can be interpreted as the idealized version of vocd-D (see McCarthy & Jarvis (2007)). For each type, the probability is computed (using the hypergeometric distribution) of drawing it at least one time when drawing randomly a certain number of tokens from the text – 42 by default. The sum of these probabilities make up the HD-D value.

The number of words drawn randomly from the text (`rand.sample`) can be specified by the user. In addition to the default (42), we include 31 and 20; these numbers were chosen arbitrarily. If any of these measures show promise as predictors of human judgements of lexical richness, we can try to optimise this setting (e.g., 27 instead of either 20 or 31).

If the number of *tokens* is less than the `rand.sample` setting, the respective HD-D measure is equal to the number of *types*.

`HDD42`: HD-D based on random sets with 42 tokens.

`HDD31`: HD-D based on random sets with 31 tokens.

`HDD20`: HD-D based on random sets with 20 tokens.

5.8 Moving-average type–token ratio (MATTR)

From `?koRpus::lex.div`:

> The Moving-Average Type-Token Ratio (Covington & McFall, 2010) calculates TTRs for a defined number of tokens (called the "window"), starting at the beginning of the text and moving this window over the text, until the last token is reached. The mean of these TTRs is the MATTR.

The default window size is 100, which exceeds the length of most of our texts. We therefore also included MATTRs for shorter window sizes, which were chosen arbitrarily and can be tweaked if MATTR shows promise as a predictor of human judgements.

`MATTR100`: MATTR for a window size of 100.

`MATTR65`: MATTR for a window size of 65.

`MATTR30`: MATTR for a window size of 30.

`MATTR10`: MATTR for a window size of 10.

Currently, if the window size exceeds the length of the text, the respective MATTR is set to `NaN`. For modelling purposes, the text's `TTR` is used as a replacement value in such cases.

5.9 Measure of textual lexical diversity (MTLD)

From `?koRpus::lex.div`:

> For the Measure of Textual Lexical Diversity (McCarthy & Jarvis, 2010) so called factors are counted. Each factor is a subsequent stream of tokens which ends (and is then counted as a full factor) when the TTR value falls below the given factor size. The value of remaining partial factors is estimated by the ratio of their current TTR to the factor size threshold. The MTLD is the total number of tokens divided by the number of factors. The procedure is done twice, both forward and backward for all tokens, and the mean of both calculations is the final MTLD result.

The default TTR factor is 0.72; we also computed MTLD measures for TTR factors of 0.83, 0.61, and 0.50. These numbers were chosen arbitrarily and can be tweaked if the MTLD measures show promise as predictors of human judgements.

`MTLD83`: MTLD for a TTR factor of 0.83.

`MTLD72`: MTLD for a TTR factor of 0.72.

`MTLD61`: MTLD for a TTR factor of 0.61.

`MTLD50`: MTLD for a TTR factor of 0.50.

The MTLD measures can't be estimated for (usually short) texts with a `TTR` of 1. Currently, the MTLD measures for these texts are set to `NA`. For modelling purposes, the corresponding MTLD measure of the text in the training sample with the highest non-1 `TTR` value is used as a replacement value in such cases. The justification for this is quite simply that it is difficult to conceive of any other reasonable replacement value. That said, people reanalysing these data can set their preferred replacement value in the `predict_ratings_*.R` scripts.

5.10 Moving-average measure of textual lexical diversity (MA-MTLD)

From `?korPus::lex.div`:

> The Moving-Average Measure of Textual Lexical Diversity (Jarvis, no year) combines factor counting and a moving window similar to MATTR: After each full factor the next one is calculated from one token after the last starting point. This is repeated until the end of text is reached for the first time. The average of all full factor lengths is the final MTLD-MA result. Factors below the min.tokens threshold are dropped.

The default `min.tokens` threshold is 9. The default TTR factor is 0.72; we also computed MTLD-MA measures for TTR factors of 0.83, 0.61, and 0.50. These numbers were chosen arbitrarily and can be tweaked if the MTLD-MA measures show promise as predictors of human judgements.

`MTLD83MA`: MTLD-MA for a TTR factor of 0.83.

`MTLD72MA`: MTLD-MA for a TTR factor of 0.72.

`MTLD61MA`: MTLD-MA for a TTR factor of 0.61.

`MTLD50MA`: MTLD-MA for a TTR factor of 0.50.

The MTLD-MA measures can't be estimated for many texts, even if their `TTR` isn't 1. Currently, the MTLD-MA measures for these texts are set to `NA`, and the MTLD-MA measures are not used for modelling purposes.

5.11 Yule's K

`YulesK`: Yule's K is a function of the total number of tokens and the frequency with which each type occurs in the text; see Tweedie & Baayen (1998):

$$K = 10^4 \times \frac{(\sum fX \times X^2) - \text{number of tokens}}{\text{number of tokens}^2} \tag{5.13}$$

"where X is a vector with the frequencies of each type, and fX is the frequencies for each X.''

By way of illustration, consider the following text:

> "When justifying their use of standardised effect sizes, researchers usually cite the need to be able to compare results that were obtained on different scales or to render results on scales that are difficult to understand more meaningful. I understand this argument up to a point, but I think it is overused. Firstly, to the extent that different outcome measures for similar constructs are commonly used, it should be possible to rescale them without relying on the variance of the sample at hand. This could be done by making reference to norming studies. Moreover, standardised effect sizes should not be an excuse to ignore one's measurements."

This text contains 106 tokens and 75 types. 57 types occur just once, 13 types occur twice, 2 types occur 3 times, another 2 occur 4 types, and 1 type (*to*) occurs 9 times:

Type frequency	Number of types
1	57
2	13
3	2
4	2
9	1

Using these numbers, Yule's K can now be computed as

$$K = 10^4 \times \frac{(57 \times 1^2 + 13 \times 2^2 + 2 \times 3^2 + 2 \times 4^2 + 1 \times 9^2) - 106}{106^2} \approx 119.3$$

Higher values indicate less lexical diversity.

`YulesKLemma`: Same as `YulesK`, but the in-text frequencies per lemma were used instead of the in-text frequencies per type.

Chapter 6

Lexical sophistication

The lexical diversity measures discussed in Chapter 5 involve expressing the variation of words used in the text. Lexical sophistication, by contrast, involves expressing how rare, sophisticated or specific the words used in the text are (Jarvis: rarity). This is accomplished by looking up the words (tokens or lemmata) that occur in the text in an independent frequency corpora and summarising this frequency information.

6.1 Token- and lemma-based versions

The lexical sophistication measures were computed with respect to both the *tokens* and the *lemmata* in the texts. Consider, for instance, for a (made-up) German sentence like *Die Mäuse fürchteten sich vor den Katzen.* When computing the token-based versions of the sophistication measures, we looked up how frequently the words *die*, *Mäuse*, *fürchteten*, *sich*, *vor*, *den* and *Katzen* were in the German frequency corpus. When computing the lemma-based versions, we first lemmatised the frequency corpus—i.e., we aggregated the frequency counts of, say, *Maus*, *Mäuse* and *Mäusen*—and then looked up how frequently the lemmata *Maus*, *fürchten*, *sich*, *vor*, *Katze* as well as the definite article occurred in the lemmatised frequency corpus.

The lemma-based sophistication measures are prefixed by the string `Lemma`.

6.2 Frequency corpora

The corpora we used for computing lexical sophistication measures of the children's texts are Lexique 3 for French (New et al., 2007), SUBTLEX-DE for German (Brysbaert et al., 2011), and SUBTLEX-PT for Portuguese (Soares et al., 2015). These frequency corpora are

highly comparable in that they are all based on television and film subtitles. A further advantage is that they are freely available online.

6.2.1 Lemmatisation of SUBTLEX-DE and SUBTLEX-PT

Unlike Lexique3, SUBTLEX-DE and SUBTLEX-PT do not come with a column containing lemma frequencies. Lemmatising SUBTLEX-DE and SUBTLEX-PT isn't ideal as the tokens obviously lack context and the corpora don't distinguish between homographs. German *ihr*, for instance, can equally refer to the fem. 3rd p. sing. possessive pronoun, the 3rd p. pl. possessive pronoun, the fem. 3rd p. sing. dative personal pronoun, and the 2nd p. pl. nominative personal pronoun. Nonetheless, even a rough lemmatisation may be useful as it gets rid of distictions between common grammatical variants such as verb tenses.

The lemmatisation of the SUBTLEX corpora operates on the same principles as the lemmatisation of the German and Portuguese texts (see Chapter 4). For German, for instance, the word forms *sie, Sie, ihnen, Ihnen* were were assigned to the lemma *sie*.

Note that Lexique 3 already contains a column containing lemma frequencies. It does not make much sense to lemmatise Lexique 3 again for the sake of greater comparability with SUBTLEX-DE/PT since by re-lemmatising the context-free tokens in Lexique 3, we would probably end up with considerably more errors than currently. Lemmatising SUBTLEX-DE/PT is not ideal, but it is necessary; lemmatising Lexique 3 would be neither ideal nor necessary.

6.2.2 Changes to the frequency corpora

To compute the lexical sophistication measures, the words and lemmata in the texts need to be aligned with the words and lemmata in the frequency corpora. This requires a handful of tweaks to the corpora. These changes are effected in the `RFunctions/LexicalSophistication*.R` scripts.

6.2.2.1 French (Lexique 3)

1. The ligature *œ* was replaced by *oe*.
2. Multiword items were combined. For instance, *tandis que* was rewritten as *tandis_que*. See Section 3.4.2.
3. `koRpus` uses the spelling *plaître*; Lexique *plaire*. Instances of *plaire* in Lexique 3 were replaced by *plaître*.

4. While Lexique 3 makes some fine-grained distinctions between homographs (e.g., copula vs. full verb instances of *être*), it is difficult to combine this with the information from TreeTagger/koRpus. To simplify matters, we aggregated frequency counts over homographs in Lexique 3. That is to say, the total frequency count of the lemma *être* includes the frequency count of the copula, the full verb, the auxiliary verb as well as the noun. Similarly, the total frequency count of the token *est* include the frequency count of the verbs and of the wind direction.

6.2.2.2 German (SUBTLEX-DE)

1. SUBTLEX-DE does not distinguish between homographs nor between capitalised and uncapitalised words (e.g., *sie* vs. *Sie*). We therefore converted SUBTLEX-DE to lowercase.

2. Swiss written German does not use the German character *ß* but used *ss* instead. Wherever *ß* occurred in SUBTLEX-DE, it was replaced by *ss*.

3. Token-based version only: Some abbreviations were written without dots in SUBTLEX-DE but with dots in the texts. The dots were added to SUBTLEX-DE. (Abbreviations concerned: *etc.*, *usw.*, *p.s.*.) This step is unnecessary for the lemma-based version since lemmatising the texts and SUBTLEX-DE takes care of this.

4. Lemma-based version only: Words with `<unknown>` lemmata were removed from the frequency corpus, i.e., they were not considered when computing frequency ranks.

5. Lemma-based version only: Several words in the SUBTLEX-DE were ambiguous in terms of the lemma they represent. For such words, `koRpus` lists all possible lemmata corresponding to the word.[1] To make an alignment between the texts and the corpus possible, we always selected the first of these double lemmata as the target lemma since closer inspection of the cases concerned suggested this made most sense.

6.2.2.3 Portuguese (SUBTLEX-PT)

1. Lemma-based version only: The tagging software experiences difficulties with some hyphenated Portuguese words: it splits them up into different words, which messes up the alignment of words to lemmata. Most of the words concerned are rare (fewer than 1 occurrences per 1,000,000 words), and the problem could largely be solved by removing hyphenated words with a frequency of < 1 p.m. prior to lemmatising the SUBTLEX-PT corpus. One additional word that caused difficulties (*dia-a-dia*) was removed as well.

[1] `koRpus` doesn't output double lemmata for Portuguese.

2. Lemma-based version only: Words with `<unknown>` lemmata were removed from the frequency corpus, i.e., they were not considered when computing frequency ranks.

3. Token-based version only: Some of the words occurring in the texts were originally cliticised, e.g., *encontramo* from *encontramo-nos*. In SUBTLEX-PT, occurrences of such words were added to the frequency count of the full words, i.e., *encontramo* was added to *encontramos*. To take this into account and thereby avoid considering words like *encontramo* to be so rare that they don't occur in the frequency corpus, entries with these originally cliticised words were added to SUBTLEX-PT. These entries were given the same corpus frequency and frequency rank as the corresponding full words, and they were added to the corpus after computing the frequency ranks so that their addition did not affect the frequency ranks of other words. The words concerned are the following:

Table 6.1: Entries added to SUBTLEX-PT.

Existing entry	Added entry
assar	assá
chateamos	chateamo
divertimos	divertimo
encontramos	encontramo
escalar	escalá
explorar	explorá
habituamos	habituamo
levantamos	levantamo
levar	levá
molhamos	molhamo
passar	passá
podemos	podemo
podíamos	podíamo
puxar	puxá
refrescamos	refrescamo
vamos	vamo
vender	vendê

6.3 Extracting frequency information from the texts

Chapter 4 describes how the texts were tagged. The result of this tagging is an `.Rda` object that contains a data frame that in turn contains one `kRp` object for each tagged text. To compute the lexical sophistication measures, these `kRp` objects are read out of the data frame and a number of functions were applied to them (see `RFunctions/LexicalSophistication*.R`).

1. For each text, the tokens or lemmata are returned, separated by spaces. This leaves out punctuation, stopwords (Section 4.2), foreign words, and cardinal numbers. If, for the lemma-based versions, a token could not unambigiously be assigned to a lemma, `koRpus` lists all possible lemmata. Since we can only deal with one lemma per token, only the first alternative proposed by `koRpus`, which seemed to be the correct lemma in most cases, was returned.

2. Each text was then rendered as a data frame (one row per word) and frequency information from the frequency corpora was added to each entry in this data frame.

6.4 Lexical sophistication measures

6.4.1 Frequency bands

This type of sophistication measure answers questions such as *Which proportion of the text consists of the top-100 most frequent words in the frequency corpus?* and *Which proportion of the text consists of words that don't belong to the top-5000 most frequent words in the frequency corpus?*. The cut-offs are set arbitrarily and can be fine-tuned during the exploratory phase of the analysis.

Currently, the `RFunctions/LexicalSophistiation*.R` scripts (specifically the `frequencyBands.R` subroutine) calculate the proportion of *tokens* in each text within the following frequency rank brackets:

`Freq100` / `LemmaFreq100`: 1–100 most frequent words in the reference corpus. For the unlemmatised version, *words* refers to word forms; for the lemmatised version, *words* refers to lemmata.

`Freq200` / `Freq200`: 101–200 most frequent words;

`Freq300` / `Freq300`: 201–300;

`Freq400` / `LemmaFreq400`: 301–400;

`Freq500` / `LemmaFreq500`: 401–500;

`Freq600` / `LemmaFreq600`: 501–600;

`Freq700` / `LemmaFreq700`: 601–700;

`Freq800` / `LemmaFreq800`: 701–800;

`Freq900` / `LemmaFreq900`: 801–900;

`Freq1000` / `LemmaFreq1000`: 901–1000;

`Freq2000` / `LemmaFreq2000`:1001–2000;

`Freq5000` / `LemmaFreq5000`: 2001–5000;

`FreqInf` / `LemmaFreqInf`: the rest. This does *not* include words that do not occur in the frequency corpus.

The cut-offs can be customised in the `MasterScript*.R` files, in which case the variable names are changed automatically. On the basis of these variables, broader frequency bands can be computed, e.g., `TopFreq500` (1–500 most frequent words), `TopFreq1000` (1–1000) and `TopFreq2000`.

The `otherStats.R` subroutine adds the following variables:

`notInSUBTLEX` / `LemmanotInSUBTLEX`: The proportion of words (tokens or lemmata) in the text that don't occur in the respective frequency corpus.

`wordsNotInSUBTLEX` / `LemmaWordsNotInSUBTLEX`: A character string with the words in the text that don't occur in the respective frequency corpus. This is useful for debugging as these words often turn out to contain spelling errors.

6.4.2 Average corpus frequency

The `otherStats.R` subroutine in the `RFunctions/LexicalSophistication*.R` scripts computes the following variables.

`meanSUBTLEX` / `LemmaMeanSUBTLEX`: The mean frequency (per 1,000,000 tokens) according to the frequency corpus (F) of the words (tokens, lemmata) in the text:

$$\text{meanSUBTLEX} = \frac{1}{n_{\text{tokens}}} \sum_{i=1}^{n_{\text{tokens}}} F_i \tag{6.1}$$

`meanSUBTLEXUnique` / `LemmaMeanSUBTLEXUnique`: For `meanSUBTLEX`, words are counted more than once if they occur more than once in the text. For `meanSUBTLEXUnique`, each word is counted only once, regardless of whether it occurs once or seven times. In other

words, this measure is based on the token or lemma frequency for the *types* occurring in the texts.

$$\text{meanSUBTLEXUnique} = \frac{1}{n_{\text{types}}} \sum_{i=1}^{n_{\text{types}}} F_i \tag{6.2}$$

When computing this measure, Kyle & Crossley (2015) did not consider words occurring in the texts that didn't occur in the frequency corpus, but they did not discuss their rationale for doing so. In contrast to these researchers, we did consider words occurring in the texts but not in the frequency corpus (i.e., we assigned a corpus frequency of 0 to these words).

`meanZipf` / `LemmameanZipf`: van Heuven et al. (2014) proposed the Zipf scale as a frequency scale that captures language users perception of the relative frequency of words. It is, in essence, a logarithmic transformation of the words' frequencies with a correction for words unobserved in the frequency corpus (see Brysbaert & Diependaele, 2013):

$$\text{Zipf} = \log_{10}\left(\frac{F_{\text{count}} + 1}{\frac{C_{\text{tokens}}}{10^6} + \frac{C_{\text{types}}}{10^6}}\right) + 3, \tag{6.3}$$

where F_{count} is the raw frequency count of the word in question (not the frequency per 1,000,000 words), C_{tokens} the size of the corpus in tokens, and C_{types} the size of the corpus in types (or lemmata, for the lemma-based version). For words not occurring in the corpus, Zipf scores were computed by simply using $F_{\text{count}} = 0$. The mean Zipf value per text was then computed.

We did not include the median Zipf value since it is a direct function of `medianSUBTLEX` described below.

`meanZipfUnique` / `LemmameanZipfUnique`: Like `meanZipf` / `LemmameanZipf` but each type in the text was only counted once.

`medianSUBTLEX` / `LemmaMedianSUBTLEX`: The median frequency (per 1,000,000 words) according to the frequency corpus of the words (tokens, lemmata) in the text.

`medianSUBTLEXUnique` / `LemmaMedianSUBTLEXUnique`: The median frequency (per 1,000,000 words) according to the frequency corpus of the words (tokens, lemmata) in the text, counting each type in the text only once.

If a word occurring in the text does not occur in the frequency corpus, it was replaced by the Zipf value corresponding to a word with 0 occurrences (i.e., with $F_{\text{count}} = 0$ in the equation above).

A handful of measures concern the 25th (`Q25`) and 75th (`Q75`) percentile of the frequency distribution rather than the central tendency:

`Q25SUBTLEX` / `LemmaQ25SUBTLEX`

`Q75SUBTLEX` / `LemmaQ75SUBTLEX`

`Q25SUBTLEXUnique` / `LemmaQ25SUBTLEXUnique`

`Q75SUBTLEXUnique` / `LemmaQ75SUBTLEXUnique`

Higher values on the variables in this subsection reflect the use of more common words and should hence be reflective of *less* lexical sophistication.

6.4.3 Average corpus frequency rank

The most common word in the frequency corpus has a frequency rank of 1; the second of 2 etc. Ties can result in fractioned ranks (e.g., frequency rank 5006.5 in case words 5006 and 5007 are tied).

`meanFreqRank` / `LemmaMeanFreqRank`: The mean frequency rank according to the frequency corpus of the words (tokens, lemmata) (R_i) in the text.

$$\text{meanFreqRank} = \frac{1}{n_{\text{tokens}}} \sum_{i=1}^{n_{\text{tokens}}} R_i \tag{6.4}$$

`meanFreqRankUnique` / `LemmaFreqRankUnique`: For `meanFreqRank`, words are counted more than once if they occur more than once in the text. For `meanFreqRankUnique`, each word is counted only once, regardless of whether it occurs once or seven times. In other words, this measure is based on the token or lemma frequency for the *types* occurring in the texts.

$$\text{meanFreqRankUnique} = \frac{1}{n_{\text{types}}} \sum_{i=1}^{n_{\text{types}}} R_i \tag{6.5}$$

If a word occurring in the text does not occur in the frequency corpus, its frequency rank is the maximum frequency rank in the frequency corpus.

Higher values on the variables in this subsection reflect the use of less common words and should hence be reflective of *more* lexical sophistication.

The CSV output also contains columns containing median, 25th percentile and 75th percentile frequency ranks, but these are necessarily highly correlated with the median, 25th and 75th percentile frequencies themselves so they are not analysed.

6.4.4 Miscellaneous frequency measures

`SUBTLEXrarestWord` / `LemmaSUBTLEXrarestWord`: The frequency per 1,000,000 words of the rarest type/lemma occurring in the text.

`NoFreqInf` / `NoLemmaFreqInf`: The number, rather than the proportion, of words not belonging to the 5,000 most frequent types/lemmata in the language, i.e., `FreqInf` (or `LemmaFreqInf`) times `nTokens`.

6.5 Advanced TTR and advanced Guiraud

Daller et al. (2003) introduced two metrics that combine diversity with sophistication information. The advanced TTR and advanced Guiraud are compute like the TTR and Guiraud indices, but only 'advanced' types are included in the numerator (all tokens are included in the denominator). Here, three variants were computed for each measure:

`AdvancedTTR500` and `AdvancedGuiraud500`: 'Advanced' types are defined as types not among the 500 most frequent in the frequency list.

`AdvancedTTR1000` and `AdvancedGuiraud1000`: 'Advanced' types are defined as types not among the 1000 most frequent in the frequency list.

`AdvancedTTR2000` and `AdvancedGuiraud2000`: 'Advanced' types are defined as types not among the 2000 most frequent in the frequency list.

Lemma-based versions were also computed.

Chapter 7

Evenness, disparity and dispersion

The measures discussed in this chapter are attempts to operationalise what Jarvis (2013a,b, 2017) calls evenness, disparity, and dispersion.

7.1 Evenness

Following Jarvis (2013b), evenness indices were computed by counting the number of tokens per type/lemma and calculating their standard deviation. Higher values reflect less evenness.

`evenness_type`: Standard deviation of the number of tokens per type.

`evenness_lemma`: Standard deviation of the number of tokens per lemma.

7.2 Disparity

No indices of semantic disparity were computed.

Formal disparity was operationalised by computing string-edit distances between each type or unique lemma in the text and every other type or unique lemma in the text and then taking the mean of all distances. The string-edit distances were computed using the Levenshtein (1966) algorithm, which computes the minimum number of operations (insertions, deletions, substitutions) required to transform one string into another. This operation cost was length-normalised by dividing it by the length of the lowest-cost alignment. Higher Levenshtein distances reflect less overlap between two strings, so higher mean Levenshtein distances indicate more formal disparity in the text.

1	2	3	4	5	6	7	
		a	c	t	o	r	
r	e	a	c	t		s	
I	I				D	S	= 4

Figure 7.1: The string *actor* can be transformed into the string *reacts* using a minimum of four operations (two insertions (I), one deletion (D) and one substitution (S)), so the (raw) Levenshtein distance between these strings equals four. To achieve this transformation, an alignment of 7 slots is needed. The length-normalised Levenshtein distance is therefore 4/7 = 0.57. Occasionally, aligments of different lengths yield the same raw Levenshtein distance. In such cases, the longest alignment that still yields the lowest raw Levenshtein distance is used for normalisation (see Heeringa, 2004, p. 131).

Figure 7.1 shows an example of a Levenshtein distance calculation. We can perform such a computation between each pair of types occurring in a text and then take the mean value.

(The values in the output are actually the complement of this value, i.e., 1 - the value, but they were reversed during the analysis.)

`disparity_type`: Formal disparity between the types as they occur in the text.

`disparity_lemma`: Formal disparity between the unique lemmata in the text.

7.3 Dispersion

Following Jarvis (p.c., August 2, 2017), we computed the dispersion index as follows. For the ith word in the text ($i \in 1, 2, \ldots, n$), we looked up how often it occurred in the next k words (i.e., words $i + 1$ through $i + k$). For all n words except those in the top-5 most frequent in the language's frequency list, the number of 'close repeats' was summed and then divided by the total number of words. Different dispersion indices were computed by (a) basing the calculation on either the types or lemmata and (b) setting k to 10, 20 and 30 (6 measures in total). Higher values reflect more 'clustericity' and hence less dispersion.

`dispersion_type_10`: Dispersion using the words as they occur in the text and $k = 10$.

`dispersion_type_20`: Dispersion using the words as they occur in the text and $k = 20$.

`dispersion_type_30`: Dispersion using the words as they occur in the text and $k = 30$.

`dispersion_lemma_10`: Dispersion using the lemmata instead of the words as they occur in the text and $k = 10$.

`dispersion_lemma_20`: Dispersion using the lemmata instead of the words as they occur in the text and $k = 20$.

`dispersion_lemma_30`: Dispersion using the lemmata instead of the words as they occur in the text and $k = 30$.

Part IV

Human judgements of lexical richness

Chapter 8

Collecting human ratings

We asked uninstructed native speaker judges to rate the lexical richness of the children's texts on a 1–9 scale. The texts rated were grammatically and orthographically corrected to lessen the effect of non-lexical features on the ratings, as described in Chapter 3. The texts were rated in 20 batches of 50–52 texts; raters only rated one batch each.

8.1 Raters

Raters were not paid for their participation, but could optionally leave behind their e-mail address to receive a summary of the project's results.

8.1.1 French

289 people rated at least one French text. A substantial number of these already quit after having rated a handful of texts, however, and only 183 people rated at least 50 out of 52 texts (including the two training texts).[1] Only these raters were retained. 37 further raters were excluded because they did not consider themselves native speakers of French, leaving 146 raters that were considered in the analyses. A visual inspection of the raters' responses throughout the rating session did not reveal strong satisficing patterns (e.g., constantly providing the same rating or predictably alternating between two or three ratings).

Most of these 146 raters hailed from Switzerland (83), followed by France (50), Belgium (4), Canada and Italy (2), and Germany, Australia, the USA, Norway, and the UK (1 each). Of the 21 raters who considered themselves to be bilingual in French and another language,

[1] Two raters provided more than 52 ratings, presumably because they logged in to the rating platform twice; only the ratings from their first login were retained.

7 considered German or Swiss-German their other native language, 6 Italian, 3 Teochew (a Chinese dialect), 2 English or Spanish, and 1 Portuguese.

8.1.2 German

556 people rated at least one German text. A substantial number of these already quit after having rated a handful of texts, however, and only 371 people rated at least 50 out of 53 texts (including the two training texts). Only these raters were retained. 41 further raters were excluded because they did not consider themselves native speakers of German or did not specify their native language, leaving 330 raters. 3 of those didn't specify their sex or age and were also removed from the analyses. Another 3 participants were (or claimed to be) younger than 16 years and were also excluded from the analyses, leaving 324 participants. Two of the remaining participants showed strong satisficing patterns (Figure 8.1) and were excluded, leaving 322 participants for the analyses.

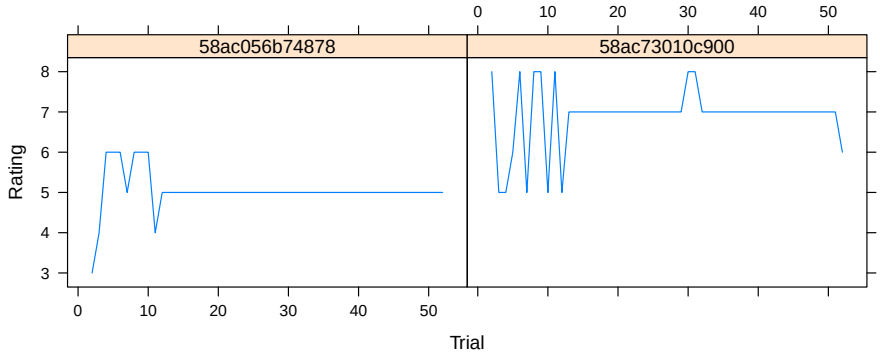

Figure 8.1: Two raters for the German texts revealed strong satisficing patterns and were discarded from the analyses.

Most of these 322 raters hailed from Switzerland (254), followed by Germany (46), Austria and Italy (4 each), Liechtenstein (2), and sundry other countries (1 each). Of the 44 raters who considered themselves to be bilingual in German and another language, 12 considered Italian their other native language, 7 French, 4 English, and the others a host of other languages.

8.1.3 Portuguese

212 people rated at least one Portuguese text. A substantial number of these already quit after having rated a handful of texts, however, and only 119 people rated at least 50 out of 54 texts (including the two training texts). Only these raters were retained.

Table 8.1: Description of raters. It's unclear whether the oldest French rater was indeed 95 years old or whether they meant that they were born in 1995.

	French	German	Portuguese
Number of raters	146	322	106
Percentage men	21	17	29
Median age (years)	27	25	35
Minimum age (years)	18	19	18
Maximum age (years)	95	76	75
Percentage bilinguals	14	14	13
Percentage linguists	10	8	10
Percentage teachers	16	26	14
Percentage students	47	57	21

11 further raters were excluded because they did not consider themselves native speakers of Portuguese or did not specify their native language, leaving 108 raters. Two of the remaining participants showed strong satisficing patterns (Figure 8.2) and were excluded, leaving 106 participants for the analyses.

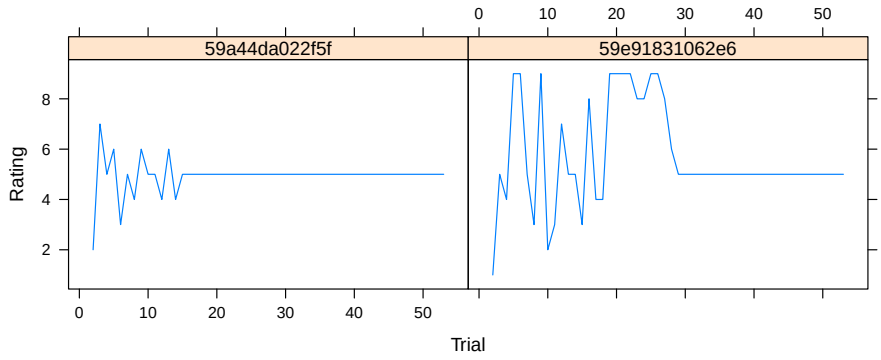

Figure 8.2: Two raters for the Portuguese texts revealed strong satisficing patterns and were discarded from the analyses.

About half (48) of the raters hailed from Portugal, followed by Switzerland (38), Brazil (9), Germany (5), Spain (2) and Mozambique (1). Three raters did not specify their country. Of the 14 raters who considered themselves to be bi- or multilingual in Portuguese and another language, 8 considered French their other native language, 2 Spanish, and one each for a couple of other languages.

8.2 Texts and text sets

The texts to be rated were orthographically and grammatically corrected versions of the children's written productions as documented in Chapter 3.

To lessen the chances that exceedingly short texts affected the raters' baseline for the lexical richness of the other texts, texts with fewer than 45 (French) or 50 (German and Portuguese) *letters* (not including spaces and punctuation) were not presented for rating. The French cut-off was slightly less stringent than the German and Portuguese ones in order to retain 1,000 texts (20 sets × 50 texts per set, see below) for rating. Each text presented for rating included at most one string that was labelled as illegible.

Rather than have all raters rate all texts for a given language, the texts for each language were split up into 20 equal-sized sets that were compiled so as to be maximally similar in terms of the number of texts written by children from the different regions (German-speaking Switzerland, French-speaking Switzerland, and Portugal), the number of argumentative vs. narrative texts, and the time at which the texts were produced (script: `construct_sets.R`). For both French and German, the distribution of these variables was highly similar but not identical in the different sets; for Portuguese, each set contained the same number of, for instance, narrative texts written at Time 3 by children in French-speaking Switzerland. See Figures 8.3–8.5.

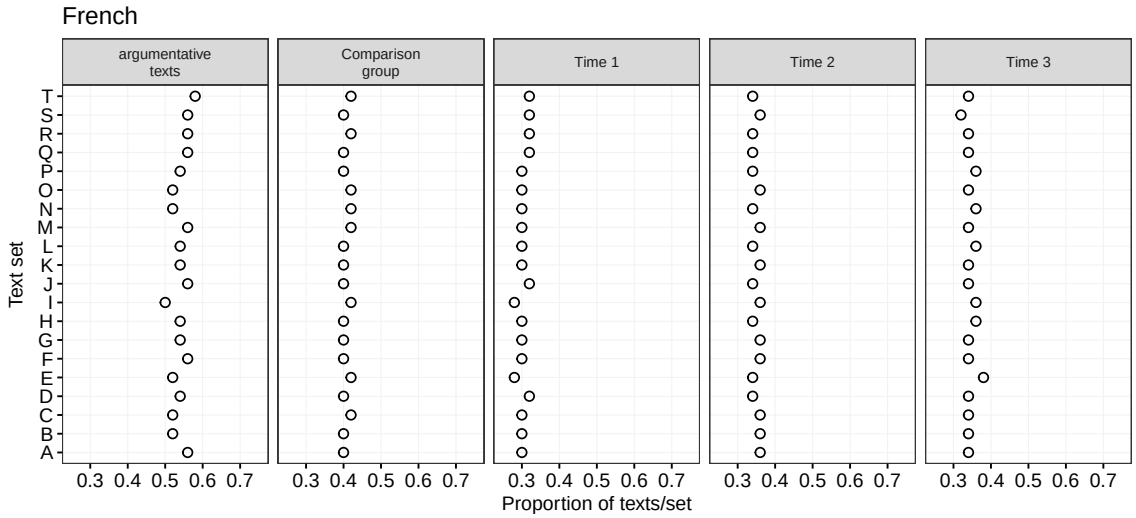

Figure 8.3: Variation between the French text sets in terms of the children's background, genre, and time of data collection.

For French and German, a limited number of texts (the number of suitable texts *modulo*

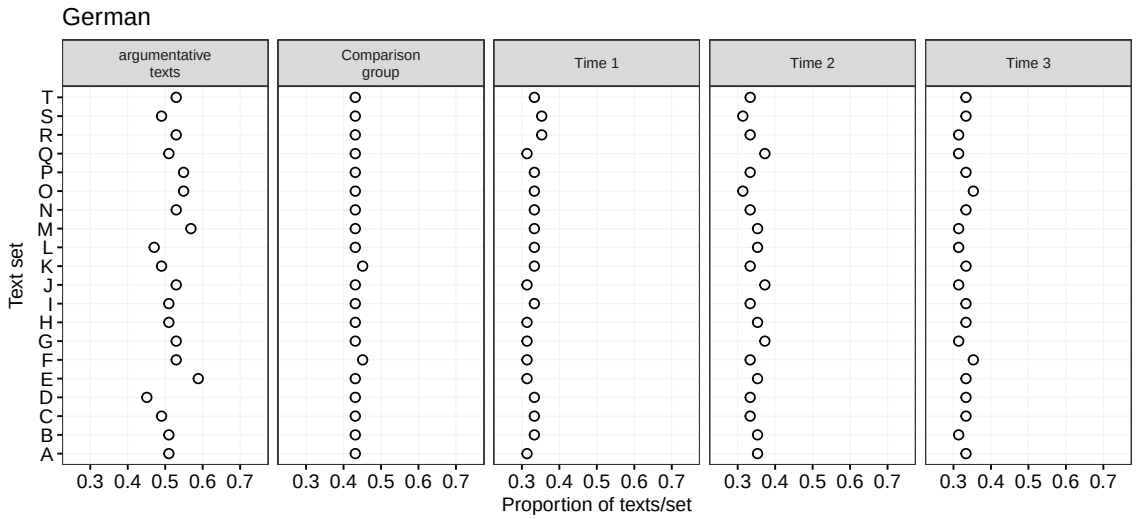

Figure 8.4: Variation between the German text sets in terms of the children's background, genre, and time of data collection.

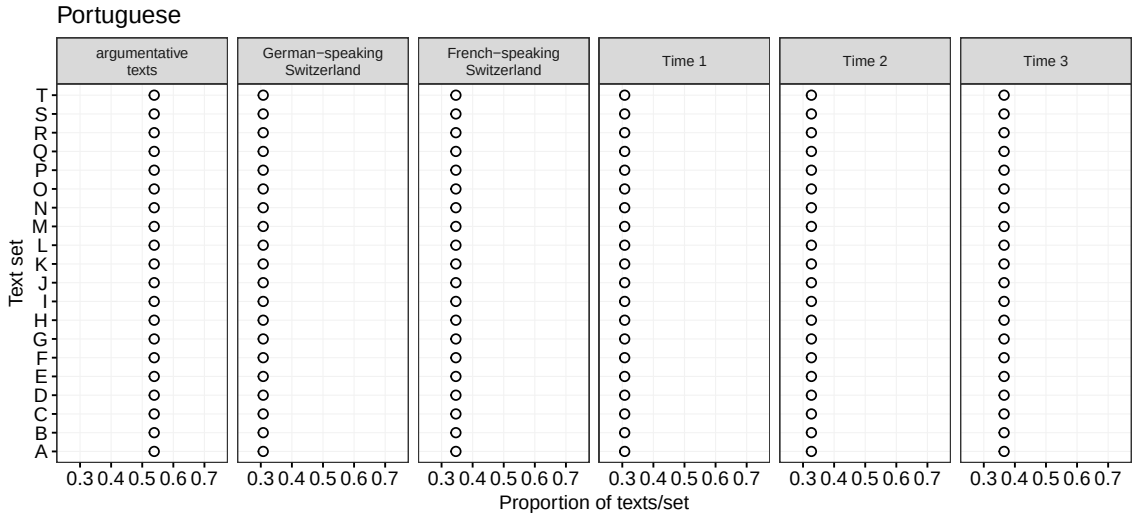

Figure 8.5: Variation between the Portuguese text sets in terms of the children's background, genre, and time of data collection.

Table 8.2: Distribution of texts and raters per set of texts. The total number of ratings does not include non-responses or responses that weren't logged.

	French	German	Portuguese
Sets	20	20	20
Texts per set	50	51	52
Texts (total)	1000	1020	1040
Mean number of raters per set	7.3	16.1	5.3
Minimum number of raters per set	4	11	3
Maximum number of raters per set	9	18	6
Ratings (total)	7284	16390	5505

20) was not assigned to any set and were left unrated so as to ensure that each batch consisted of the same number of texts. For Portuguese, considerably more than 1,000 texts were available, but fearing that we wouldn't be able to recruit a sufficient number of raters per text if we were to have all texts rated in about 33 batches of 50 texts each, we only retained 1,040 texts for rating.[2]

As for the length of the texts rated, the median text length was 37 tokens for French, 33 tokens for German, and 39 for Portuguese. For French, 90% of the texts consisted of 81 tokens or fewer; for German, of 66 tokens or fewer; for Portuguese, of 90 tokens or fewer. See Figure 8.6.

8.3 Rating procedure

The raters accessed an Internet platform where they first filled out a short questionnaire (native language(s), age, profession). They were then asked to read the 50–52 texts and rated the lexical richness of each text on a labelled 9-point scale (1 = very bad; 9 = very good; see Figures 8.7 and 8.8). After having rated all texts, the participants could optionally leave any comments they had regarding the task.

The raters were not told at which point in time the texts were written or that some texts were produced by by children with a Portuguese background residing in Switzerland.

The Internet platform was designed in the language of the texts to be rated (i.e., there was a French-, a German-, and a Portuguese-language version).

[2] 50 and 51 were the largest possible set sizes for French and German, respectively, whereas a set size of 52 for Portuguese ensured that each set had the same exact make-up in terms of the number of texts written by children from Portugal vs. French-speaking Switzerland vs. German-speaking Switzerland, narrative vs. argumentative texts, and the point in time at which the texts were written.

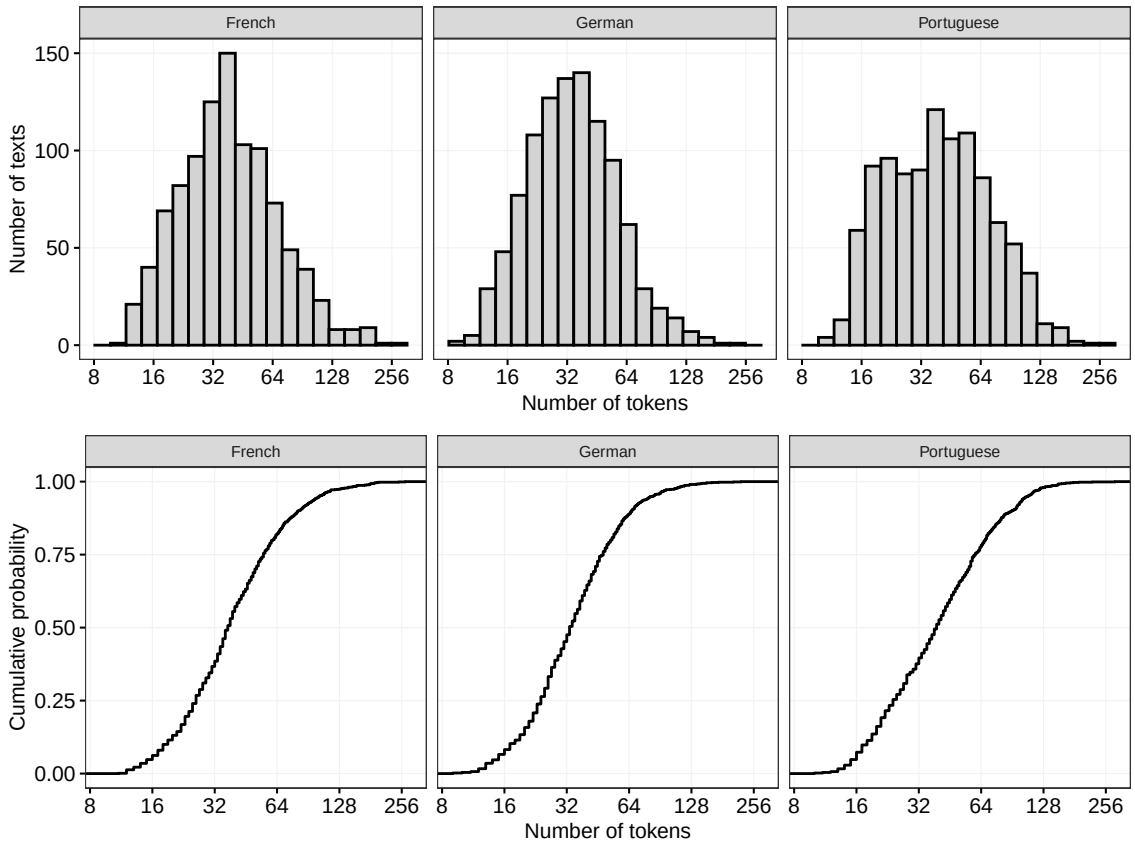

Figure 8.6: Length of the texts in tokens (logarithmic scale).

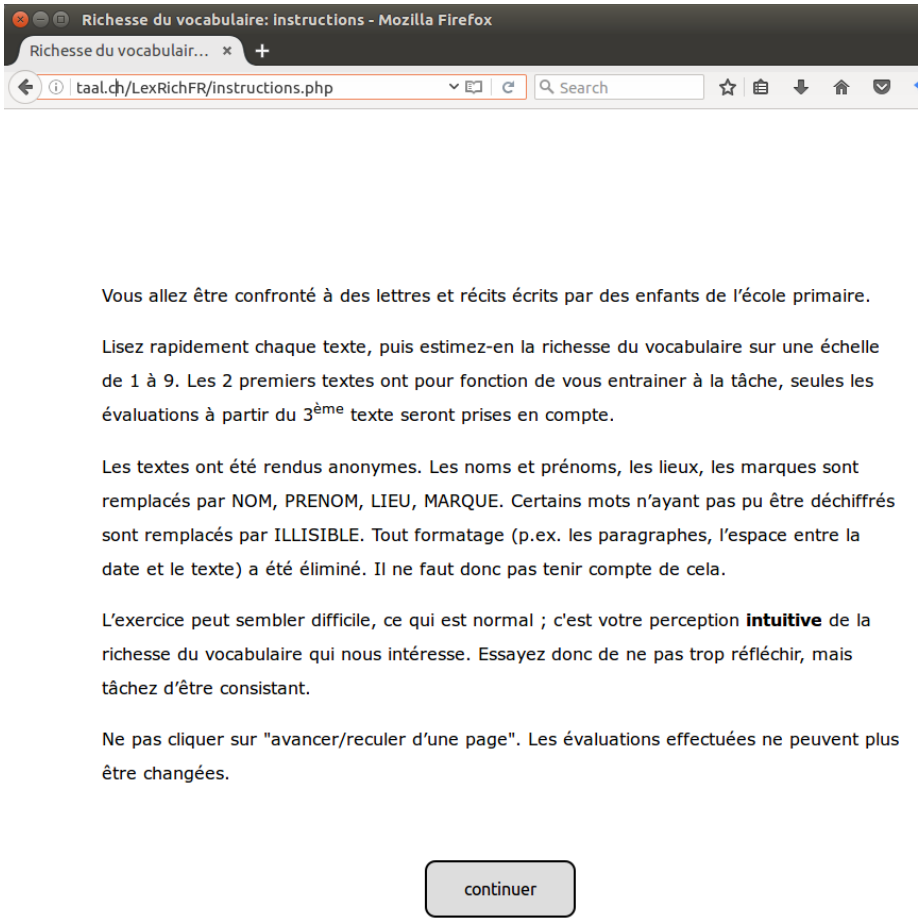

Figure 8.7: Instructions to raters (French version).

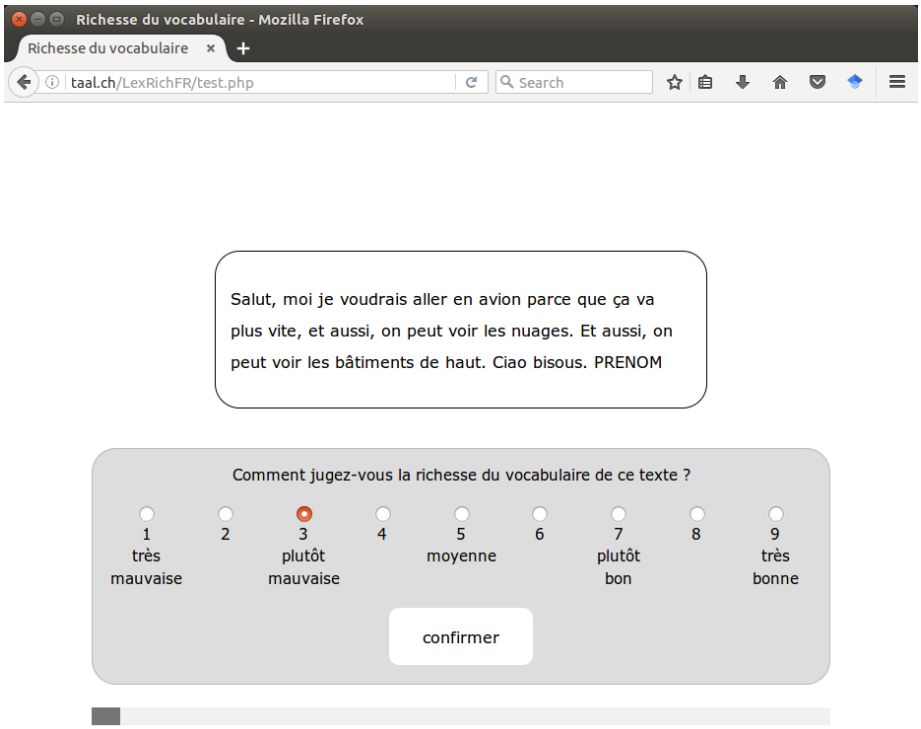

Figure 8.8: Rating scale (French version).

As raters trickled in, they were assigned to the set that had been completed by the fewest raters. If several sets shared the current minimum numbers of raters, the new rater was randomly assigned to one of these sets. This way, each set, and hence each text, would be rated by about the same number of raters, which wouldn't necessarily be the case had the raters been assigned to any random set of texts. Differences in the number of raters per set used in the analyses nonetheless occurred since some raters were excluded from the analyses (e.g., for not considering themselves native speakers).

The raters saw all texts within the set they were assigned to and only those (with the exception of the two training texts). The texts were shown in a new random order for each rater.

8.4 Piloting

Before the actual rating phase began, the French and German versions of the Internet platform were tested by some 20 pilot raters to check if the instructions were clear and the platform's design intuitive. These pilot raters were also asked to make a note of any remaining grammatical and orthographic errors. While not all pilot raters did so, this allowed us to iron out a handful of such errors that had slipped through the maze. Since the platform had been tested extensively for French and German, it was only pilot-tested by one additional rater, who was asked to focus particularly on the clarity of the Portuguese instructions and questionnaire items.

Chapter 9

Description of human ratings

Rather than working with the individual ratings, we computed the *mean* rating for each text as the outcome variable. The main reasons are, firstly, that we are at present more interested in modelling and predicting the average human rating of a text's lexical richness than in predicting ratings by a single rater and, secondly, that doing so greatly simplifies the analysis. When analysing the data, we took care to ensure that any findings would stand a good chance of applying to new texts *and* to new panels of raters.

9.1 Distribution

Figure 9.1 shows that the mean ratings per text are roughly symmetrically distributed.

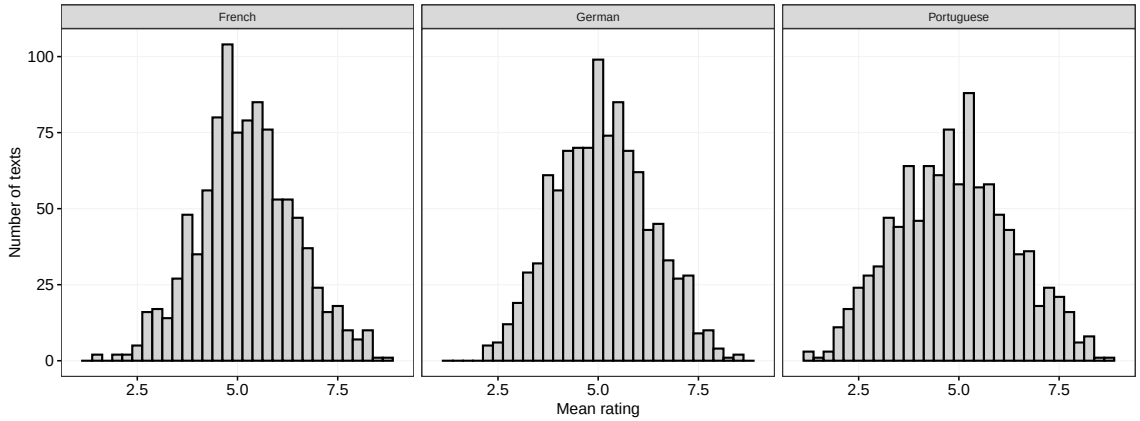

Figure 9.1: Average rating per text.

Figure 9.2 shows the distribution of the texts' mean ratings according to participant group, time of data collection, text type and language. Crucially, this figure reveals a systematic pattern where ratings of lexical richness increase with from the first through the third point of data collection and tend to be higher for the comparison groups than for the Portuguese–French and Portuguese–German bilinguals, even though the raters were not told how old the children were when they wrote each text nor whether the text was written by a child of Portuguese heritage or not. This systematicity indicates that, even when judging the lexical richness of short texts, the judgements of untrained raters are non-random.

9.2 Reliability

To assess more formally the extent to which raters responded non-randomly, a reliability measure was computed. Shrout & Fleiss (1979) discuss three types of rating study (p. 421):

1. "Each target is rated by a different set of k judges, randomly selected from a larger population of judges.
2. "A random sample of k judges is selected from a larger population, and each judge rates each target, that is, each judge rates n targets altogether.
3. "Each target is rated by each of the same k judges, who are the only judges of interest."

The present rating study doesn't correspond to any of these three types inasmuch as each judge rated only 20% of the targets (contra 2 and 3) and some targets were rated by the same set of judges (contra 1), namely all texts in the same set. However, the second type of study is—making allowance for a handful of missing responses—an apt description of the present study within each set of texts; the third type isn't as we seek to generalise beyond the present panels of raters. Since our interest lies in the reliability of the *mean* ratings and not of the individual ratings, the appropriate reliability coefficient for each set of texts is $ICC(2, k)$, where k is the number of raters.

Figure 9.3 shows the $ICC(2, k)$ reliability coefficients for each set of texts (computed using the `psych` package Revelle, 2021). The mean reliability coefficient is 0.79 (95% CI: [0.76, 0.82]) for French 0.90 (95% CI: [0.89, 0.92]) for German, and 0.71 (95% CI: [0.66, 0.77]) for Portuguese confirming our earlier impression that the raters' judgements were decidedly non-random, despite their lack of training and the short texts.[1]

It bears pointing out that these $ICC(2, k)$ estimates assume that the present set of raters is but a subset of the population of raters one wishes to generalise the findings to ('raters

[1] In terms of constructing a predictive model, this means that the theoretically maximal variance in the ratings such a model could account for is capped at 79%, 90%, and 71%, respectively.

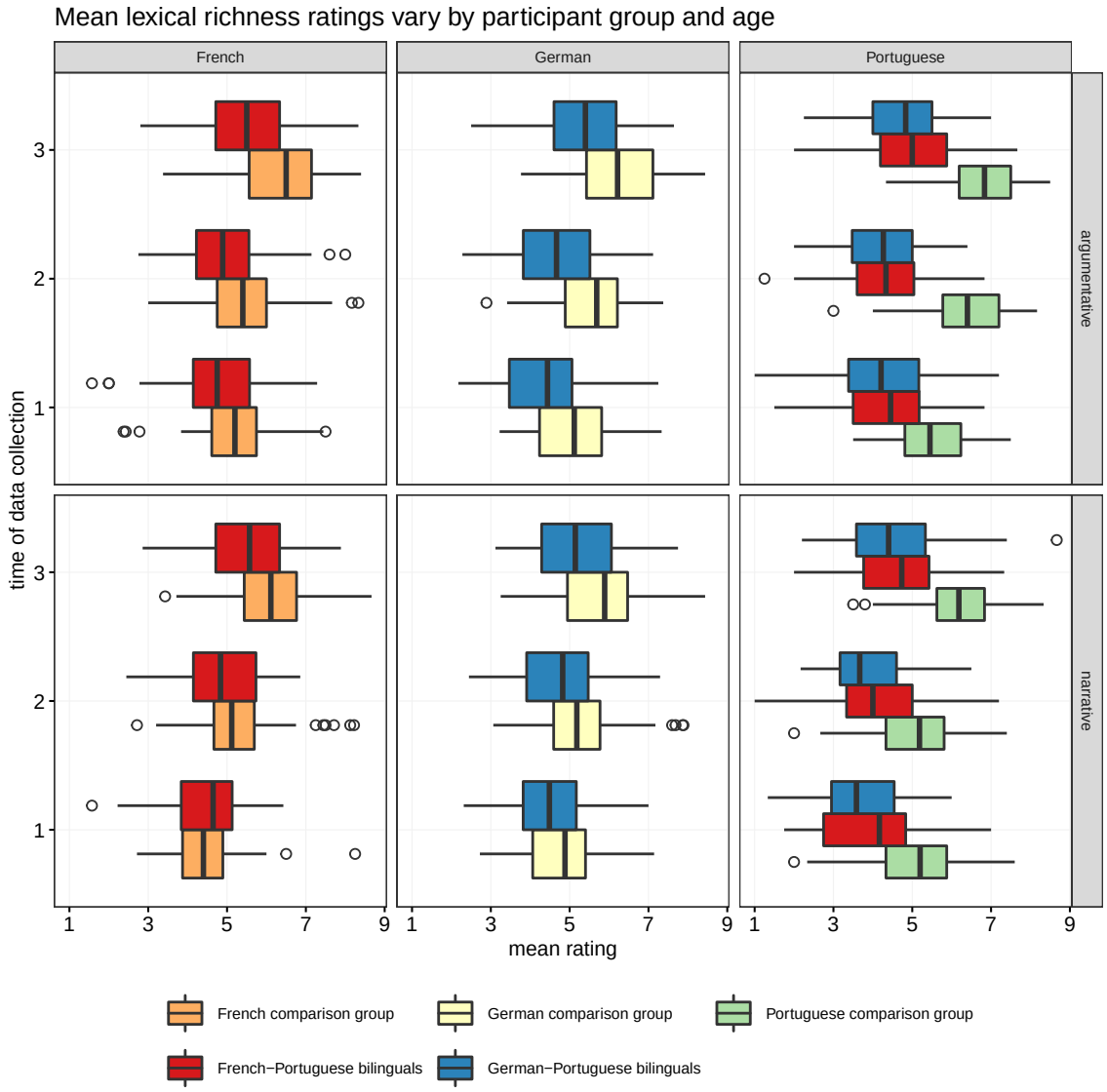

Figure 9.2: Distribution of the texts' mean ratings according to participant group, time of data collection, text type and language.

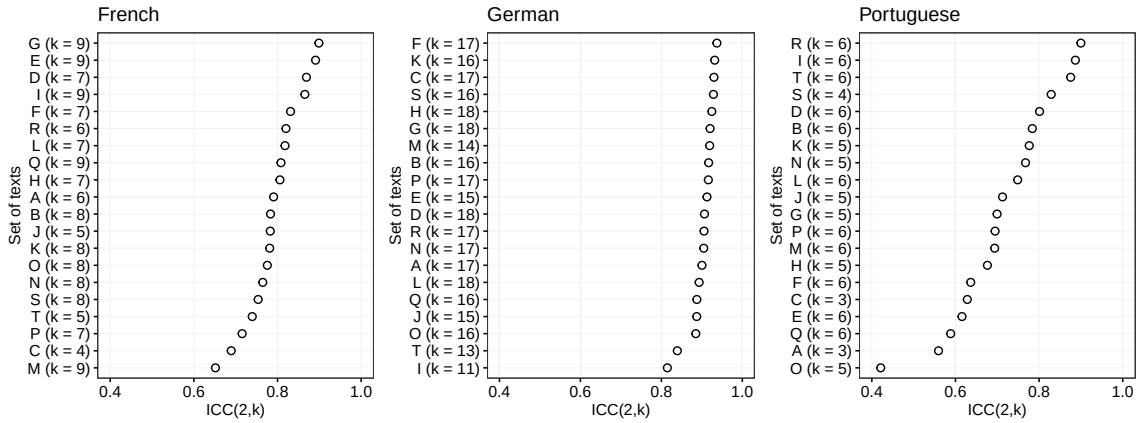

Figure 9.3: Intraclass correlation coefficient (ICC) for each set of texts. The set labels are arbitrary; set A for French and set A for German are unrelated. (k = number of raters)

as a random effect'). The often-used Cronbach's α—or, equivalently, $ICC(3, k)$—assumes that *only* the present set of raters is of interest and that one doesn't wish to generalise to raters in general. The corresponding Cronbach's $\alpha/ICC(3, k)$ values would be higher (0.84, 0.93, and 0.81, for French, German, and Portuguese, respectively).

9.3 Within-set correlation

The different text sets were rated by different panels of judges. This could have given rise to a hierarchical structure in the data if some panels of judges happened to be stricter than others. Figure 9.4 suggests this may indeed be the case to some degree. The intraclass correlation associated with the text sets for the French ratings was 0.04; for German <0.01; for Portuguese 0.15.

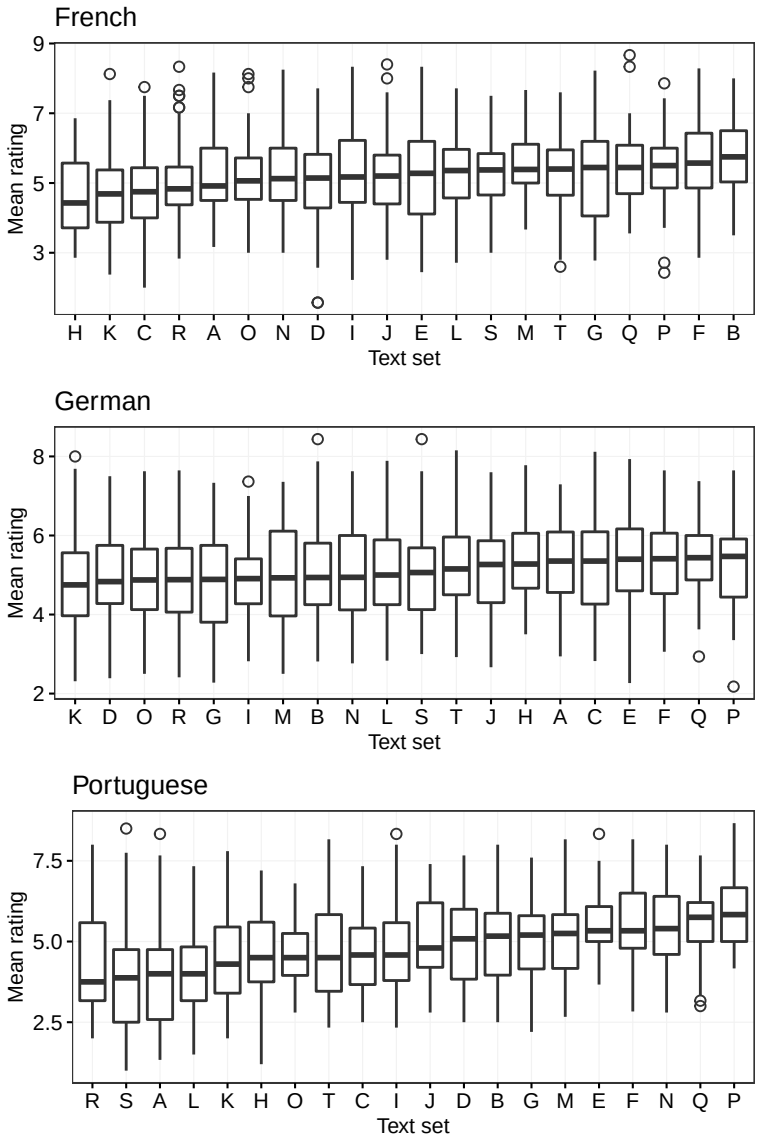

Figure 9.4: For French, the mean ratings are slightly correlated within each set (ICC = 0.04). For German, this didn't seem to be the case (ICC < 0.01). For Portuguese, more strongly so (ICC = 0.15).

Part V

Predictive modelling of human judgements of lexical richness

Chapter 10

Modelling strategy

10.1 Goal

The goal of the analysis was to model the human ratings of the texts' lexical richness in terms of text-based properties. To this end, we used as the outcome/dependent variable the mean human rating per text. The text-based properties which serve as possible predictors/independent variables included measures of lexical diversity and lexical sophistication derived from the texts as well as more mundane features such as the texts' length. Since it was not a priori clear which measures of lexical diversity and sophistication predict human ratings or how precisely such measures should be computed (see Chapters 5 and 6), this analysis necessarily included a substantial exploratory component. To offset the danger that extensive data exploration led to a model that tightly fits the present data but that does not apply to similar but new data ('overfitting'),[1] we applied techniques from predictive modelling (or machine learning; see Kuhn & Johnson, 2013).

10.2 Data partitioning

For each language, the data was split into a *training set* and a *test set*. The training set consisted of 16 randomly selected text sets; the test set consisted of the 4 remaining text sets per language (i.e., a 80%–20% split). The data in the text set, then, were not affected by texts in the training set nor by the judges who rated the texts in the training set.

To the training set, we applied a host of exploratory and modelling techniques (see Kuhn & Johnson, 2013, for an overview) in order to find one or several statistical models that

[1] Importantly, the often-used adjusted R^2 metric is *not* immune to overfitting if the modelling stage features an exploratory stage: http://janhove.github.io/analysis/2016/04/22/r-squared.

have the greatest predictive power. This step included:

1. trying out different implementations of the diversity and sophistication measures, as well as trying out some new operationalisations. All measures eventually used are documented in the earlier chapters;
2. trying out different data transformations. Mostly this was done automatically (using the Yeo–Johnson family of transformations), but for the more interpretable GAMs, the predictor data were manually transformed;
3. tweaking model parameters;
4. judging the models' interpretability.

Resampling techniques using the training set helped us to adjudicate between different models and specifications. This allowed for great flexibility in terms of choosing, transforming, and combining variables without overestimating the models' explanatory power.

The test set help us judge how well the final model (arrived at on the basis of the training data) would fit to new data. Since it consisted of texts not included in the training set that were judged by raters not included in the data set, applying the model of our choice to the test set allowed us to ascertain how well the model generalised to how panels of *new raters* would judge similar but *new texts*.

10.3 Resampling techniques and optimisation criterion

When trying out different models and model specifications on the training data, cross-validation was used to estimate how well the model would work for new data. This technique consists of splitting up the training set into k 'folds', training the model on $k-1$ of the folds, and then using it to predict the data in the remaining fold. This training–then–predicting process is repeated k times so that each fold is used $k-1$ times for training and one time for prediction.

Cross-validation is typically based on randomly defined folds. One potential disadvantage of this in the present context is that it doesn't take into account the dependency structure of the data: the mean ratings aren't independent of one another inasmuch some are based on ratings by the same judges. Cross-validation based on random folds could yield overoptimistic estimates of how well the findings generalise to new panels of raters if the same panels of raters are included in both the training and hold-out folds. To address this concern, the models' generalisability was assessed using *block cross-validation* (see Roberts et al., 2017): each of the sixteen sets of texts in the training data was used 15 times for training and once for prediction. This idea is similar to ordinary cross-validation but the

folds aren't constructed randomly. See Figue 10.1.

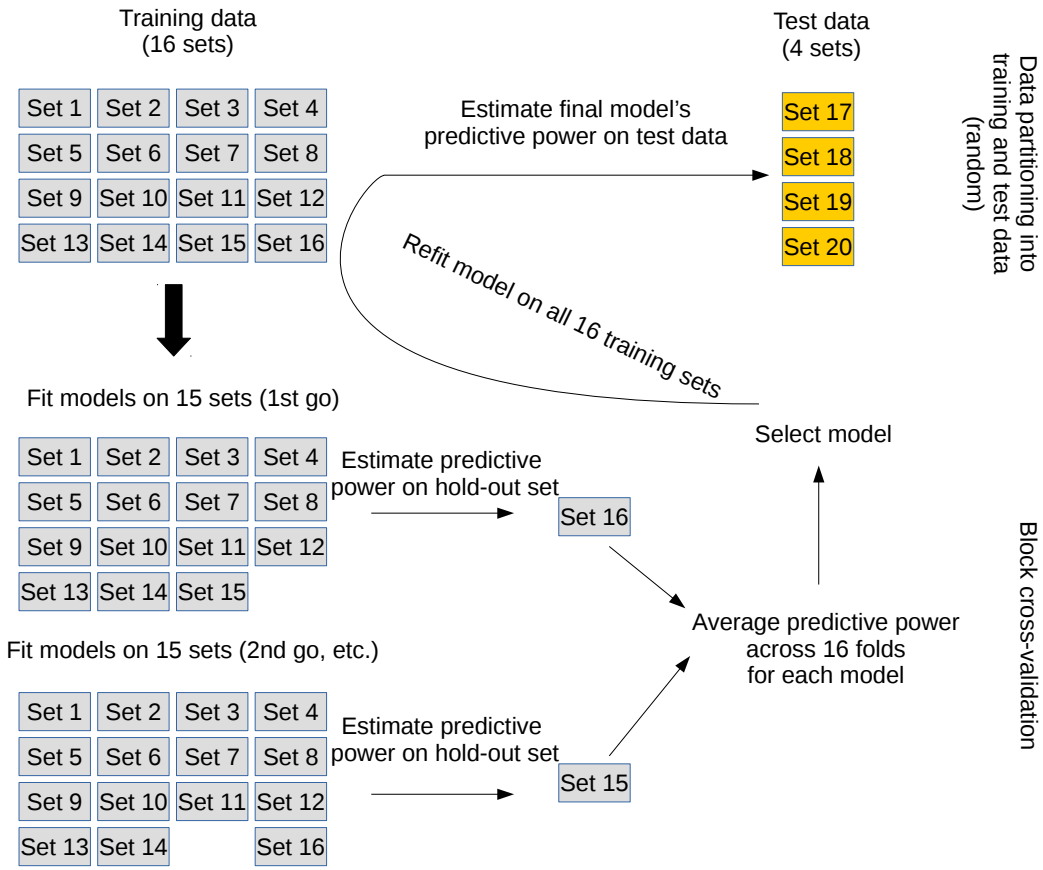

Figure 10.1: Illustration of how the data were partitioned into a training and a test set and of how block cross-validation works. Only two iterations of block cross-validation are shown; in reality, sixteen took place for each model. Each 'Set' refers to 50–52 texts that were rated by rated by a panel of judges. The panels of judges for different sets did not overlap.

Model fit was evaluated using the *root mean square error*, that is, the square root of the mean squared discrepancy between the model's predictions and the actually observed values ($\sqrt{\frac{1}{n}\sum(\hat{y_i} - y_i)^2}$). The R^2 metric was of secondary importance, but is also reported below.

We use the RMSE as it expresses directly how well the model predictions correspond to the observed values. The problem with R^2 is that there exist different formulae for

computing R^2 (see Kvålseth, 1985). For ordinary regression models, these all yield the same result. However, when the model is used to predict observations that were not used when fitting the model, they do not. One popular method for computing R^2, namely computing the correlation between the predicted and observed values and squaring it, is particularly problematic, since the correlation between the predicted and observed values can be excellent even if the former correspond poorly to the latter (e.g., the values 1, 2, 3 correlate perfectly with the values 1000, 3000, 9000 but correspond poorly to them). The R^2 values in this article were therefore calculated as the proportional reduction in the residual sum of squares relative to a baseline model without any predictors: $R^2 = 1 - \frac{\sum(y_i - \hat{y}_i)^2}{\sum(y_i - \hat{y}_0)^2}$, where y_i is a text's perceived lexical richness, $\hat{y}_i$ its predicted perceived lexical richness by the predictive model, and $\hat{y}_0$ its 'predicted' perceived lexical richness by a model without any predictors (i.e., an intercept-only model).

10.4 Predictor selection

Unsurprisingly, a host of predictor variables were highly correlated with each other. Based on the intercorrelations between predictors in the training data, a number of predictors that were highly correlated with other predictors was weeded out prior to modelling.

10.5 Predictive models and algorithms applied

The following models and algorithms were applied:

- Regression trees (CART);
- Random forests (including bagging), both the CART- and conditional importance (CI) flavours;
- Support vector machines (SVM);
- k-nearest neighbours (KNN);
- Multiple linear regression, with and without the prior application of principal component analysis (PCA);
- Robust regression (with PCA);
- Ridge regression;
- Elastic net (including LASSO);
- Partial least squares regression (PLS);
- Multivariate adaptive regression splines (MARS);
- Stochastic gradient boosting;
- Cubist.

See Kuhn & Johnson (2013) for details. Some tutorials geared to language researchers can

be found in Baayen (2008), Tagliamonte & Baayen (2012).

10.6 Interpretation and further tuning

For the models with the best (lowest) RMSE, the relationship between the mean out-of-fold predictions and the actual outcome was investigated. Any conspicuously outlying oder underpredicted texts were manually inspected to see if further variables could be derived from them and added to the model. The new variable thus proposed did not substantially help the models' predictive power, however.

10.7 Model stacking

Greater predictive accuracy can sometimes be achieved by generating predictions on the basis of multiple models and using these predictions as the input for a new predictive model. This technique is known as STACKING (Wolpert, 1992; Breiman, 1996) and was applied to the current data set as well.

10.8 Models with fewer predictors

Many of the predictive models and algorithms applied to the data are known as 'black boxes', that is, the precise way in which they relate a set of input values to a predicted outcome can be difficult to understand (but see Goldstein et al., 2015). Indeed, even for linear regression models with many, often correlated predictors it can be difficult to assess the independent effect of each predictor on the outcome: a model coefficient might tell you how the outcome is expected to change when varying the TTR but keeping the numbers of types and tokens constant, but this is obviously impossible to do. Moreover, models with substantial different architectures were often estimated to have similar predictive power. While this underscores that when it comes to predicting outcomes on the basis of moderately rich predictor data, there are many ways to skin a cat (the 'Rashomon effect', see Breiman, 2001), it further compounds their lack of interpretability. Thus, while our main aim concerned prediction, we also tried to fit more transparent models whose construction was guided by Jarvis (2013a)'s 6- dimensional theoretical framework (a 6-predictor model) and by a desire to fit as simple a model as possible (a single-predictor model) to see how well the black boxes' predictive power could be emulated using only a handful of predictors.

Chapter 11

Predictive modelling: French

11.1 Data splitting

Text sets D, K, P and T were randomly selected and together constituted the test set. These 200 observations were not looked at during data exploration and model tuning/selection.

11.2 Predictor transformation

Many predictor variables were right-skewed so that a Yeo–Johnson transformation (Yeo & Johnson, 2000) was applied to the entire predictor set.[1] Of the 154 predictors, 153 were transformed in order to get a more symmetrical distribution. The predictors were subsequently centred at their mean in the training data and scaled using their standard deviation in the training data.

When tuning models, predictor transformations was effected during cross-validation.

11.3 Bivariate relationship between ratings and predictors in training data

The bivariate relationships between the mean ratings and the transformed predictors as well as among the transformed predictors themselves were inspected. In the scatterplot matrices that follow, the distribution and name of the variables are shown on the main diagonal. The upper triangle shows scatterplots and a LOESS fit (in blue). The bottom

[1] The Yeo–Johnson family of transformations is similar to the Box–Cox (Box & Cox, 1964) family of transformation but accommodates data with zeroes and negative values.

triangle shows the squared correlation between the y variable and the LOESS fit ($\hat{y}$): values close to 1 indicate that one variable can be entirely expressed as a (linear or nonlinear) function of the other; values close to 0 indicate that the variables are orthogonal to one another.[2]

Since the number of predictors is too large to show in one plot, several scatterplot matrices are shown. On the basis of these scatterplot matrices, highly correlated variables were identified and removed.

11.3.1 Tokens, types and lemmata

See Figure 11.1. `nTypes`, `TTR` and `Guiraud` removed.

11.3.2 TTR variations

See Figure 11.2. `Carroll`, `Dugast`, `Summer` and `Brunet` removed.

11.3.3 LTR variations

See Figure 11.3. `CarrollLemma`, `DugastLemma`, `SummerLemma` and `BrunetLemma` removed.

11.3.4 TTR variations vs. LTR variations

See Figure 11.4. `Herdan`, `Rubet`, `Maas` and `LN` removed.

11.3.5 MSTTR

See Figure 11.5. Nothing removed.

11.3.6 MTLD

See Figure 11.6. `MTLD61` and `MTLD72` removed.

11.3.7 HDD

See Figure 11.7. `HDD31` removed.

11.3.8 MATTR

See Figure 11.8. `MATTR65` removed.

[2]Unlike Pearson correlation coefficients, these values aren't necessarily symmetrical: their value depends on which variable is on the x- and which is on the y-axis. Seeing as a substantial number of variables were non-linearly correlated, sometimes even non-monotonously, the LOESS-based estimates were still used.

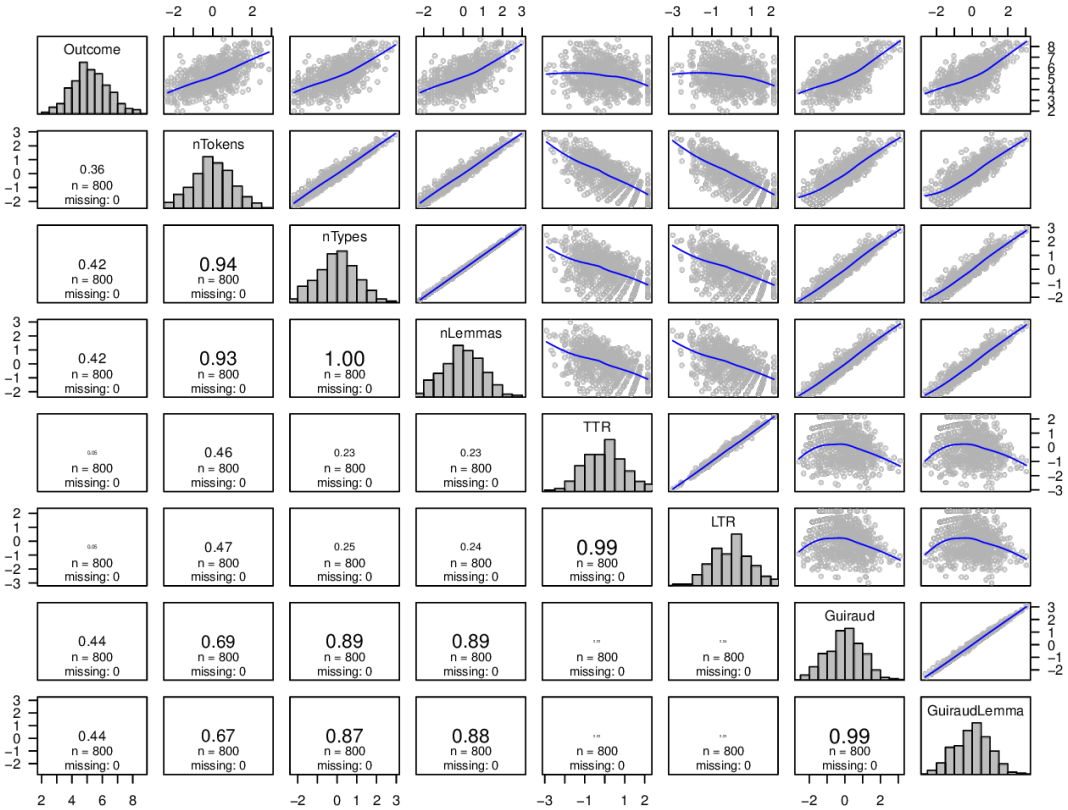

Figure 11.1: Intercorrelation between French predictors (1): Tokens, types and lemmata. `nTypes`, `TTR` and `Guiraud` were removed because of their strong intercorrelations with the other variables; `nTokens` was retained for now, despite its high intercorrelations.

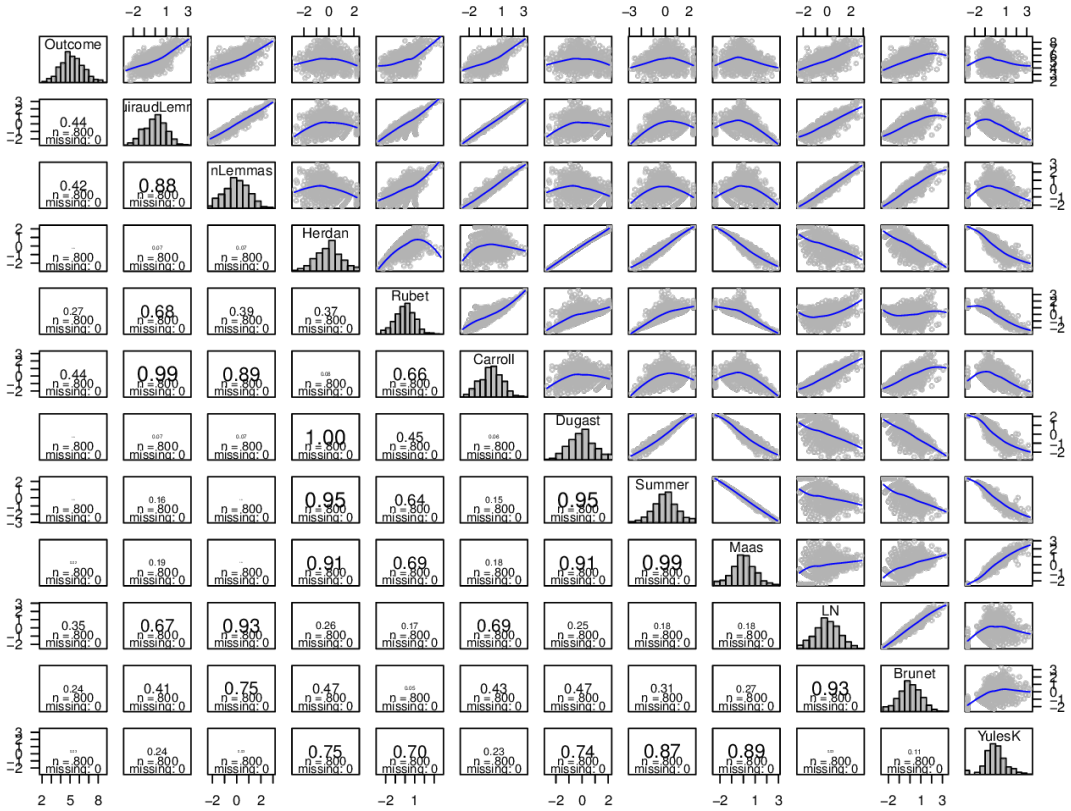

Figure 11.2: Intercorrelation between French predictors (2): TTR variations. `Carroll`, `Dugast`, `Summer` and `Brunet` were removed because of their strong intercorrelations with the other variables.

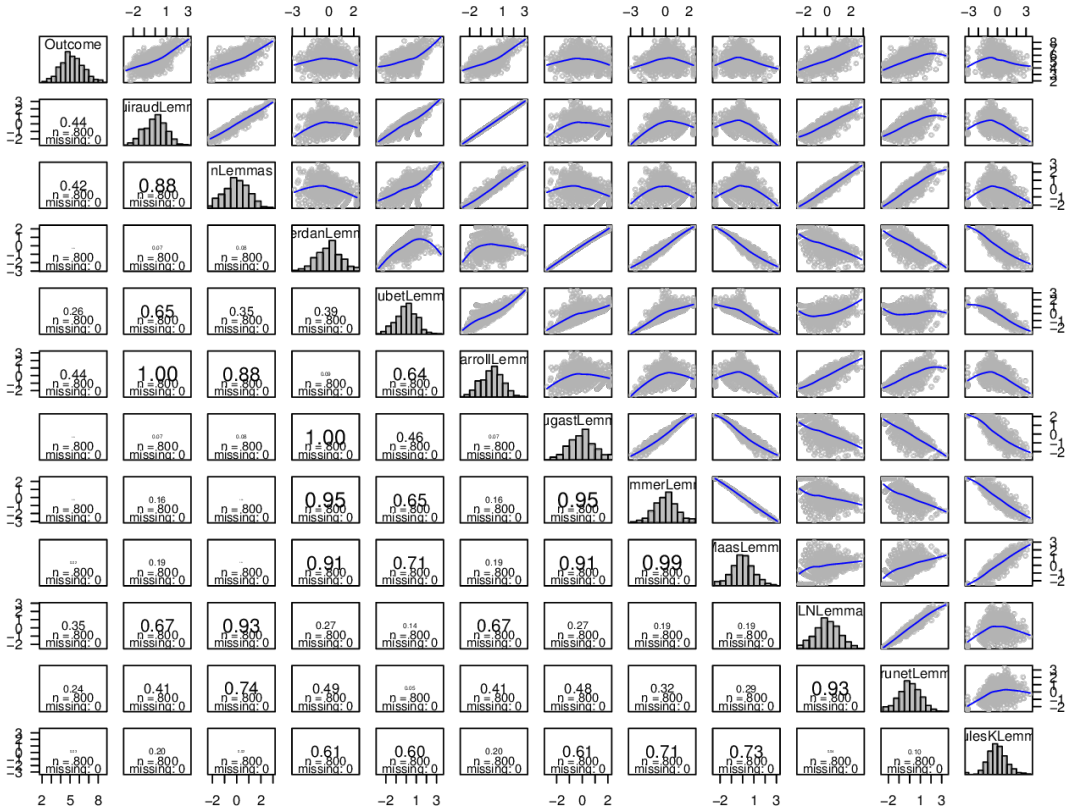

Figure 11.3: Intercorrelation between French predictors (3): TTR variations. `CarrollLemma`, `DugastLemma`, `SummerLemma` and `BrunetLemma` were removed because of their strong intercorrelations with the other variables.

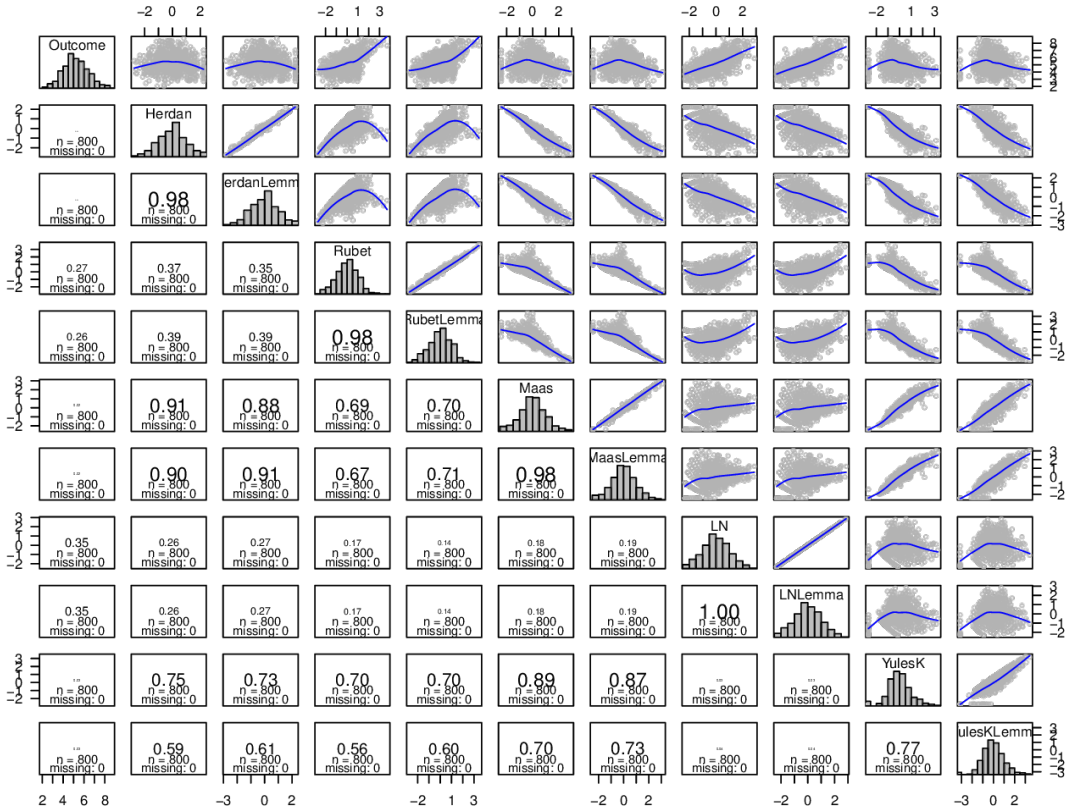

Figure 11.4: Intercorrelation between French predictors (4): TTR vs. LTR variations. `Herdan`, `Rubet`, `Maas` and `LN` were removed because of their strong intercorrelations with the other variables.

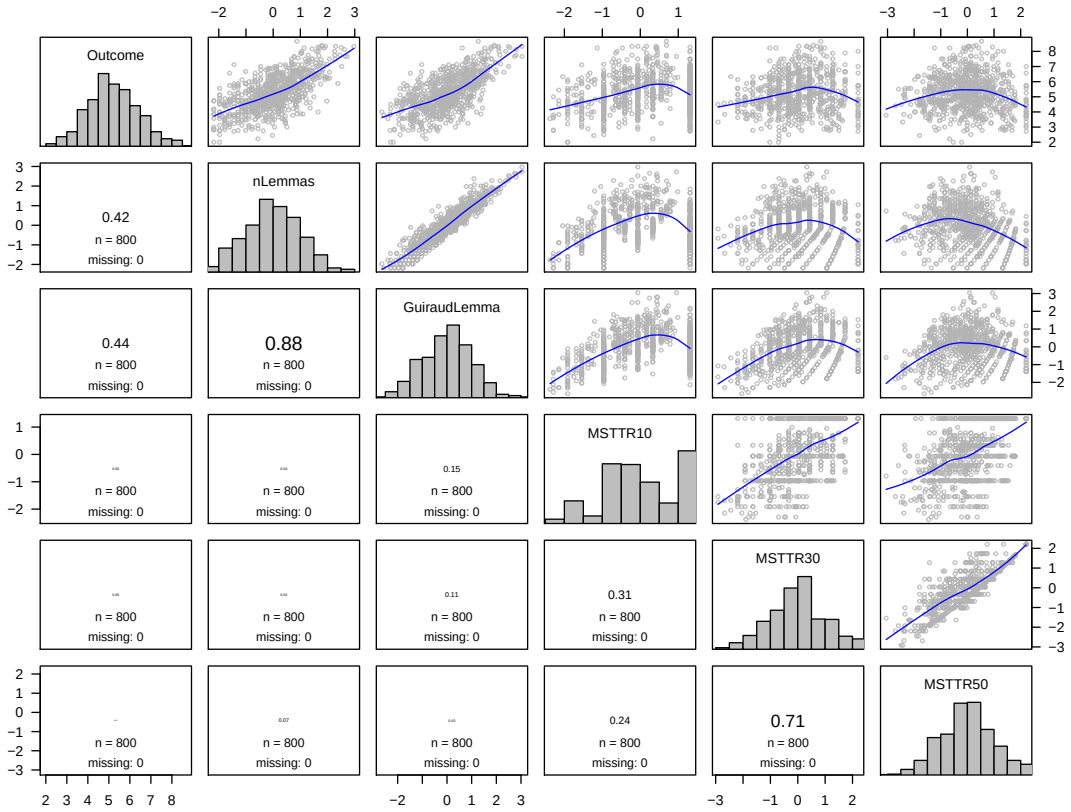

Figure 11.5: Intercorrelation between French predictors (5): MSTTR. No variables were removed because of their strong intercorrelations with the other variables.

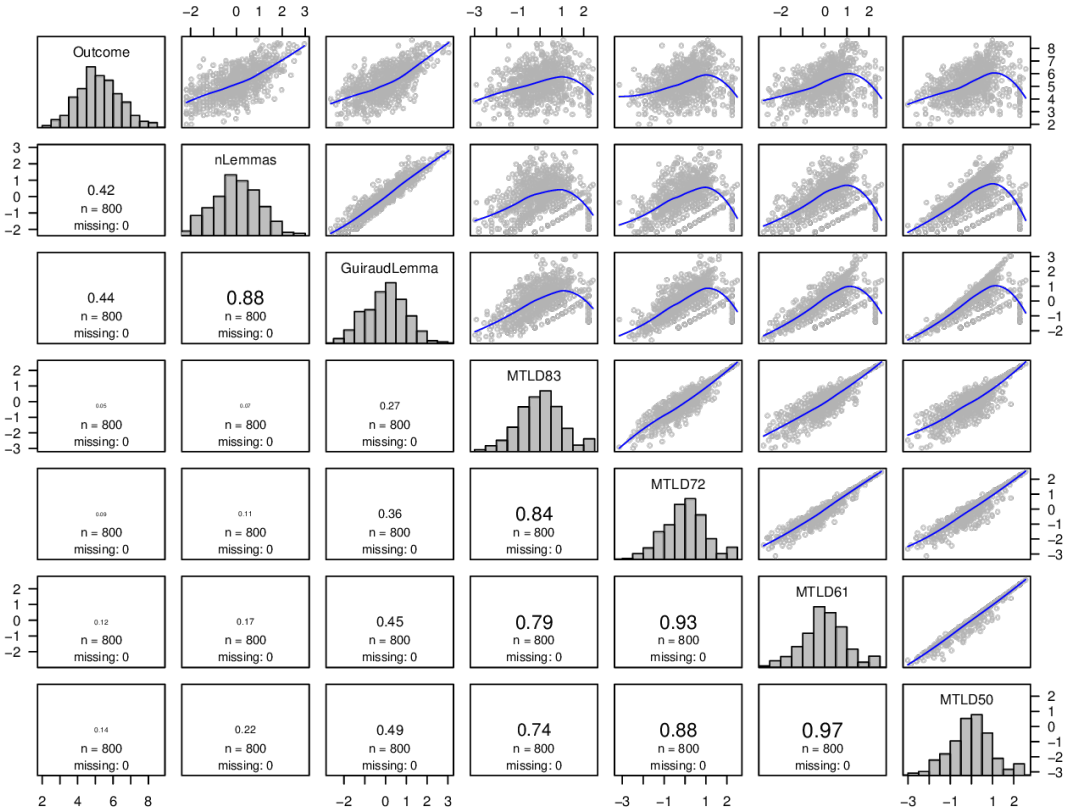

Figure 11.6: Intercorrelation between French predictors (6): MTLD. `MTLD61` and `MTLD72` were removed because of their strong intercorrelations with the other variables.

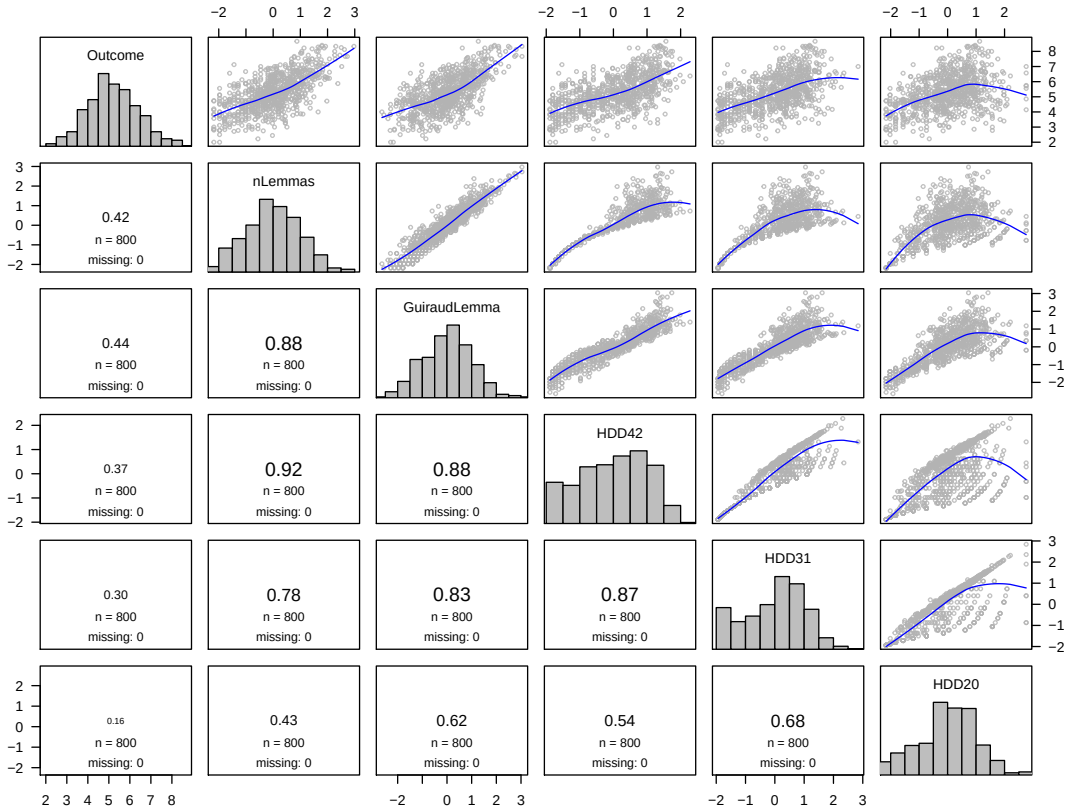

Figure 11.7: Intercorrelation between French predictors (7): HDD. `MTLD31` was removed because of their strong intercorrelations with the other variables.

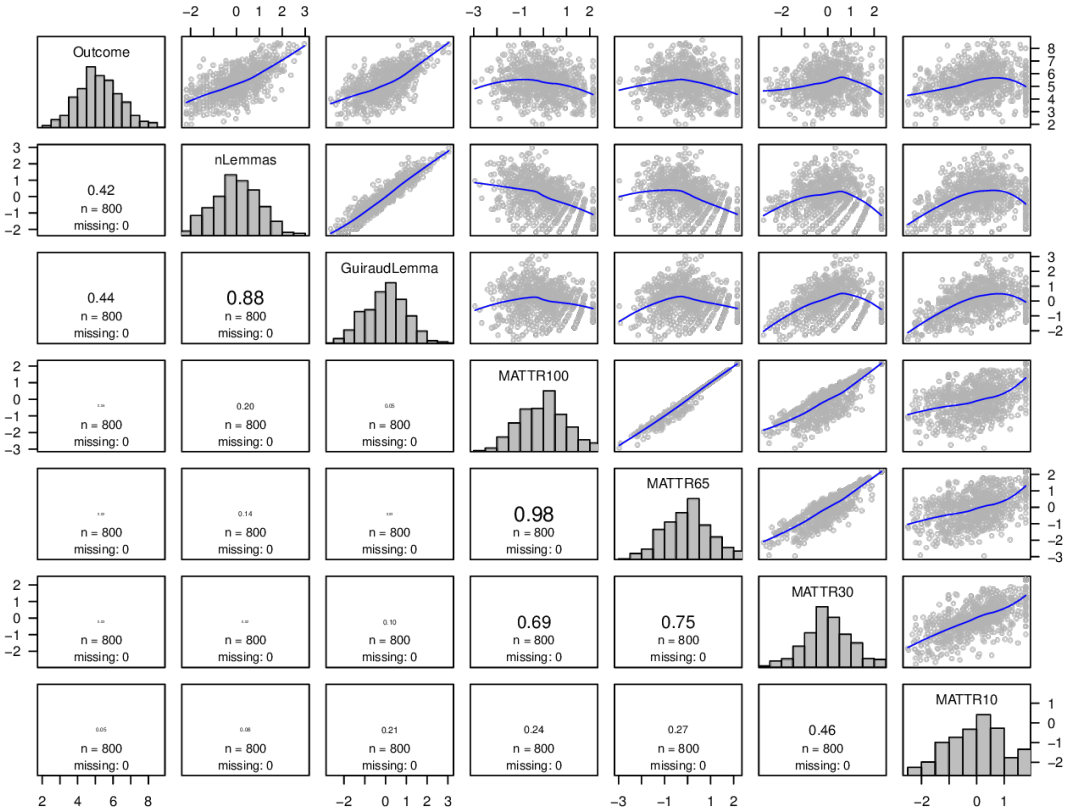

Figure 11.8: Intercorrelation between French predictors (8): MATTR. `MATTR65` was removed because of their strong intercorrelations with the other variables.

11.3.9 Lexical and syntactic complexity

See Figure 11.9. `lexWordNumber` removed.

11.3.10 Top frequency bands

See Figure 11.10. Nothing removed.

11.3.11 Frequency summaries (token- and type-based)

See Figure 11.11. Nothing removed.

11.3.12 Frequency summaries (lemma-based)

See Figure 11.12. `meanFreqRank` and `meanFreqRankUnique` removed.

11.3.13 Number of types and tokens by POS

See Figures 11.13 and 11.14. `nNounTokens`, `nNounTypes`, `nVerbTokens`, `nVerbTypes`, `nAdjTokens`, `nAdjTypes`, `nAdvTokens` and `nAdjTypes` removed.

11.3.14 TTR by POS

See Figure 11.15. `TTR.Noun`, `TTR.Verb`, `TTR.Adj` and `TTR.Adv` removed.

11.3.15 Advanced Guiraud/TTR

See Figure 11.16. `AdvancedLTR1000`, `AdvancedTTR1000`, `AdvancedGuiraudLemma1000`, `AdvancedGuiraud1000`, `AdvancedTTR2000`, `AdvancedGuiraudLemma2000` and `AdvancedGuiraud2000` removed.

11.3.16 Evenness, disparity and dispersion

See Figure 11.17. `dispersion_type_30` and `dispersion_lemma_30` removed.

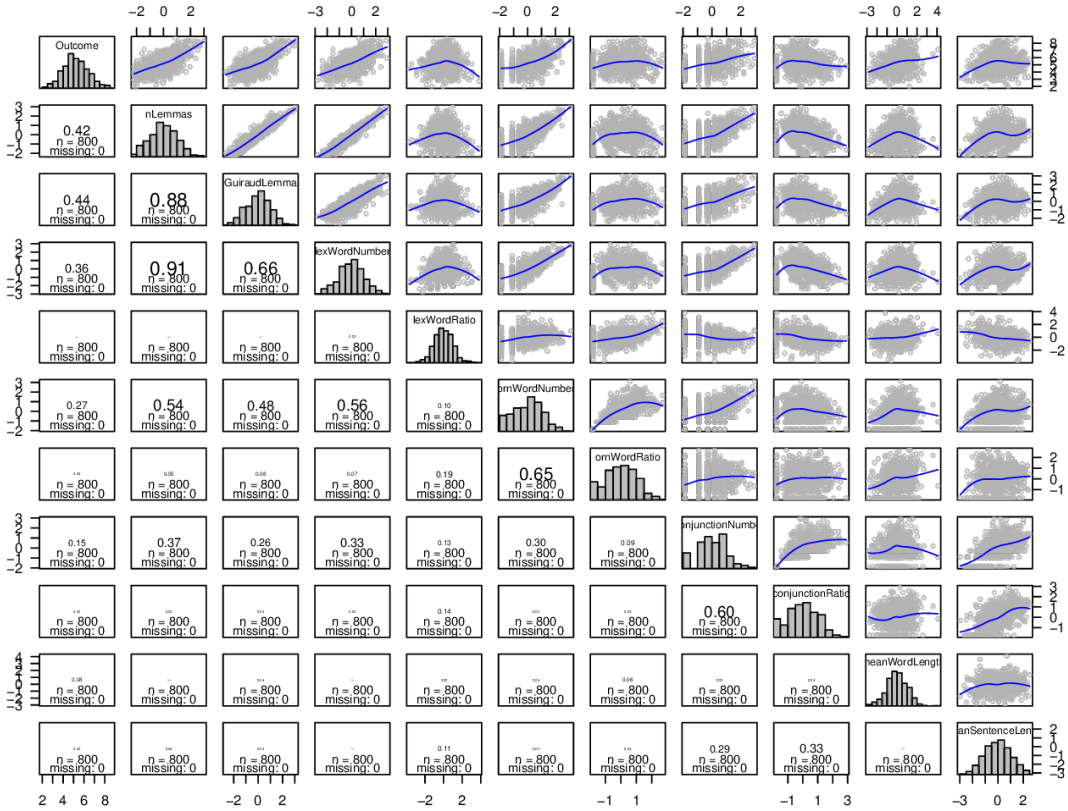

Figure 11.9: Intercorrelation between French predictors (9): Lexical and syntactic complexity. `lexWordNumber` was removed because of their strong intercorrelations with the other variables.

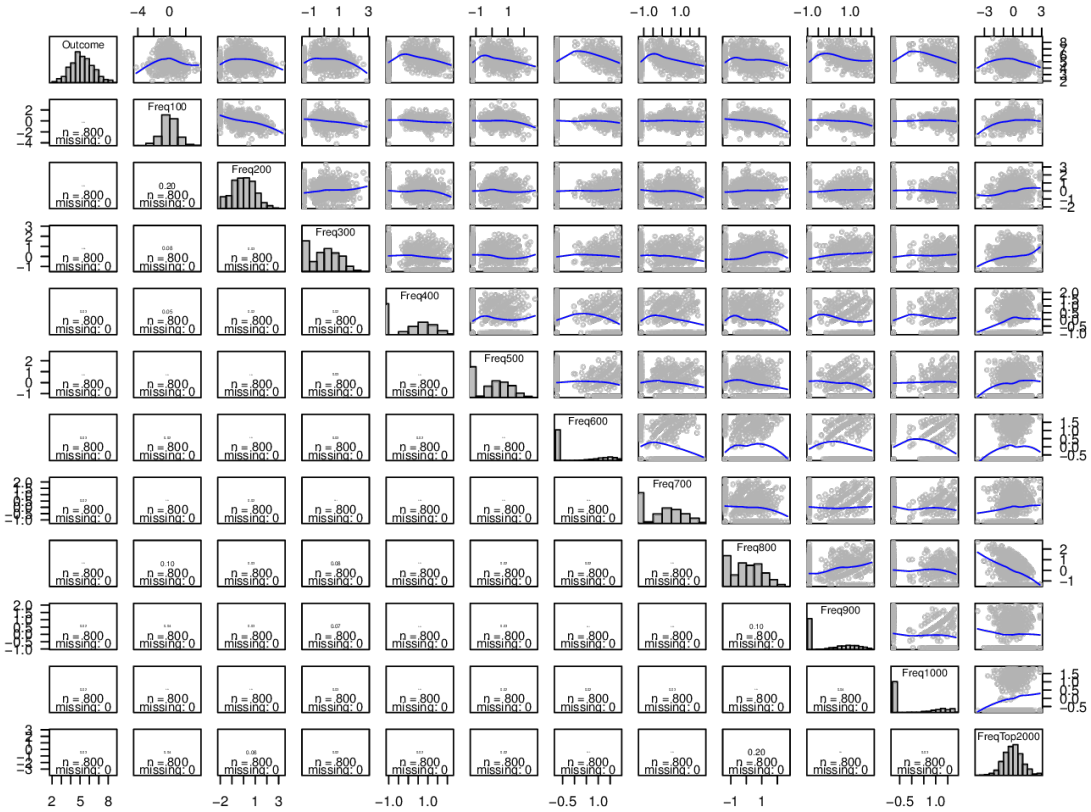

Figure 11.10: Intercorrelation between French predictors (10): Top frequency bands. No variable was removed because of their strong intercorrelations with the other variables.

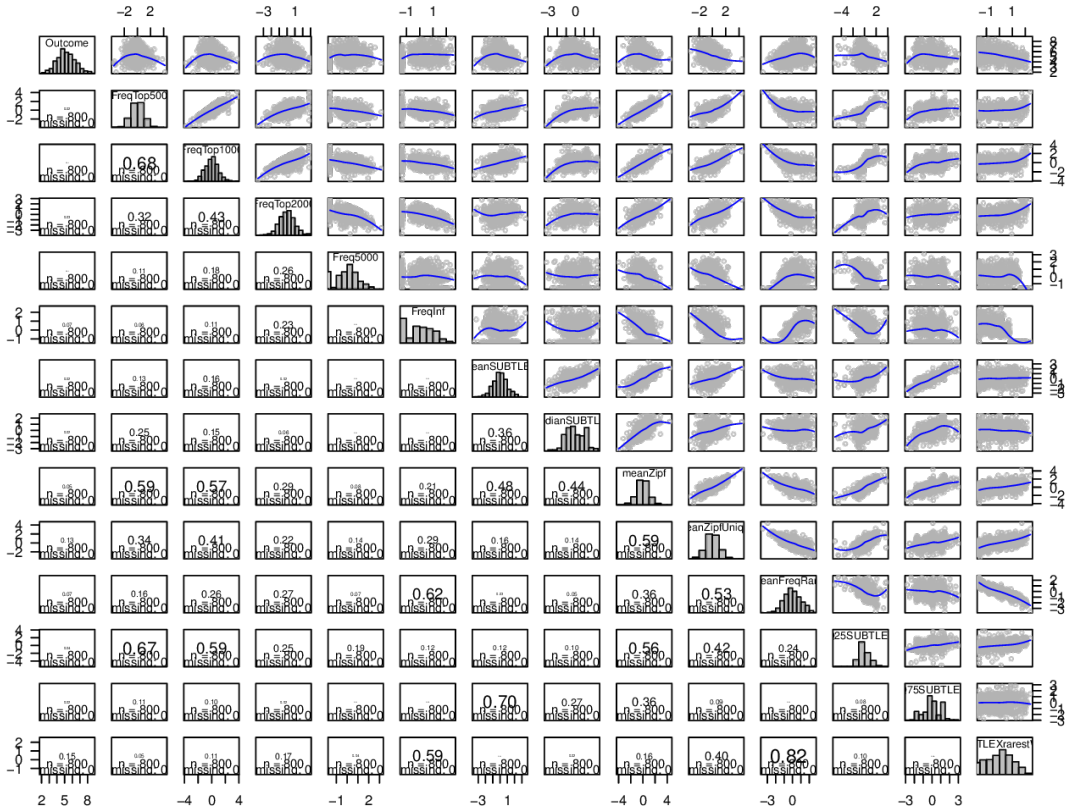

Figure 11.11: Intercorrelation between French predictors (11): Frequency summaries (token- and type-based). No variable was removed because of their strong intercorrelations with the other variables.

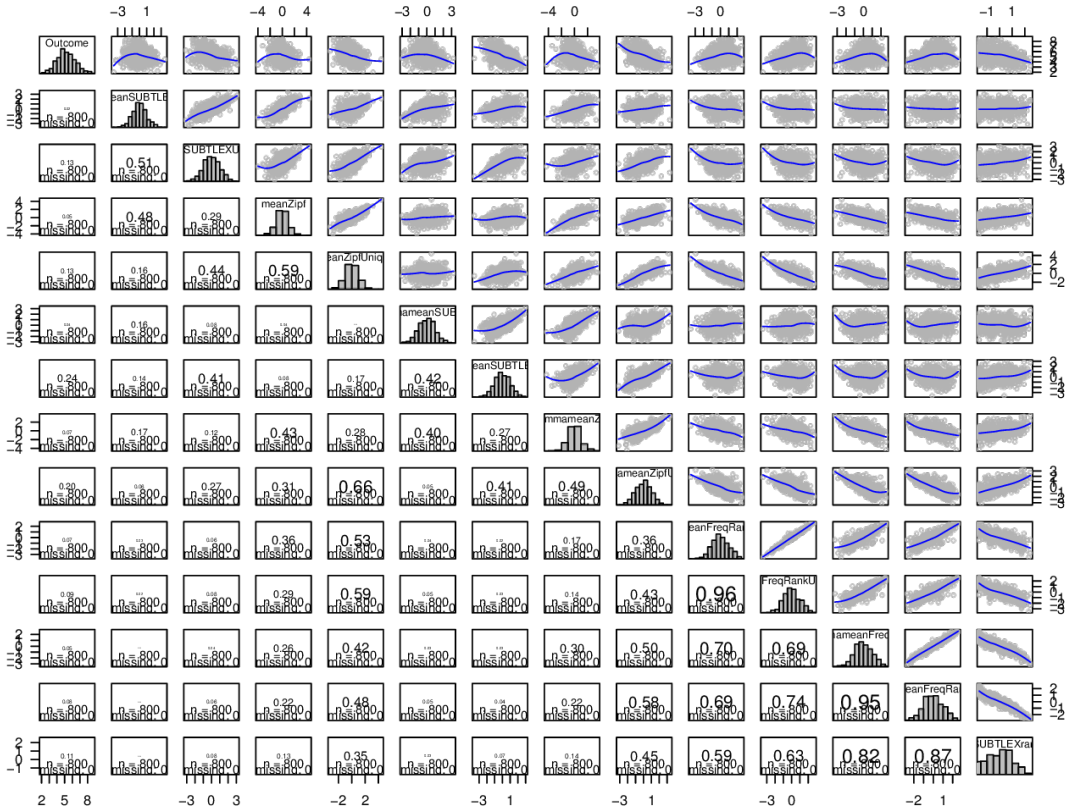

Figure 11.12: Intercorrelation between French predictors (12): Frequency summaries (lemma-based). `meanFreqRank` and `meanFreqRankUnique` were removed because of their strong intercorrelations with the other variables.

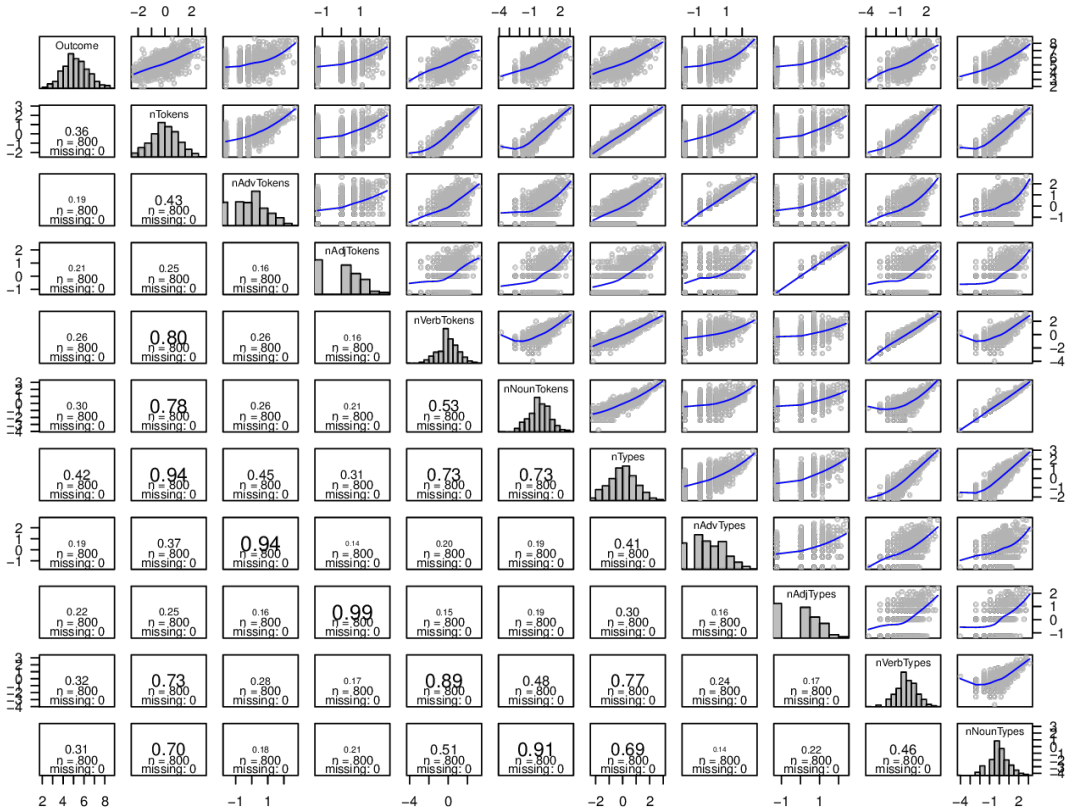

Figure 11.13: Intercorrelation between French predictors (13): Number of types and tokens by POS.

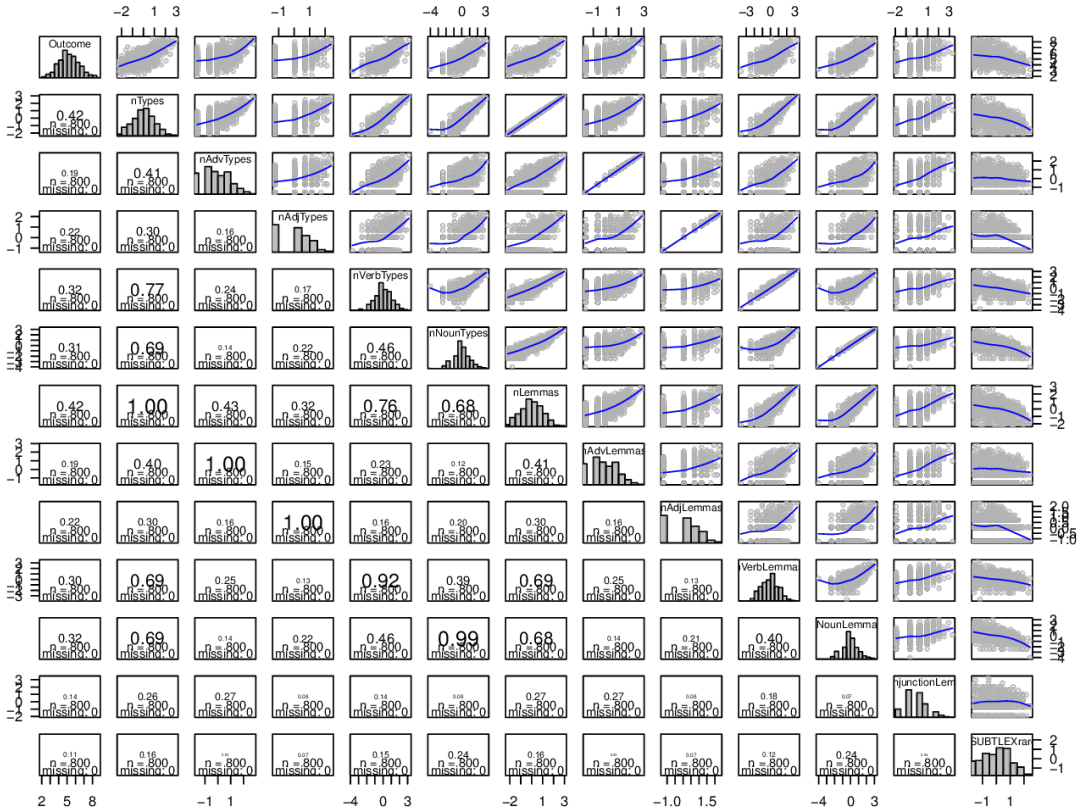

Figure 11.14: Intercorrelation between French predictors (14): Number of lemmas by POS. `nNounTokens`, `nNounTypes`, `nVerbTokens`, `nVerbTypes`, `nAdjTokens`, `nAdjTypes`, `nAdvTokens` and `nAdjTypes` were removed because of their strong intercorrelations with the other variables.

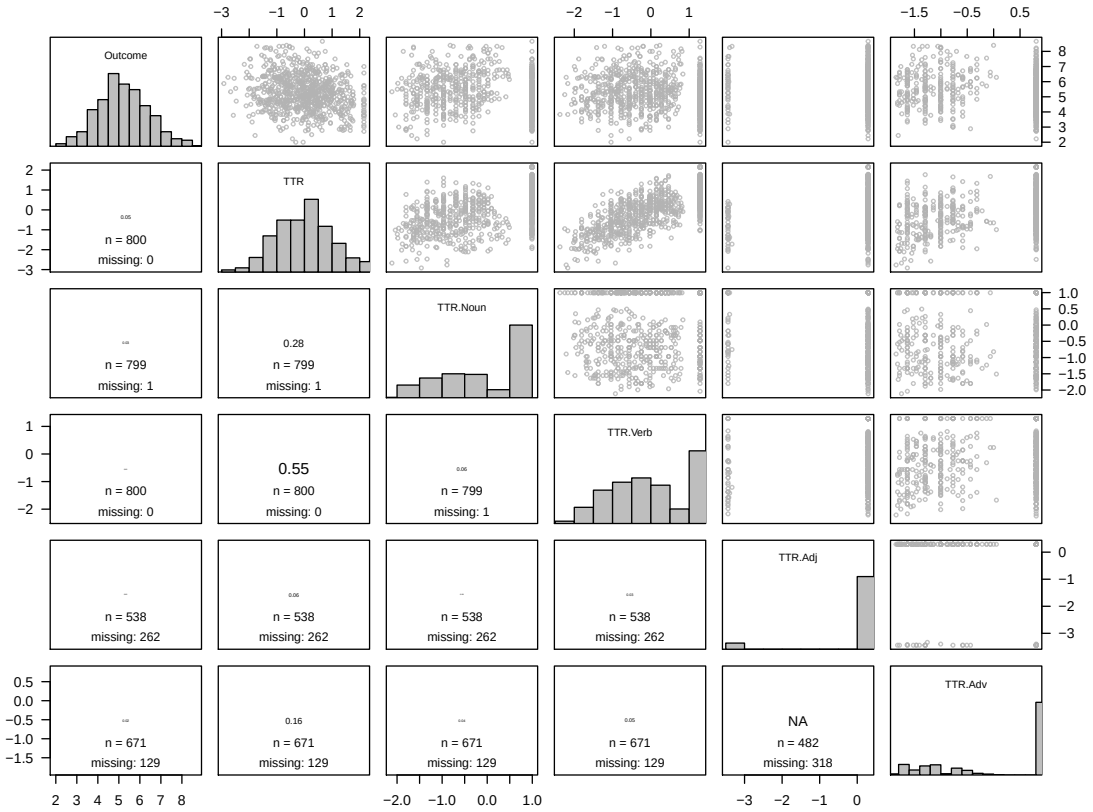

Figure 11.15: Intercorrelation between French predictors (15): TTR by part of speech. `TTR.Noun`, `TTR.Verb`, `TTR.Adj` and `TTR.Adv` removed due to missing values.

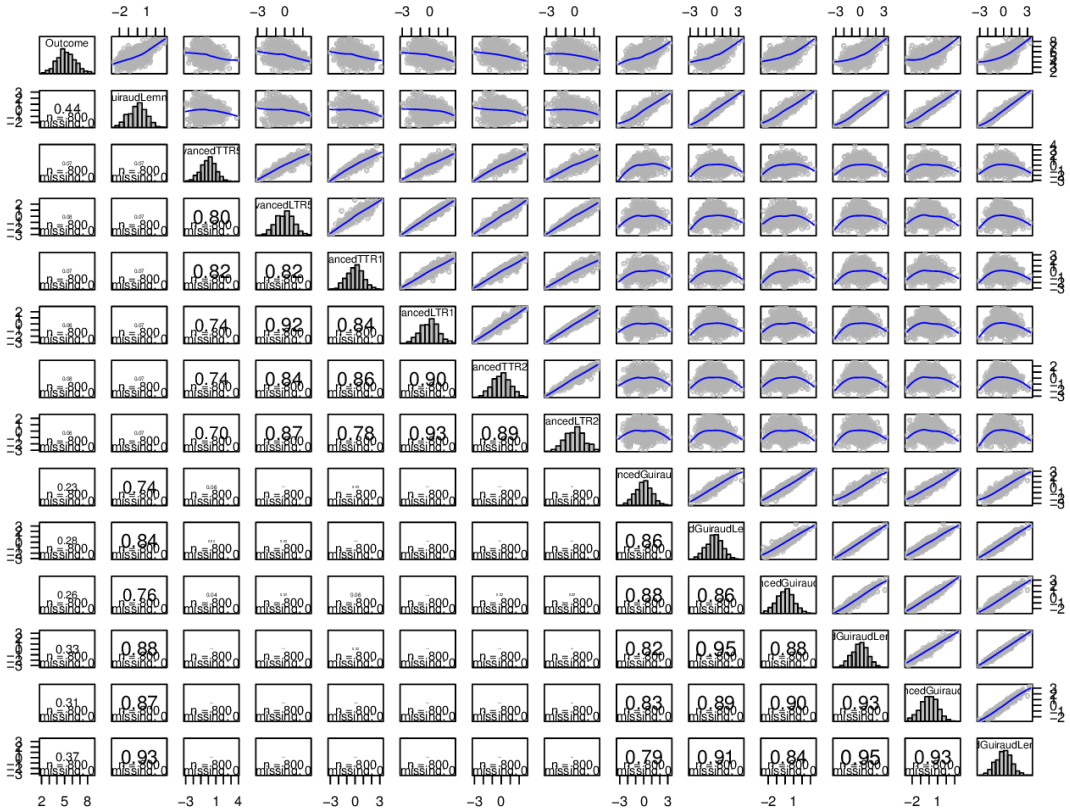

Figure 11.16: Intercorrelation between French predictors (16): Advanced Guiraud and TTR. `AdvancedLTR1000`, `AdvancedTTR1000`, `AdvancedGuiraudLemma1000`, `AdvancedGuiraud1000`, `AdvancedTTR2000`, `AdvancedGuiraudLemma2000` and `AdvancedGuiraud2000` removed due to high intercorrelations.

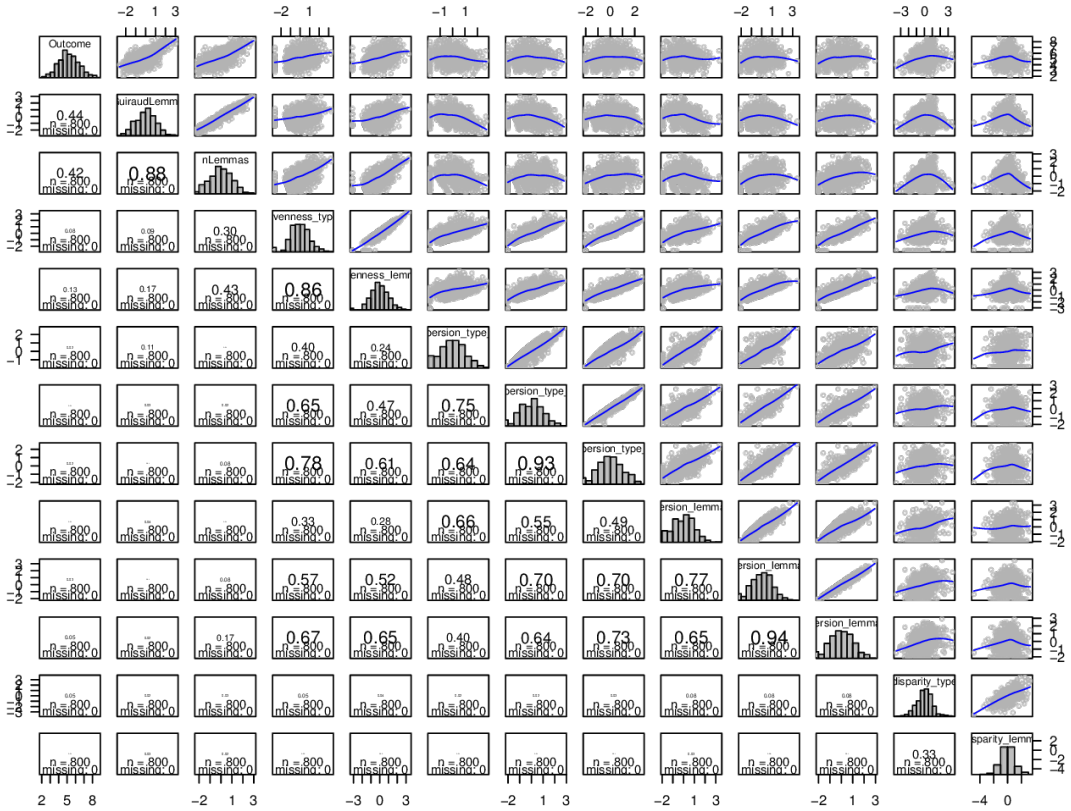

Figure 11.17: Intercorrelation between French predictors (16): Evenness, disparity and dispersion. `dispersion_type_30` and `dispersion_lemma_30` removed due to high intercorrelations.

11.4 Model performance in cross-validation

Several different algorithms were fitted to the training data and tuned using block cross-validation. For most models, the data were Yeo–Johnson transformed. The models are not described in detail here, but see Kuhn & Johnson (2013). Figure 11.18 shows the estimated predictive accuracy of 14 tuned models. The algorithm with the greatest predictive power was Cubist, with a mean RMSE of 0.828. The elastic net/LASSO and stochastic gradient boosting algorithms closely followed with mean RMSE values of 0.832 and 0.833, respectively. Given that the outcome data was measured on a 9-point scale, the differences between these RMSEs are clearly inconsequential.

11.5 Model stacking

The out-of-fold predictions for all 14 models were extracted. The Pearson correlations between the out-of-fold predictions of the different models varied between 0.80 and 0.9998 with a median correlation of 0.935. These strong correlations suggest that any gain in predictive accuracy from model stacking will be small.

The out-of-fold predictions of each model were used as predictors in a principal component regression model. This is a linear regression model for which principal component analysis was first applied to the predictors. The predictive accuracy of this model was assessed block cross-validation The RMSE was estimated to be 0.826. This represents an ever so slight increase over the single best model.

11.6 Does predictive accuracy depend on text length?

In view of warnings that many lexical diversity measures ought not to be applied to short texts, it is useful to assess whether the models' predictive accuracy is worse in short texts. Figure 11.19 shows the correlations between text length and the out-of-fold predictions and residuals according to the stacked model. While longer texts are predicted to have better ratings (left), the variance of the residuals hardly varies according to text length (middle and right). In conclusion, the models' predictive accuracy is about equally as good for short as for longer texts.

11.7 Variable importance in top-3 models

Figure 11.20 shows the variable importances of the 20 most important predictors in the top 3 models (Cubist, stochastic gradient boosting and elastic net). These values were extracted using `caret`'s `varImp()` function, and for each of the three models, the variables

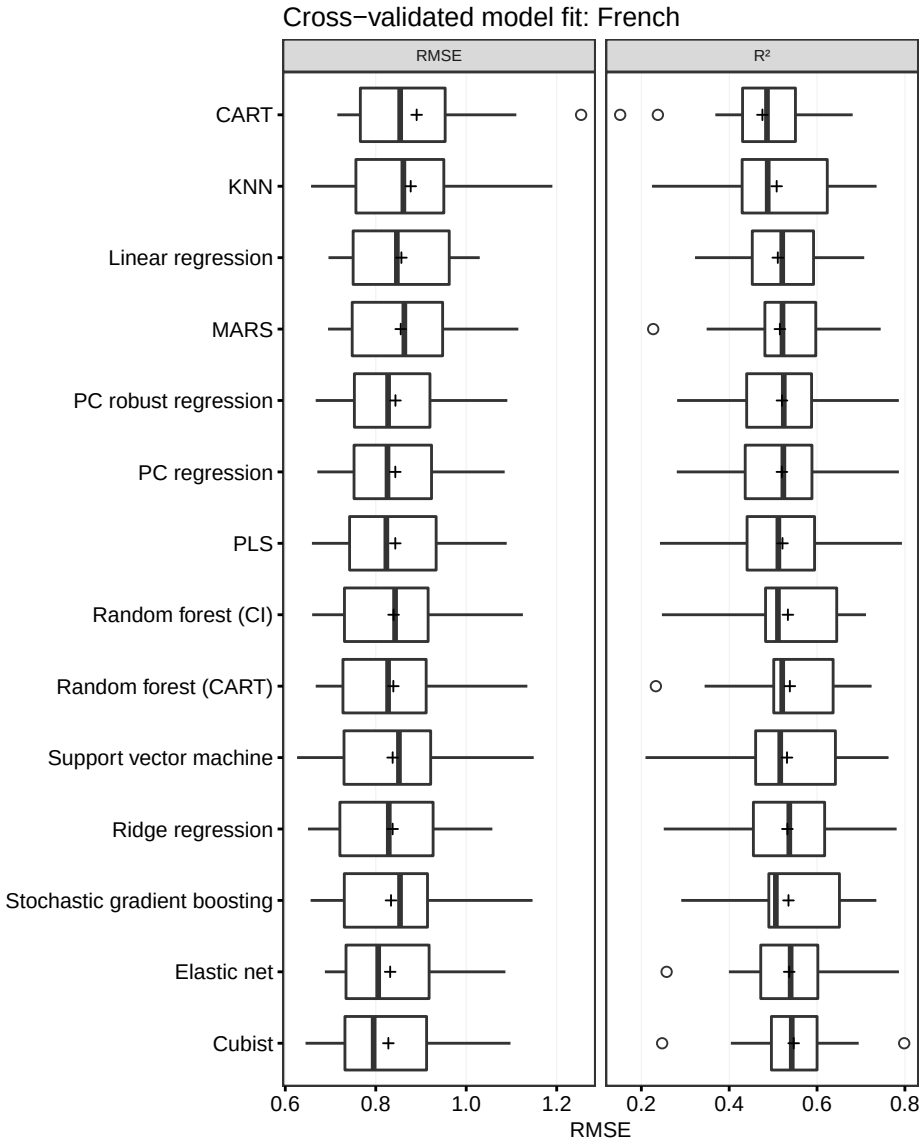

Figure 11.18: Performance of 14 tuned predictive models for the French training data in block cross-validation (with 16 blocks). The crosses mark the mean of each distribution.

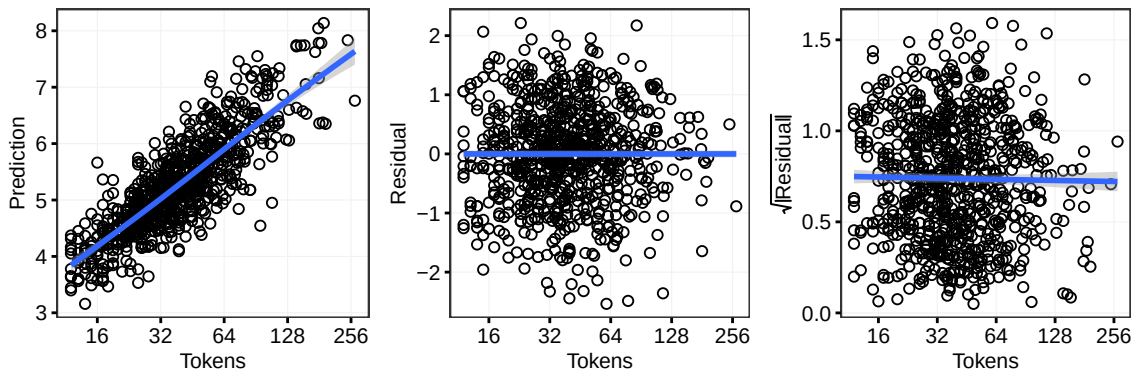

Figure 11.19: *Left:* Text length and out-of-fold prediction according to the stacked model for all 800 training texts. *Middle:* Text length and residuals (actual value − average out-of-fold prediction). *Right:* Text length and root absolute residuals.

were rank-ordered from less to more important. The twenty variables with the highest mean rank across the three models are shown in the plot.

The single most important variable in all three models is `GuiraudLemma`, that is, $\frac{\text{number of lemmata}}{\sqrt{\text{number of tokens}}}$. Another important variable in the three models is the number of lemmata occurring in the texts. After these two variables, the models diverge noticeably. In particular, the stochastic gradient boosting algorithm recognises fewer predictors as important compared to the Cubist and elastic net algorithms. Recall, however, that all three algorithms are highly comparable in terms of their predictive power and that their out-of-fold predictions are highly correlated ($r > 0.95$). These three models, then, achieve the same goal with different means (cf. Breiman's (2001) *Roshomon effect*).

11.8 A 6-dimensional model

A more interpretable 6-dimensional model based on the framework proposed by Jarvis (2013a,b) was fitted to the training data. This model contained the following predictors:

- Volume: The number of tokens. (Log-transformed)
- Variability: MTLD with a TTR setting of 0.83. The MTLD was chosen as it is not systematically affected by the texts' length. (Log-transformed)
- Evenness: The lemma-based evenness index. (Square-root transformed)
- Rarity: The mean Zipf value of the unique lemmata occurring in the texts. Other rarity indices were tried and yield fairly similar results.

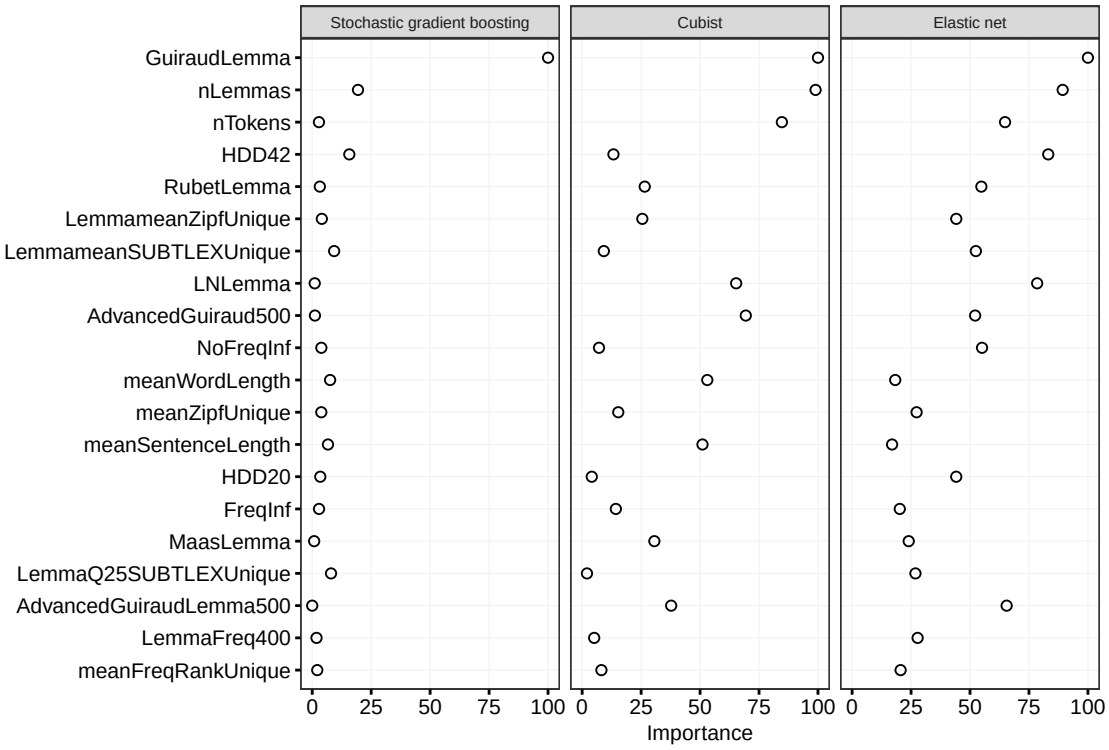

Figure 11.20: Variable importances of the 20 predictors with the highest mean rank in the three best performing predictive models.

Table 11.1: Summary of a generalised additive model fitted on the French training data.

Term	Type	Estimate / edf	Test statistic	p
Intercept	parametric	6.6	$t = 4.3$	< 0.001
Number of tokens (log2)	smooth	2.8	$F = 48$	< 0.001
MTLD 0.83 (log2)	parametric	0.22	$t = 3.6$	< 0.001
Evenness lemmata (sqrt)	smooth	3.4	$F = 11$	< 0.001
Mean Zipf, unique lemmata	smooth	4.5	$F = 15$	< 0.001
Disparity, lemmata	parametric	-3.0	$t = 1.7$	0.09
Dispersion, lemmata, $k = 20$ (sqrt)	parametric	0.16	$t = 0.5$	0.61

- Disparity: The disparity index computed with respect to lemmata.
- Dispersion: The dispersion index computed with respect to lemmata and $k = 20$. (Square-root transformed)

As Figure 11.21 shows, these predictors weren't entirely independent of one another. Particularly the evenness index showed strong correlations with other variables.

These predictors were fitted in a generalised additive model whose RMSE was estimated to be 0.871 ± 0.031 in block cross-validation, respectively.

The partial effects of the six predictors in this GAM are shown in Figure 11.22. Table 11.1 summarises the model numerically.

11.9 A single-predictor model

`GuiraudLemma` emerged as the single best predictor in the black boxes. A linear regression model with it as its sole predictor achieves a RSME of 0.881 in block cross-validation. The predictive function is:

$$\text{Predicted mean rating} = 1.44 + 0.81 \times \texttt{GuiraudLemma} \tag{11.1}$$

Since `GuiraudLemma` and `Guiraud` (type-based) were strongly correlated, a regression model with `Guiraud` as its sole predictor has virtually the same predictive accuracy. Its regression equation is:

$$\text{Predicted mean rating} = 1.53 + 0.78 \times \texttt{Guiraud} \tag{11.2}$$

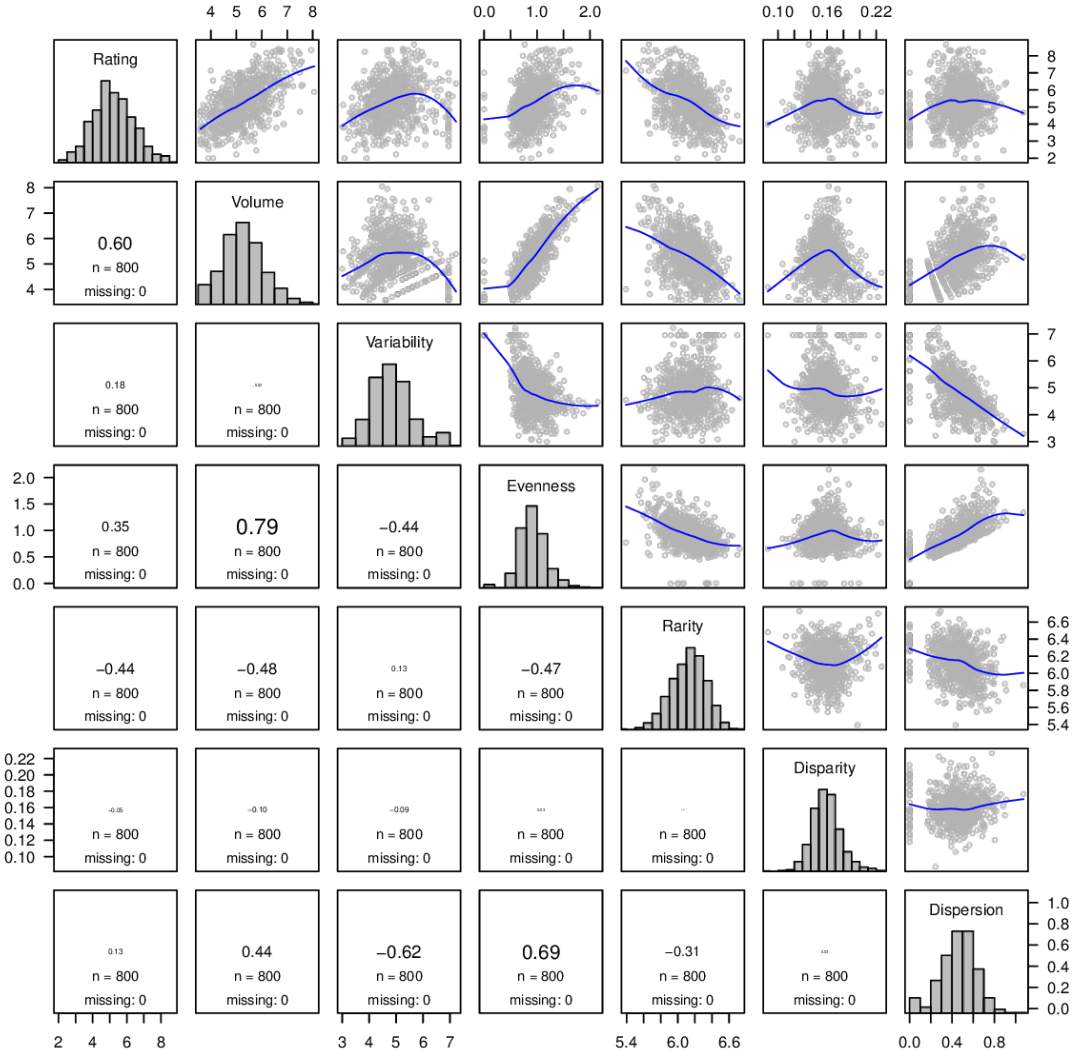

Figure 11.21: Bivariate relationships between the six predictors in the generalised additive model predicting the French ratings. The numbers in the bottom triangle are Pearson correlations.

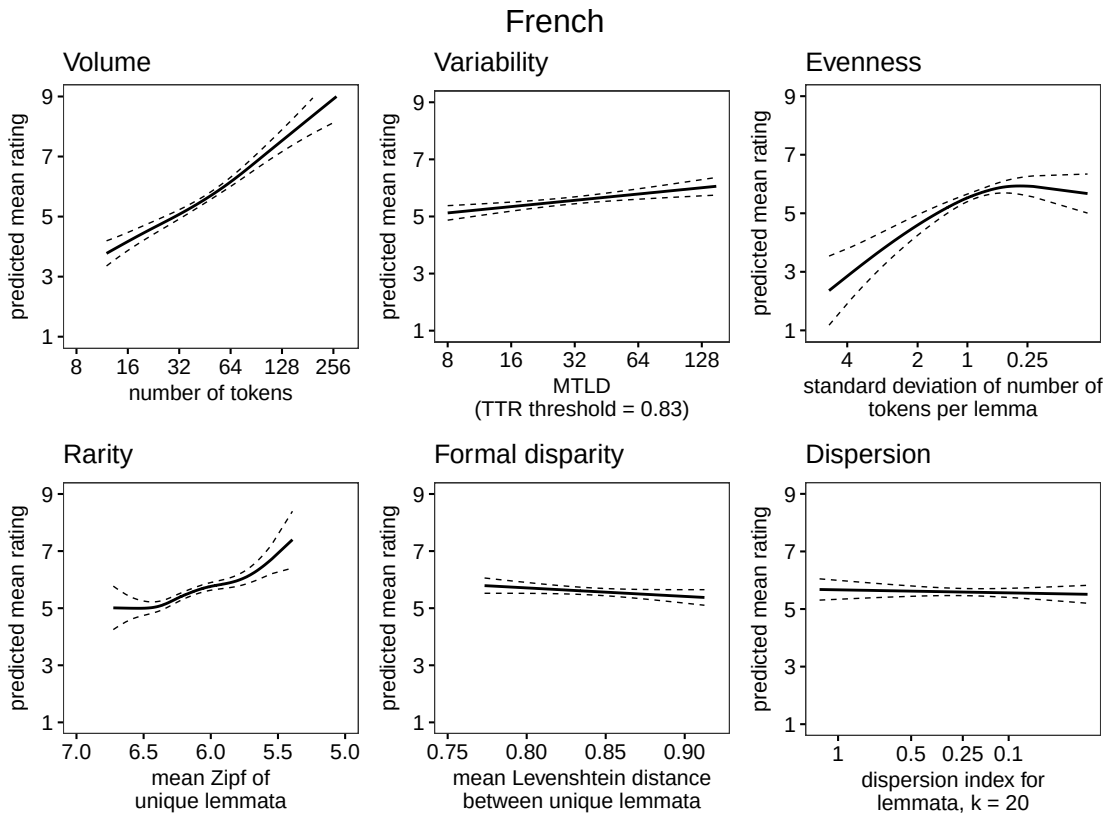

Figure 11.22: Partial effects of a generalised additive model fitted on the French training data using six predictors corresponding to Jarvis' 6 dimensions. The dashed lines delineate approximate 95% point-wise confidence bands. The x-axes are arranged such that value to the right indicate more volume, variability etc.

11.10 Comparison of the three approaches

For the sake of completeness, the predictive accuracy of the three approaches (black-box stacking; 6-predictor GAM; Guiraud-based regression) was directly compared using a series of paired t-tests ran on the 16 cross-validation estimates. (Paired t-tests are appropriate as the same cross-validation folds were used for each approach.) All comparisons are RMSE-based:

- Black-box versus 6 dimensions: Black-box 0.045 points better on average ($t(15) = 6.7$, $p < 0.001$).
- Black-box versus Guiraud: Black-box 0.055 points better on average ($t(15) = 5.0$, $p < 0.001$).
- 6 dimensions versus Guiraud: 6 dimensions 0.010 points better on average ($t(15) = 0.9$, $p = 0.36$).

Chapter 12

Predictive modelling: German

12.1 Data splitting

Text sets G, F, I and N were randomly selected and together constituted the test set. These 204 observations were not looked at during data exploration and model tuning/selection.

12.2 Predictor transformation

Many predictor variables were right-skewed so that a Yeo–Johnson transformation (Yeo & Johnson, 2000) was applied to the entire predictor set. Of the 154 predictors, 152 were transformed in order to get a more symmetrical distribution. The predictors were subsequently centred at their mean in the training data and scaled using their standard deviation in the training data.

When tuning models, predictor transformations were be effected during cross-validation.

12.3 Bivariate relationship between ratings and predictors in training data

The bivariate relationships between the mean ratings and the transformed predictors as well as among the transformed predictors themselves were inspected. In the scatterplot matrices that follow, the distribution and name of the variables are shown on the main diagonal. The upper triangle shows scatterplots and a LOESS fit (in blue). The bottom triangle shows the squared correlation between the y variable and the LOESS fit ($\hat{y}$): values close to 1 indicate that one variable can be entirely expressed as a (linear or nonlinear)

function of the other; values close to 0 indicate that the variables are orthogonal to one another.

Since the number of predictors is too large to show in one plot, several scatterplot matrices are shown. On the basis of these scatterplot matrices, highly correlated variables were identified and removed. While the German data were analysed in their own right, the same predictors were removed on the grounds of their intercorrelations as for the French data.

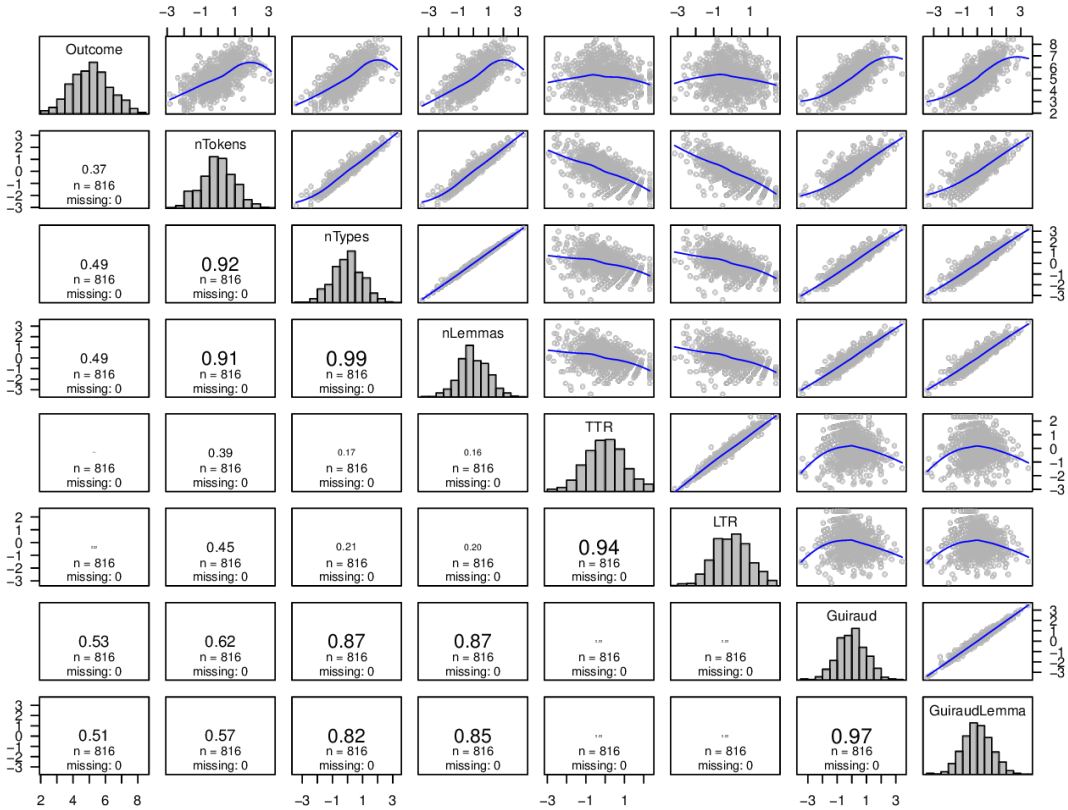

Figure 12.1: Intercorrelation between German predictors (1): Tokens, types and lemmata. `nTypes`, `TTR` and `Guiraud` were removed because of their strong intercorrelations with the other variables; `nTokens` was retained for now, despite its high intercorrelations.

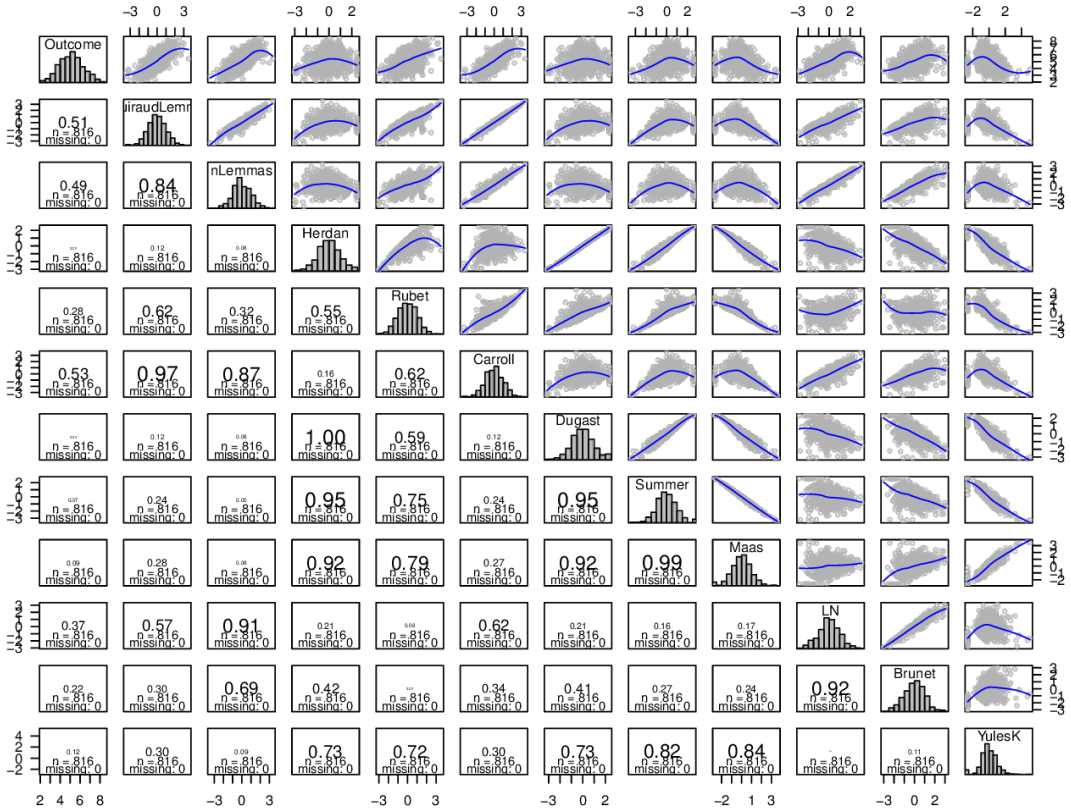

Figure 12.2: Intercorrelation between German predictors (2): TTR variations. `Carroll`, `Dugast`, `Summer` and `Brunet` were removed because of their strong intercorrelations with the other variables.

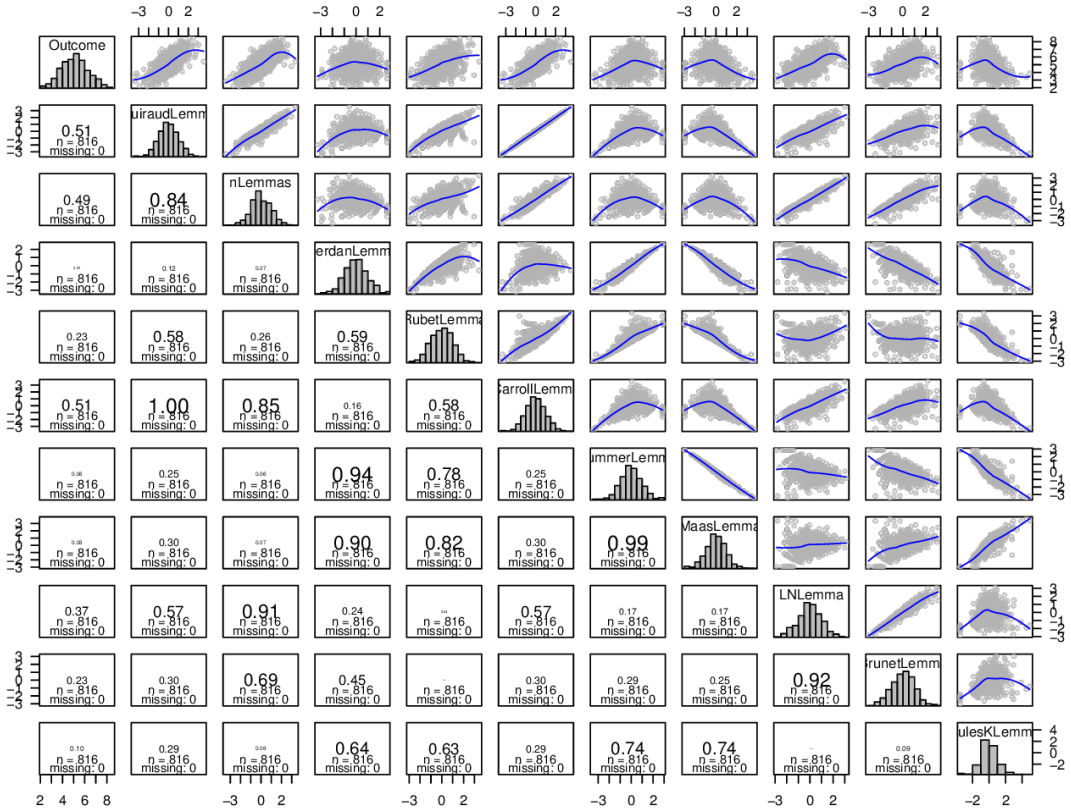

Figure 12.3: Intercorrelation between German predictors (3): TTR variations. `CarrollLemma`, `DugastLemma`, `SummerLemma` and `BrunetLemma` were removed because of their strong intercorrelations with the other variables.

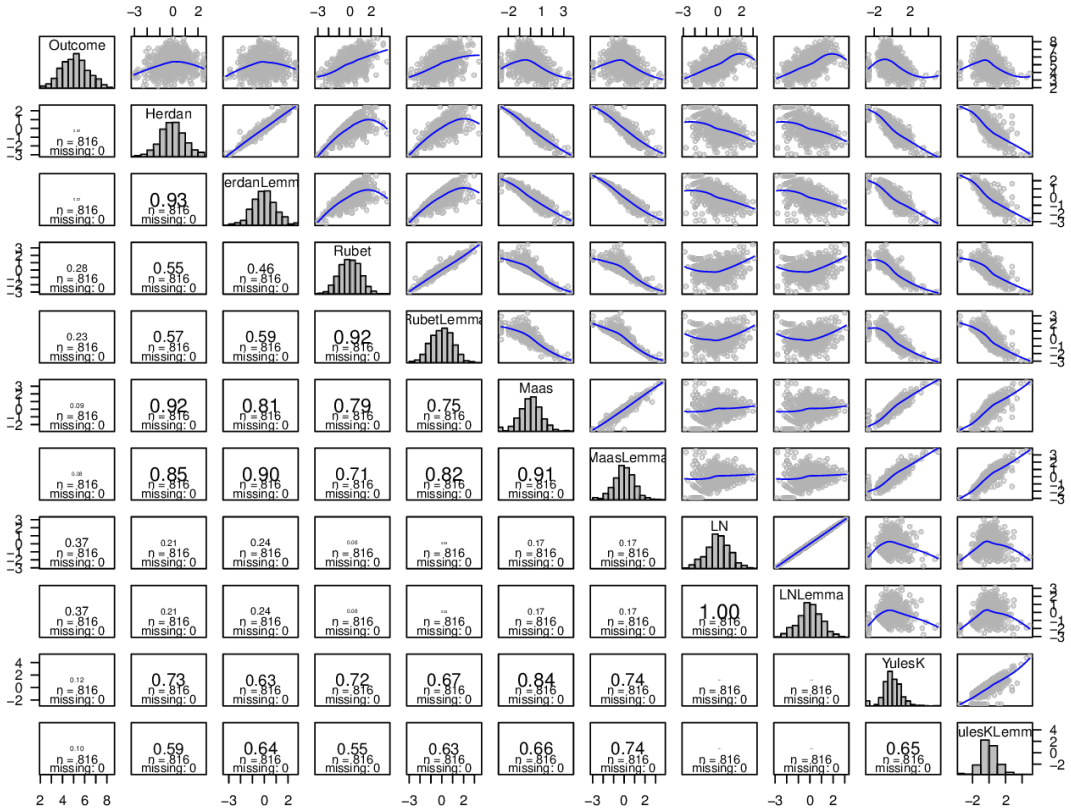

Figure 12.4: Intercorrelation between German predictors (4): TTR vs. LTR variations. `Herdan`, `Rubet`, `Maas` and `LN` were removed because of their strong intercorrelations with the other variables.

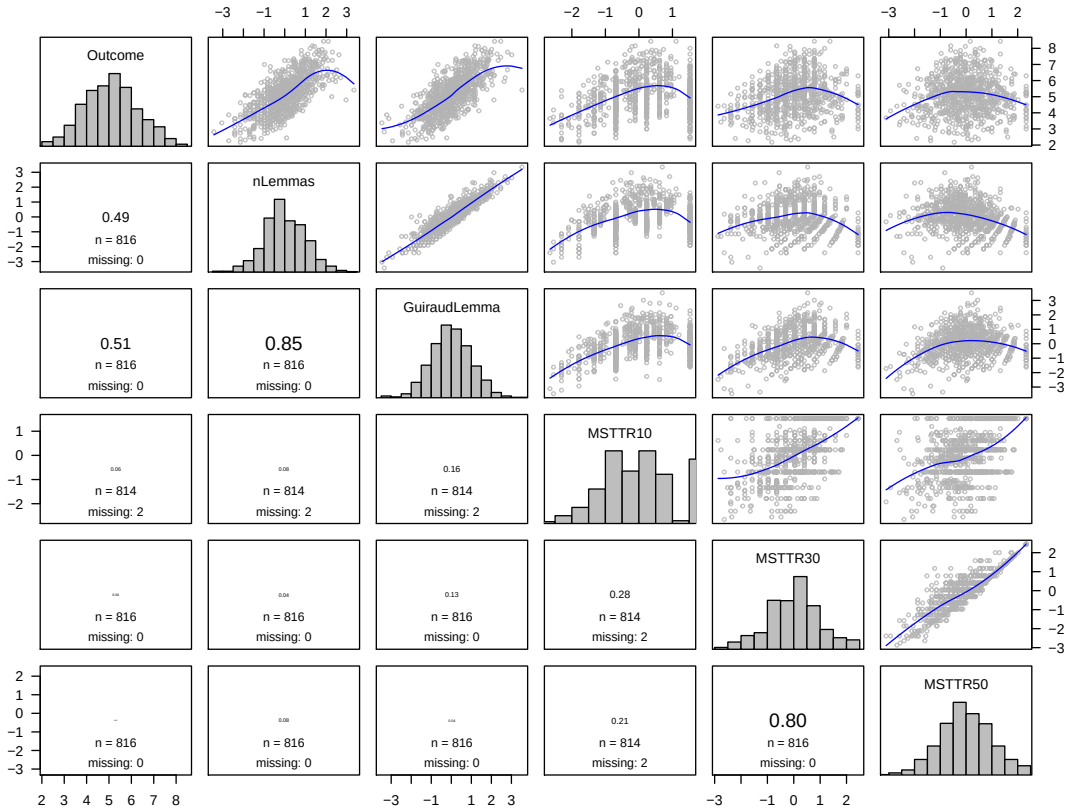

Figure 12.5: Intercorrelation between German predictors (5): MSTTR. No variables were removed because of their strong intercorrelations with the other variables.

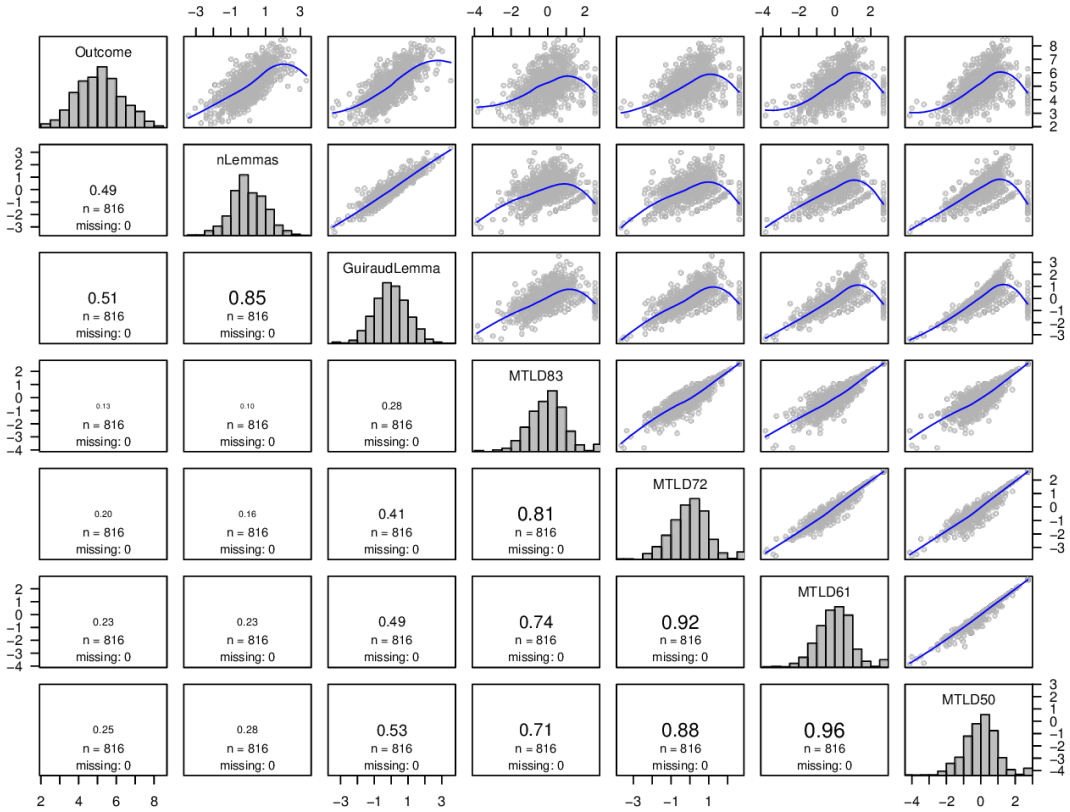

Figure 12.6: Intercorrelation between German predictors (6): MTLD. `MTLD61` and `MTLD72` were removed because of their strong intercorrelations with the other variables.

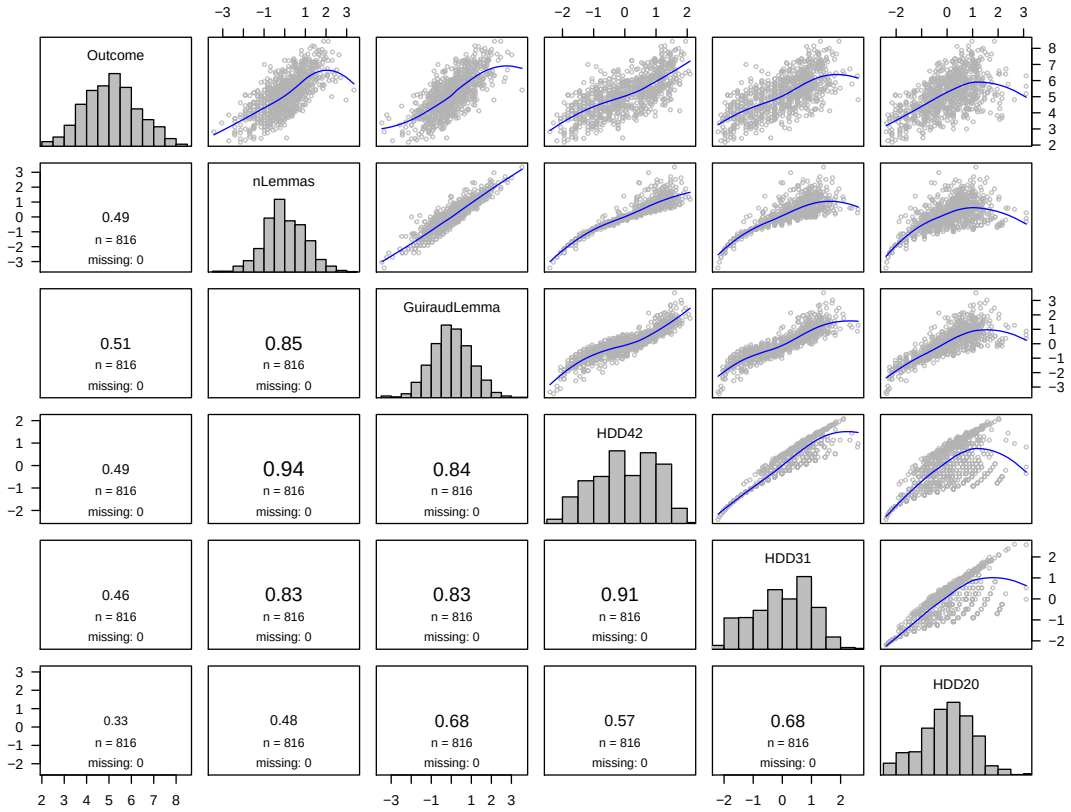

Figure 12.7: Intercorrelation between German predictors (7): HDD. `MTLD31` was removed because of their strong intercorrelations with the other variables.

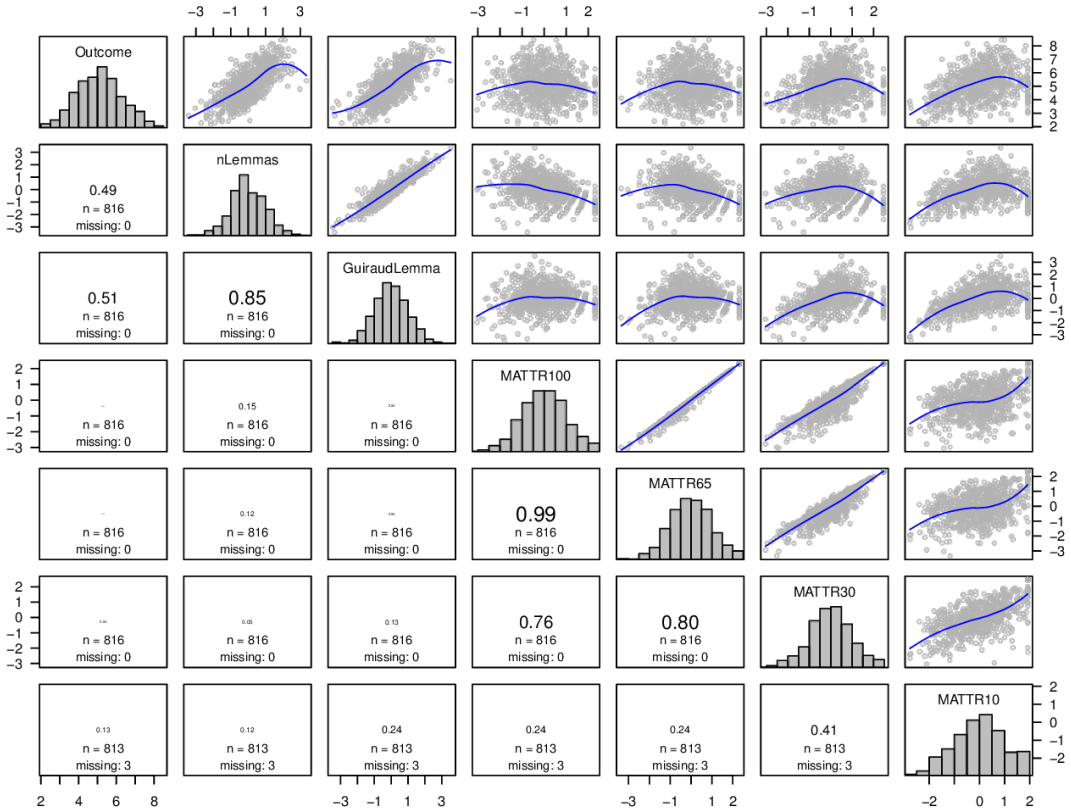

Figure 12.8: Intercorrelation between German predictors (8): MATTR. `MATTR65` was removed because of their strong intercorrelations with the other variables.

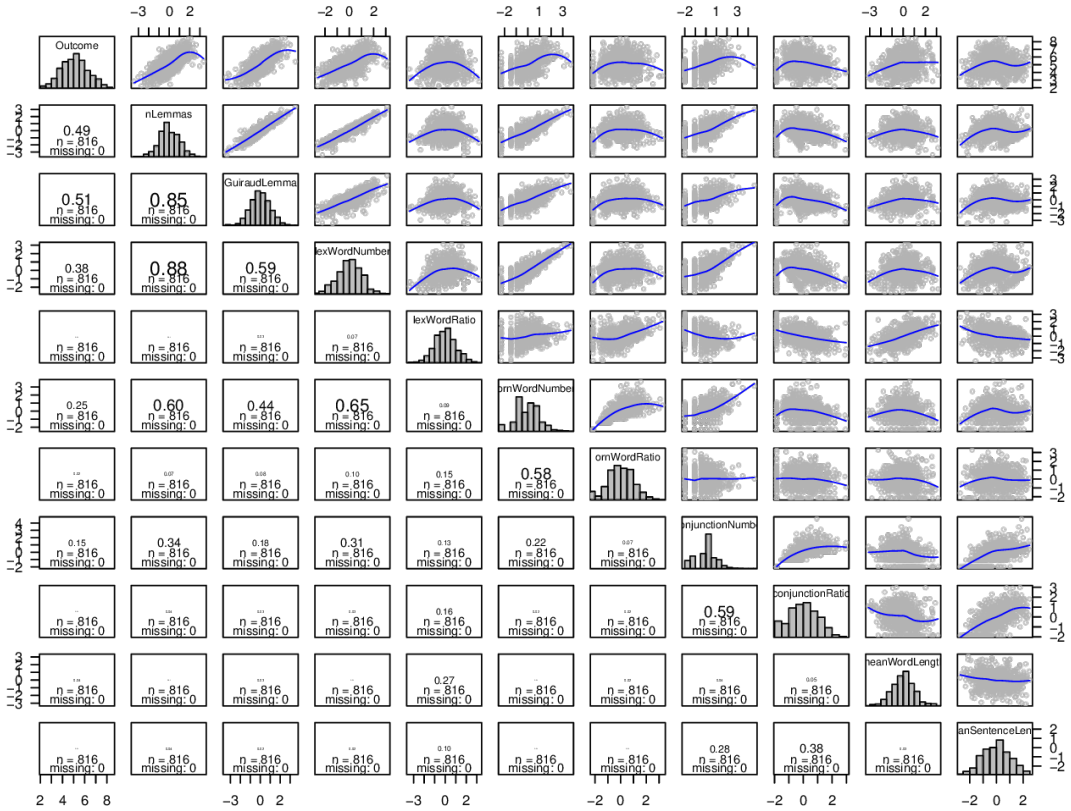

Figure 12.9: Intercorrelation between German predictors (9): Lexical and syntactic complexity. `lexWordNumber` was removed because of their strong intercorrelations with the other variables.

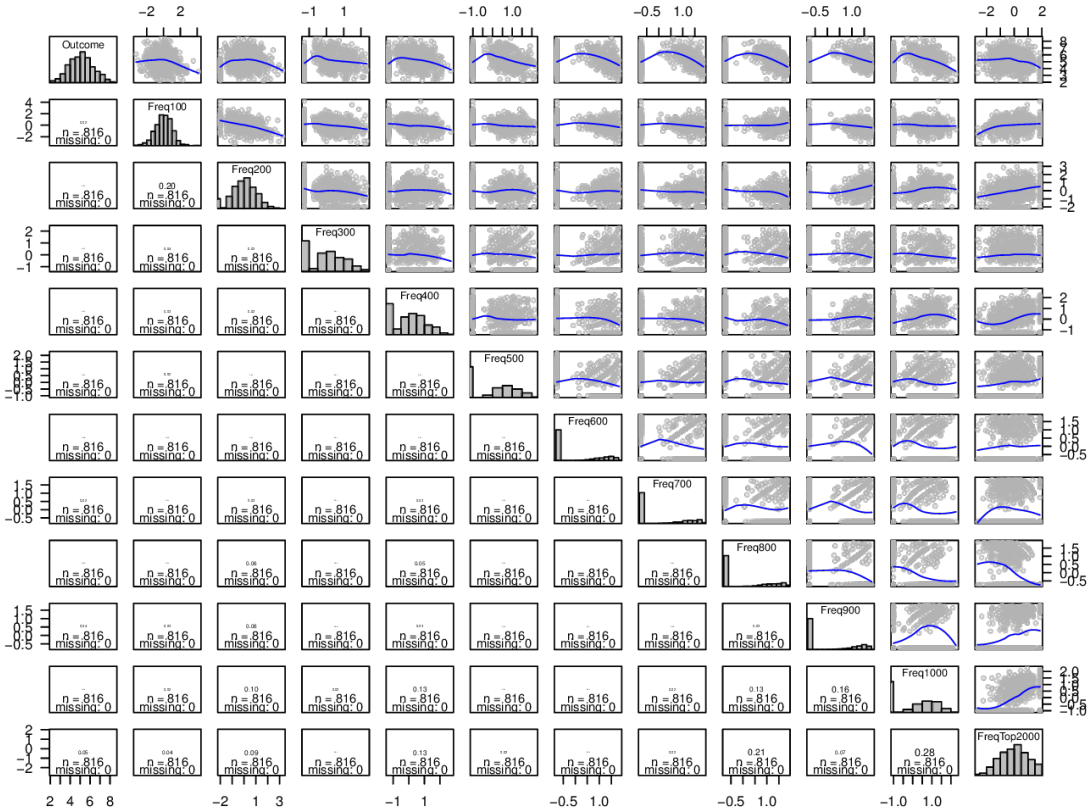

Figure 12.10: Intercorrelation between German predictors (10): Top frequency bands. No variable was removed because of their strong intercorrelations with the other variables.

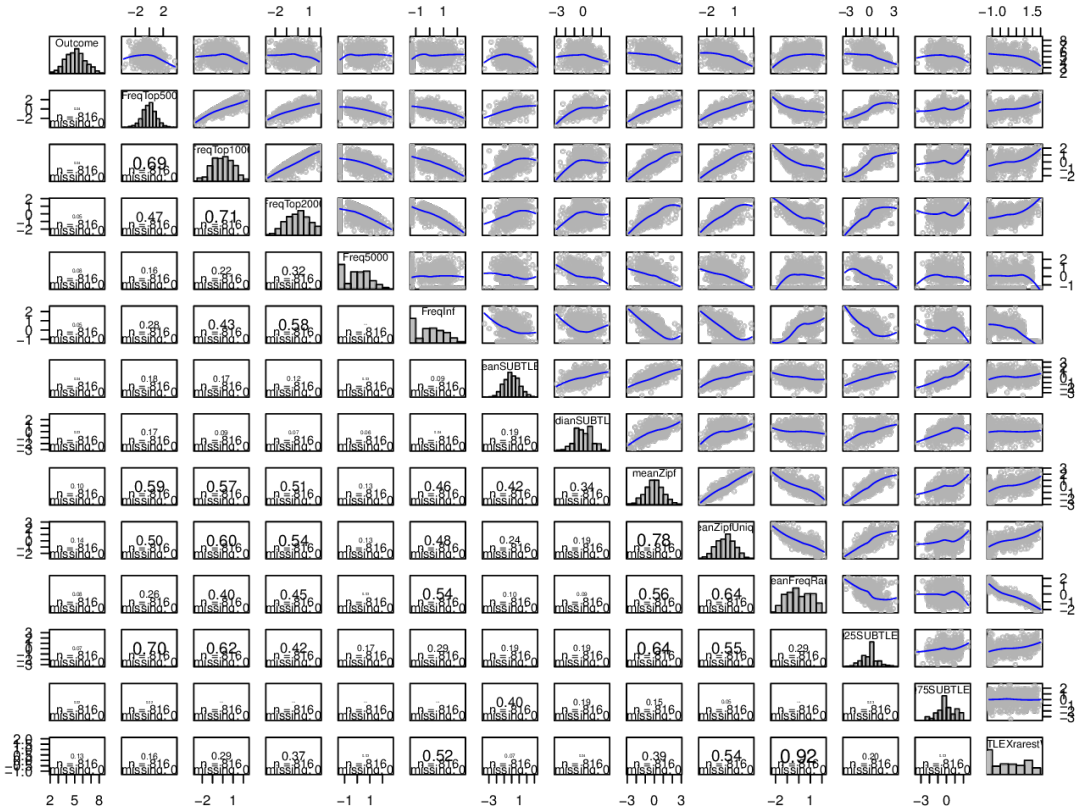

Figure 12.11: Intercorrelation between German predictors (11): Frequency summaries (token- and type-based). No variable was removed because of their strong intercorrelations with the other variables.

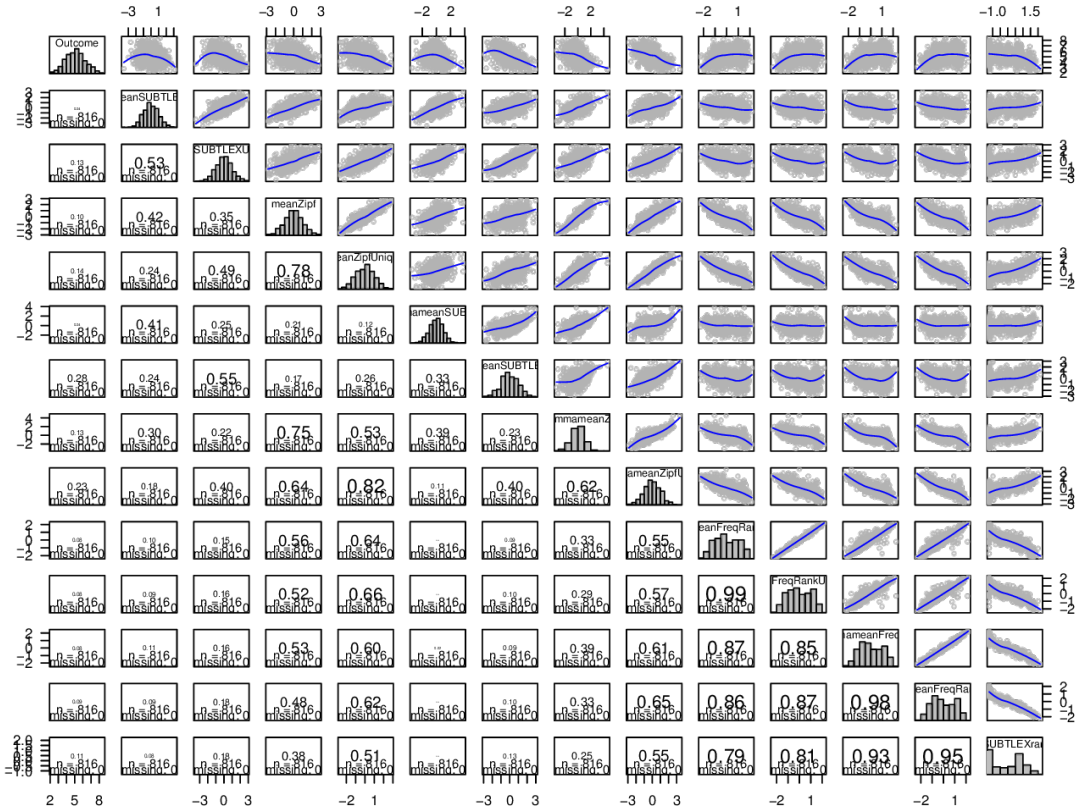

Figure 12.12: Intercorrelation between German predictors (12): Frequency summaries (lemma-based). `meanFreqRank` and `meanFreqRankUnique` were removed because of their strong intercorrelations with the other variables.

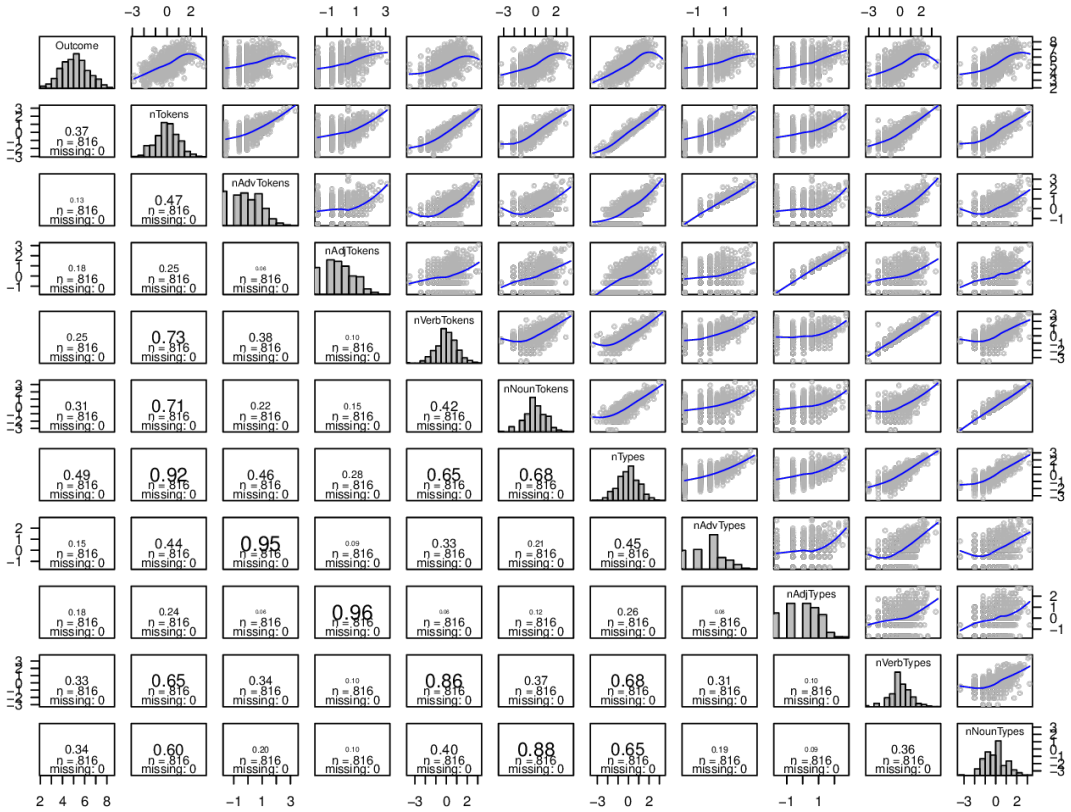

Figure 12.13: Intercorrelation between German predictors (13): Number of types and tokens by POS.

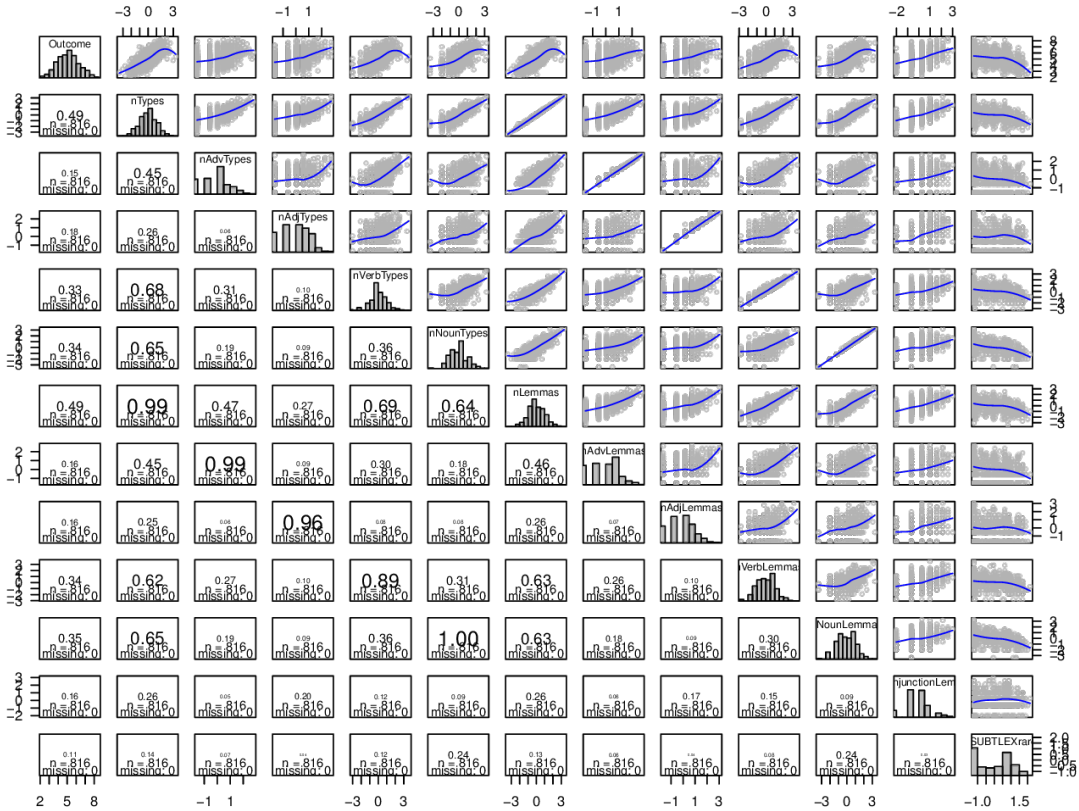

Figure 12.14: Intercorrelation between German predictors (14): Number of lemmas by POS. `nNounTokens`, `nNounTypes`, `nVerbTokens`, `nVerbTypes`, `nAdjTokens`, `nAdjTypes`, `nAdvTokens` and `nAdjTypes` were removed because of their strong intercorrelations with the other variables.

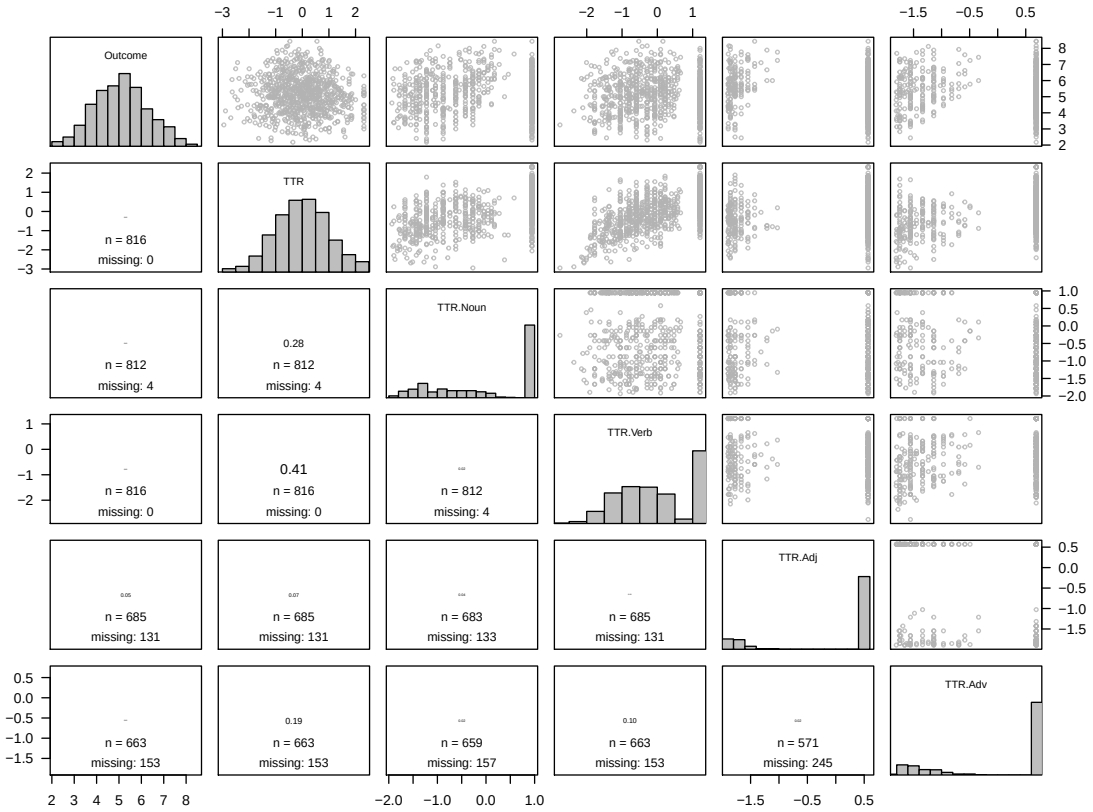

Figure 12.15: Intercorrelation between German predictors (15): TTR by part of speech. `TTR.Noun`, `TTR.Verb`, `TTR.Adj` and `TTR.Adv` removed due to missing values.

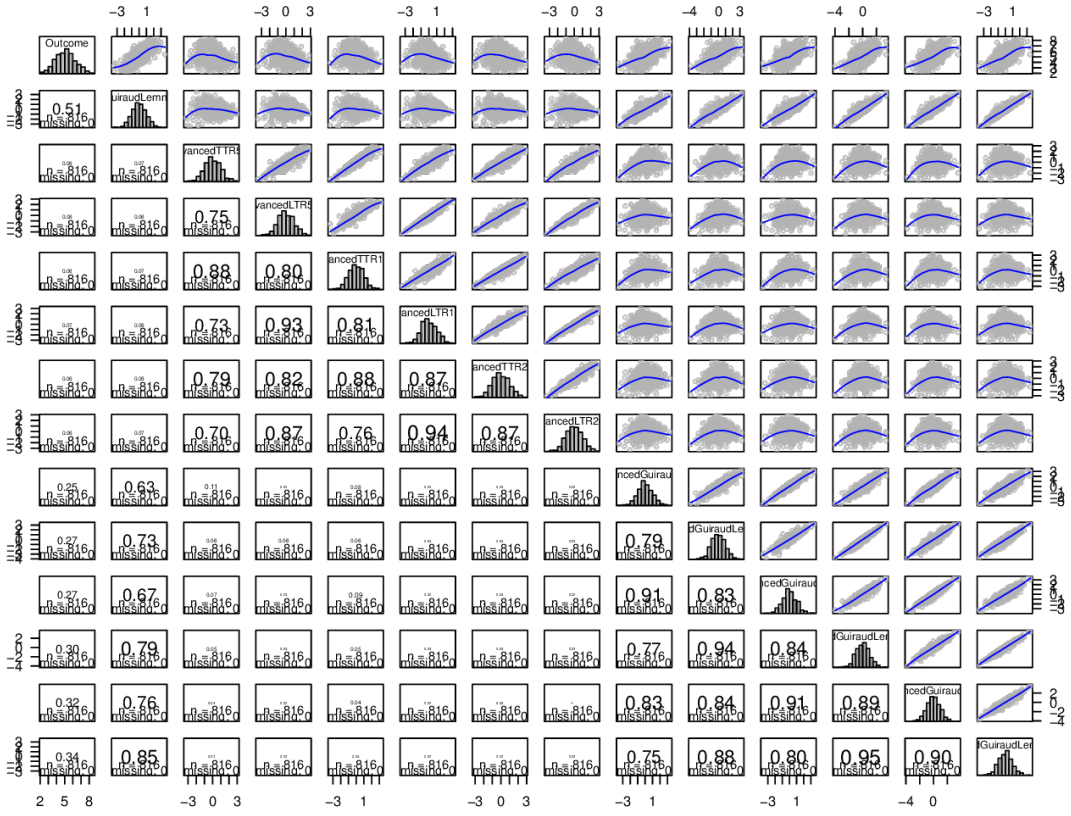

Figure 12.16: Intercorrelation between German predictors (16): Advanced Guiraud and TTR. `AdvancedLTR1000`, `AdvancedTTR1000`, `AdvancedGuiraudLemma1000`, `AdvancedGuiraud1000`, `AdvancedTTR2000`, `AdvancedGuiraudLemma2000` and `AdvancedGuiraud2000` removed due to high intercorrelations.

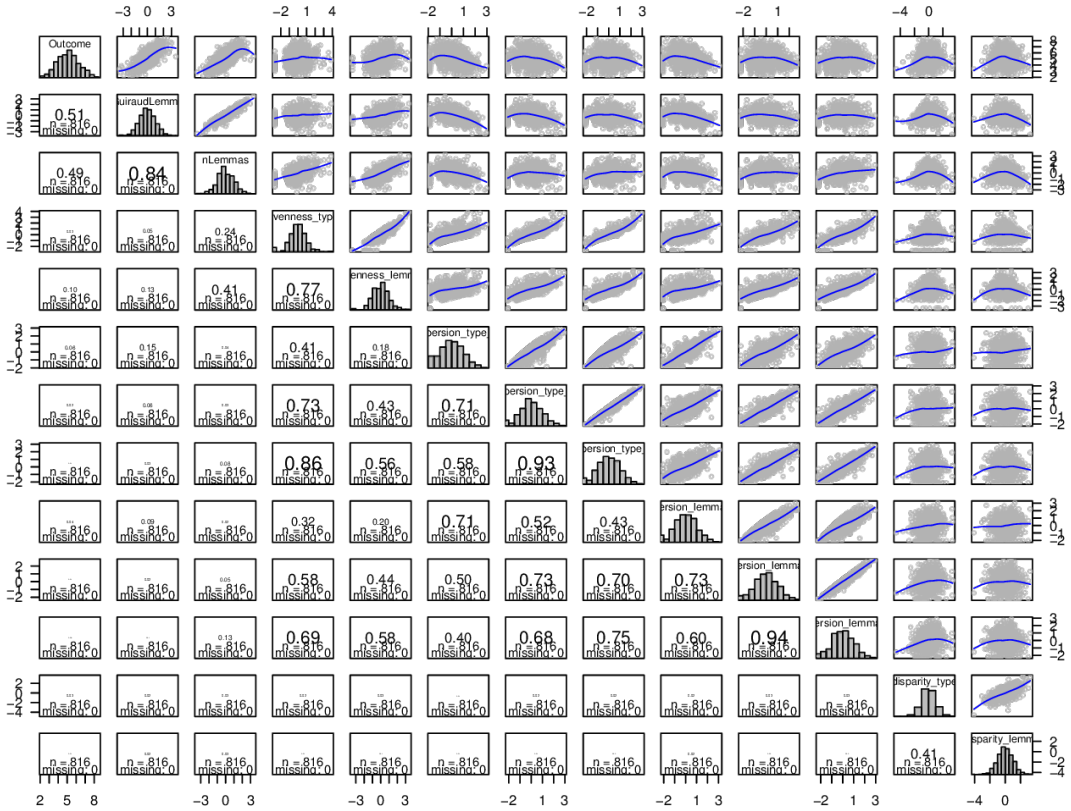

Figure 12.17: Intercorrelation between German predictors (16): Evenness, disparity and dispersion. `dispersion_type_30` and `dispersion_lemma_30` removed due to high intercorrelations.

12.4 Model performance in cross-validation

Several different algorithms were fitted to the training data and tuned using block cross-validation. For most models, the data were Yeo–Johnson transformed. Figure 12.18 shows the estimated predictive accuracy of 14 tuned models. The algorithm with the greatest predictive power was Cubist, with a mean RMSE of 0.709 . Cubist is followed by support vector machines and ridge regression with mean RMSE values of 0.710 and 0.712, respectively.

12.5 Model stacking

The out-of-fold predictions for all 14 models were extracted. The Pearson correlations between the out-of-fold predictions of the different models varied between 0.81 and 0.9998 with a median correlation of 0.942. These strong correlations suggest that any gain in predictive accuracy from model stacking will be small.

The out-of-fold predictions of each model were used as predictors in a principal component regression model. This is a linear regression model for which principal component analysis was first applied to the predictors. The predictive accuracy of this model was assessed block cross-validation The RMSE was estimated to be 0.704. This represents an ever so slight increase over the single best model.

12.6 Does predictive accuracy depend on text length?

Figure 12.19 shows the correlations between text length and the out-of-fold predictions and residuals according to the stacked model. While longer texts are predicted to have better ratings (left), the variance of the residuals hardly varies according to text length (middle and right). In conclusion, the models' predictive accuracy is about equally as good for short as for longer texts.

12.7 Variable importance in top-2 models

Figure 12.20 shows the variable importances of the 20 most important predictors in the top 2 models (Cubist, SVM).[1] These values were extracted using `caret`'s `varImp()` function, and for each of the three models, the variables were rank-ordered from less to more important. The twenty variables with the highest mean rank across the two models are shown in the plot.

[1]The third, ridge regression, produces identical variable importances to SVM.

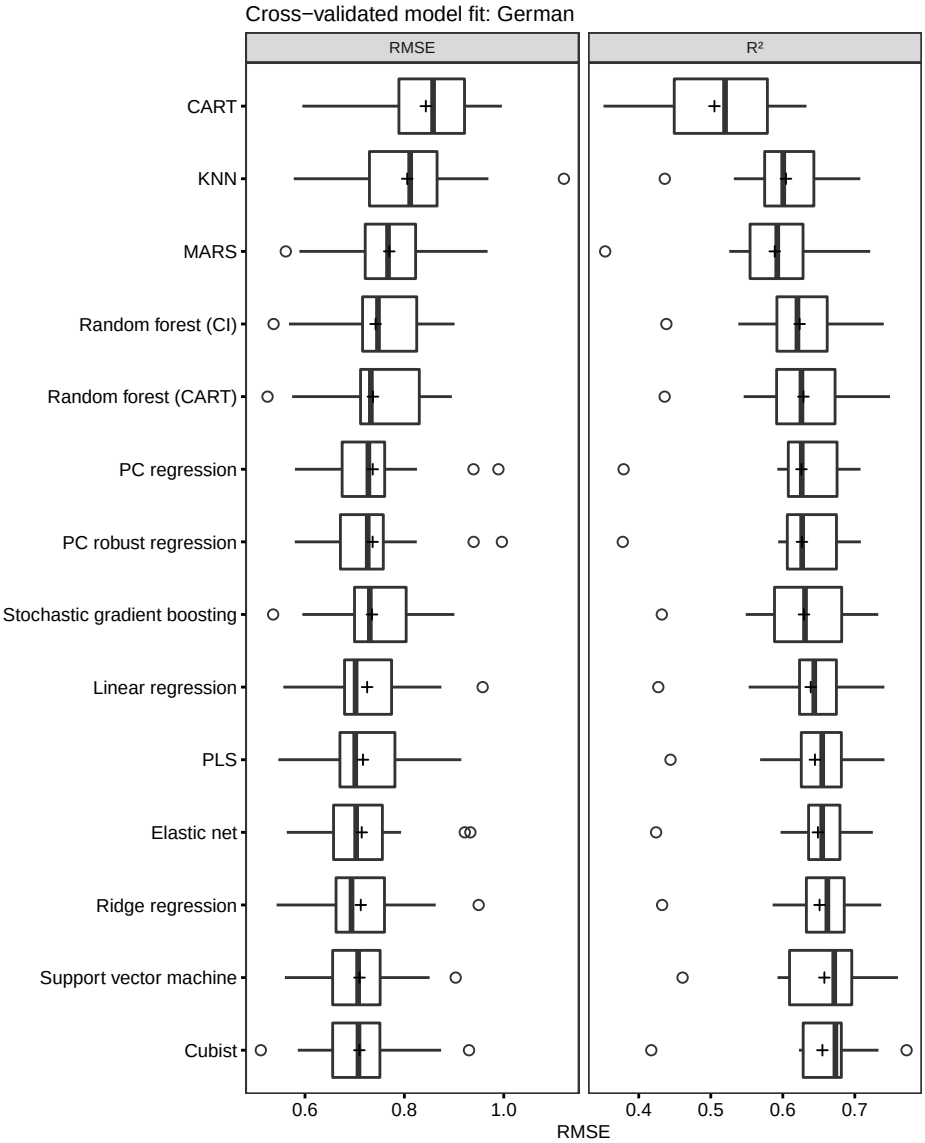

Figure 12.18: Performance of 14 tuned predictive models for the German training data in block cross-validation (with 16 blocks). The crosses mark the mean of each distribution.

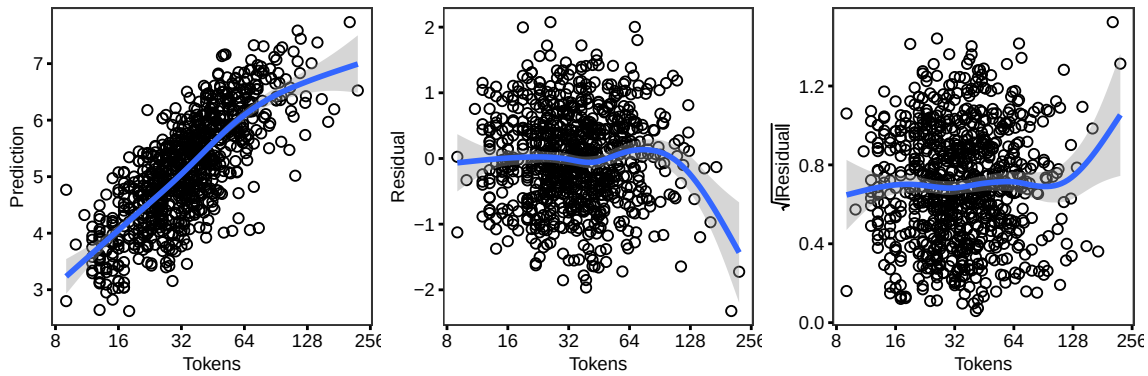

Figure 12.19: *Left:* Text length and out-of-fold prediction according to the stacked model for all 816 training texts. *Middle:* Text length and residuals (actual value − average out-of-fold prediction). *Right:* Text length and root absolute residuals.

The single most important variable in both models is `GuiraudLemma`, that is, $\frac{\text{number of lemmata}}{\sqrt{\text{number of tokens}}}$. Another important variable in the three models is the number of lemmata occurring in the texts. After these two variables, the models diverge noticeably.

12.8 A 6-dimensional model

A more interpretable 6-dimensional model based on the framework proposed by Jarvis (2013a,b) was fitted to the training data. This model contained the following predictors:

- Volume: The number of tokens. (Log-transformed)
- Variability: MTLD with a TTR setting of 0.83. The MTLD was chosen as it is not systematically affected by the texts' length. (Log-transformed)
- Evenness: The lemma-based evenness index. (Square-root transformed)
- Rarity: The mean Zipf value of the unique lemmata occurring in the texts. Other rarity indices were tried and yield fairly similar results.
- Disparity: The disparity index computed with respect to lemmata.
- Dispersion: The dispersion index computed with respect to lemmata and $k = 20$. (Square-root transformed)

As Figure 12.21 shows, these predictors weren't entirely independent of one another. Particularly the evenness index showed strong correlations with other variables.

These predictors were fitted in a generalised additive model whose RMSE was estimated

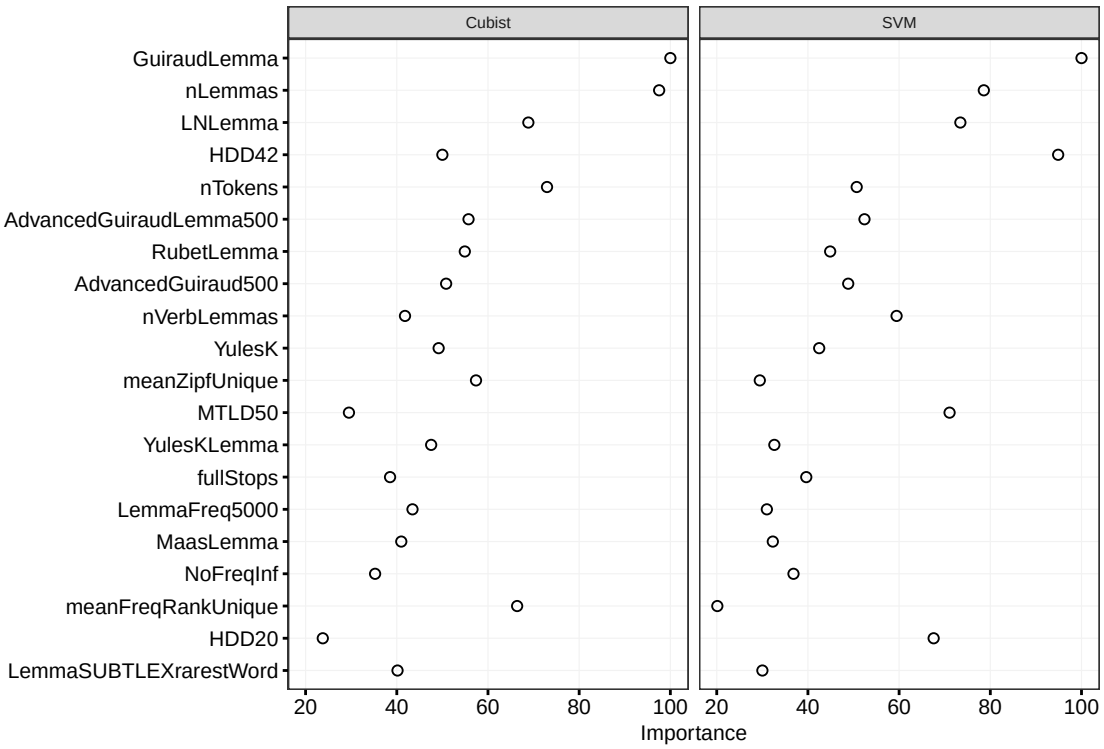

Figure 12.20: Variable importances of the 20 predictors with the highest mean rank in the two best performing predictive models. (The third, ridge regression, produces not similar but identical variable importances to SVM.)

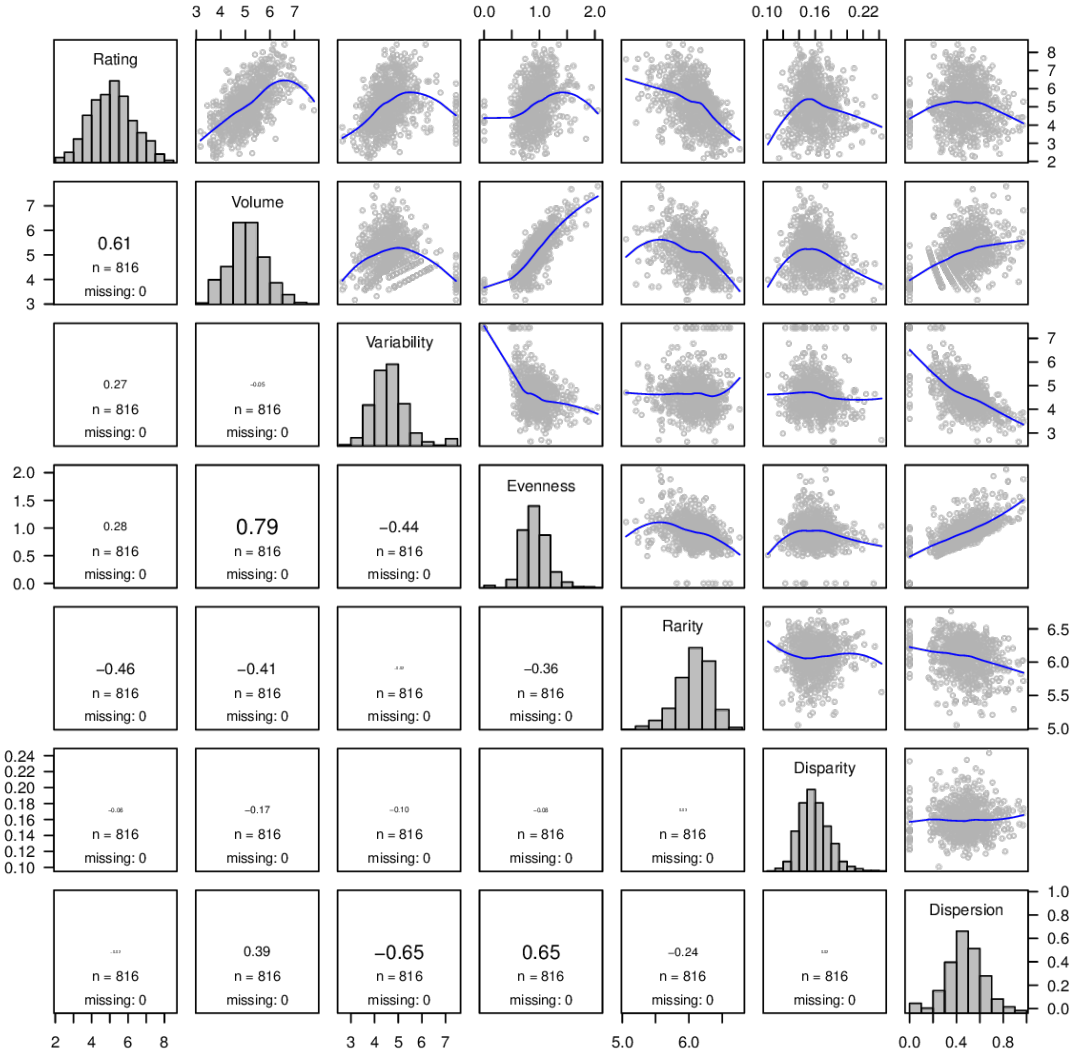

Figure 12.21: Bivariate relationships between the six predictors in the generalised additive model predicting the German ratings. The numbers in the bottom triangle are Pearson correlations.

to be 0.756 ± 0.031 in block cross-validation, respectively.

The partial effects of the six predictors in this GAM are shown in Figure 12.22. Table 12.1 summarises the model numerically.

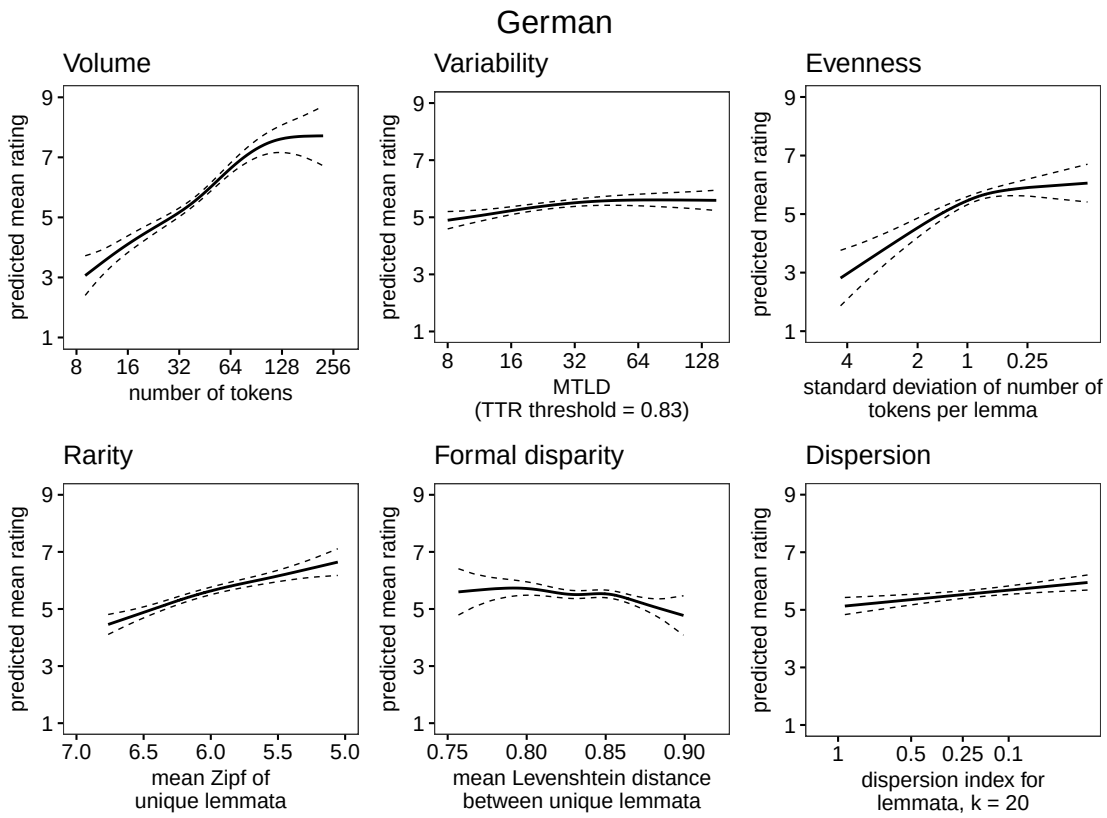

Figure 12.22: Partial effects of a generalised additive model fitted on the German training data using six predictors corresponding to Jarvis' 6 dimensions.

12.9 A single-predictor model

`GuiraudLemma` emerged as the single best predictor in the black boxes. A linear regression model with it as its sole predictor achieves a RSME of 0.82 in block cross-validation. The predictive function is:

$$\text{Predicted mean rating} = 0.77 + 1.03 \times \texttt{GuiraudLemma} \tag{12.1}$$

Since `GuiraudLemma` and `Guiraud` (type-based) were strongly correlated, a regression model

Table 12.1: Summary of a generalised additive model fitted on the German training data.

Term	Type	Estimate / edf	Test statistic	p
Intercept	parametric	5.53	$t = 44$	< 0.001
Number of tokens (log2)	smooth	4.8	$F = 46$	< 0.001
MTLD 0.83 (log2)	smooth	2.3	$F = 5.8$	< 0.001
Evenness lemmata (sqrt)	smooth	3.0	$F = 13$	< 0.001
Mean Zipf, unique lemmata	smooth	2.1	$F = 47$	< 0.001
Disparity, lemmata	smooth	4.2	$F = 3.5$	0.003
Dispersion, lemmata, $k = 20$ (sqrt)	parametric	-0.84	$t = 3.2$	0.001

with `Guiraud` as its sole predictor has virtually the same predictive accuracy. Its regression equation is:

$$\text{Predicted mean rating} = 0.95 + 0.95 \times \texttt{Guiraud} \tag{12.2}$$

12.10 Comparison of the three approaches

The predictive accuracy of the three approaches was directly compared using a series of paired t-tests ran on the 16 cross-validation estimates. All comparisons are RMSE-based:

- Black-box versus 6 dimensions: Black-box 0.052 points better on average ($t(15) = 4.4$, $p < 0.001$).
- Black-box versus Guiraud: Black-box 0.117 points better on average ($t(15) = 10$, $p < 0.001$).
- 6 dimensions versus Guiraud: 6 dimensions 0.065 points better on average ($t(15) = 4.0$, $p = 0.001$).

Chapter 13

Predictive modelling: Portuguese

13.1 Data splitting

Text sets C, J, K and M were randomly selected and together constituted the test set. These 208 observations were not looked at during data exploration and model tuning/selection.

13.2 Predictor transformation

Many predictor variables were right-skewed so that a Yeo–Johnson transformation (Yeo & Johnson, 2000) was applied to the entire predictor set. Of the 154 predictors, 153 were transformed in order to get a more symmetrical distribution. The predictors were subsequently centred at their mean in the training data and scaled using their standard deviation in the training data.

When tuning models, predictor transformations were be effected during cross-validation.

13.3 Bivariate relationship between ratings and predictors in training data

The bivariate relationships between the mean ratings and the transformed predictors as well as among the transformed predictors themselves were inspected. In the scatterplot matrices that follow, the distribution and name of the variables are shown on the main diagonal. The upper triangle shows scatterplots and a LOESS fit (in blue). The bottom triangle shows the squared correlation between the y variable and the LOESS fit ($\hat{y}$): values close to 1 indicate that one variable can be entirely expressed as a (linear or nonlinear)

function of the other; values close to 0 indicate that the variables are orthogonal to one another.

Since the number of predictors is too large to show in one plot, several scatterplot matrices are shown. On the basis of these scatterplot matrices, highly correlated variables were identified and removed. While the Portuguese data were analysed in their own right, the same predictors were removed on the grounds of their intercorrelations as for the French and German data, with the addition of one variable, viz., `LNLemma`.

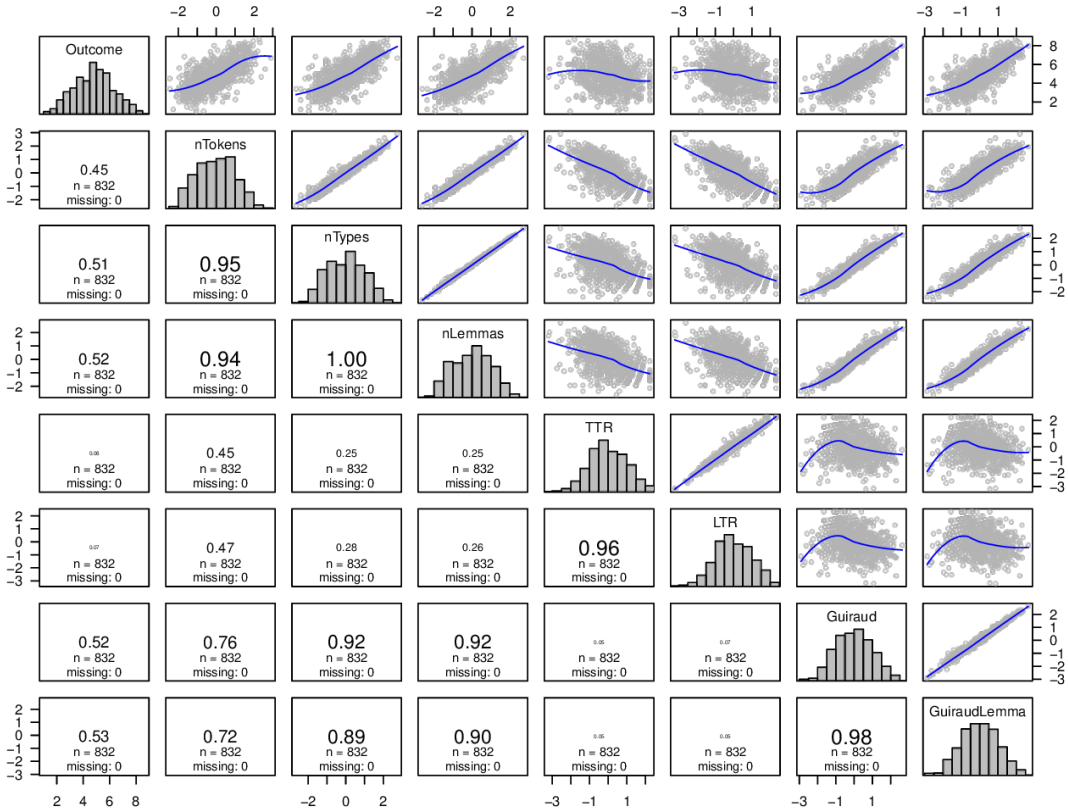

Figure 13.1: Intercorrelation between Portuguese predictors (1): Tokens, types and lemmata. `nTypes`, `TTR` and `Guiraud` were removed because of their strong intercorrelations with the other variables; `nTokens` was retained for now, despite its high intercorrelations.

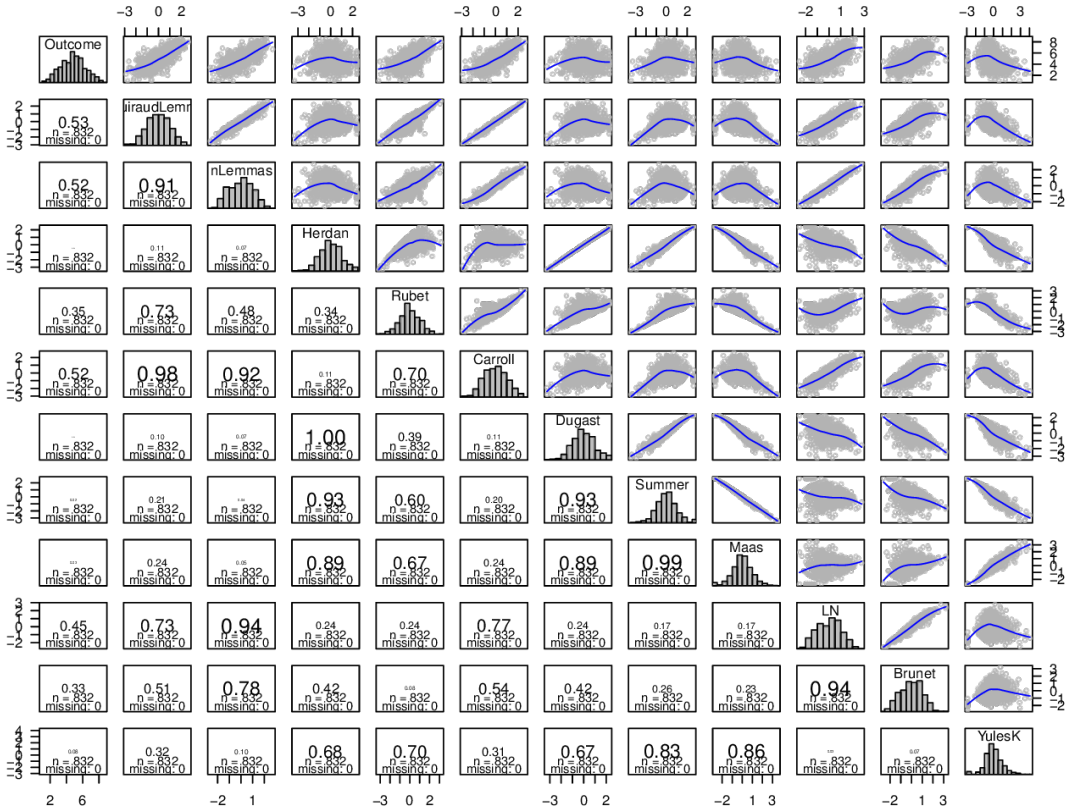

Figure 13.2: Intercorrelation between Portuguese predictors (2): TTR variations. `Carroll`, `Dugast`, `Summer` and `Brunet` were removed because of their strong intercorrelations with the other variables.

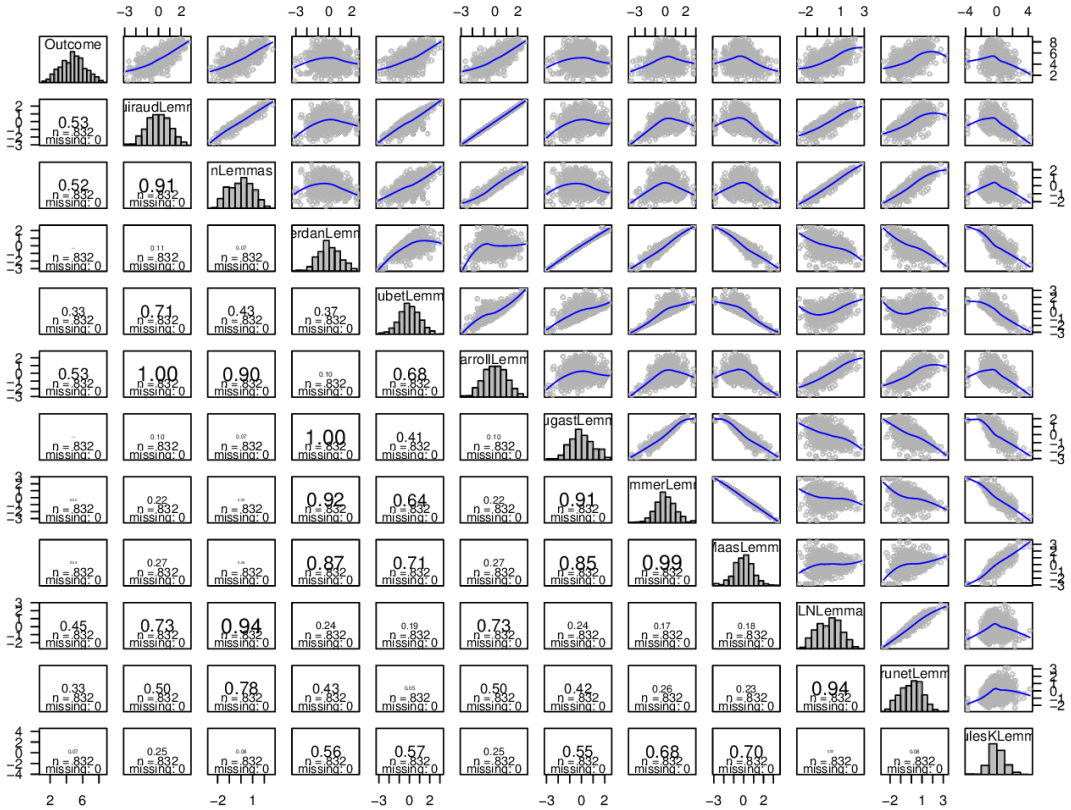

Figure 13.3: Intercorrelation between Portuguese predictors (3): TTR variations. `CarrollLemma`, `DugastLemma`, `SummerLemma`, `LNLemma` and `BrunetLemma` were removed because of their strong intercorrelations with the other variables.

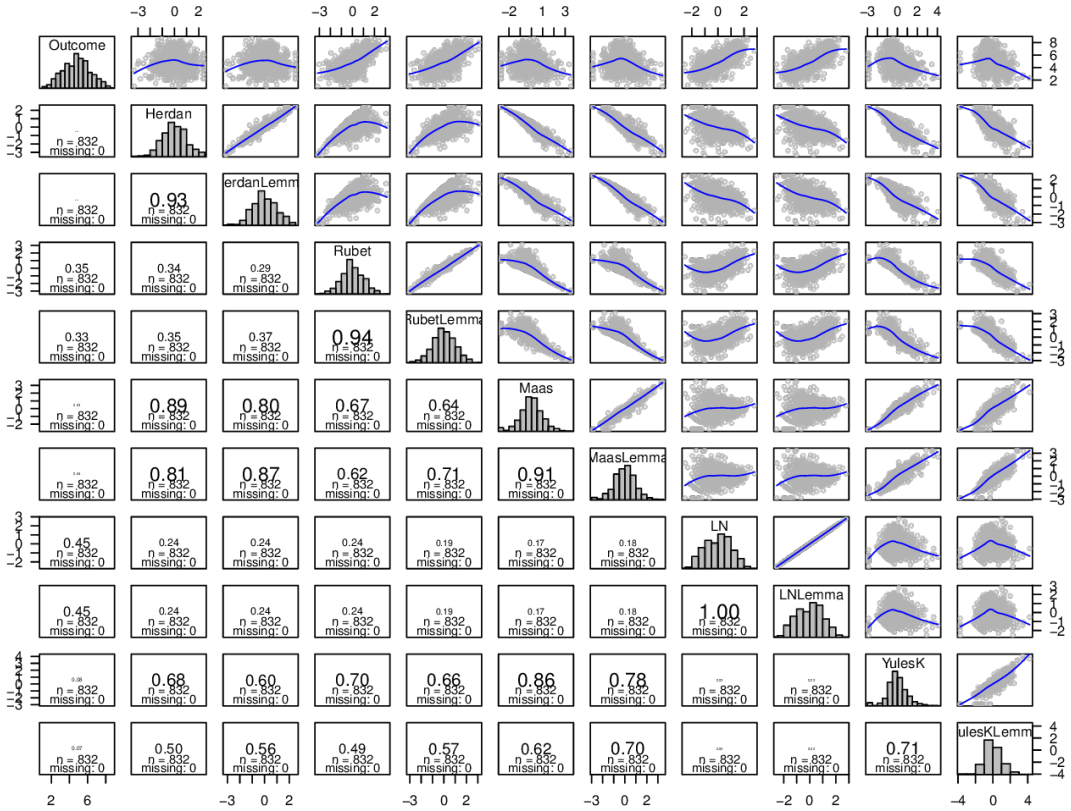

Figure 13.4: Intercorrelation between Portuguese predictors (4): TTR vs. LTR variations. `Herdan`, `Rubet`, `Maas` and `LN` were removed because of their strong intercorrelations with the other variables.

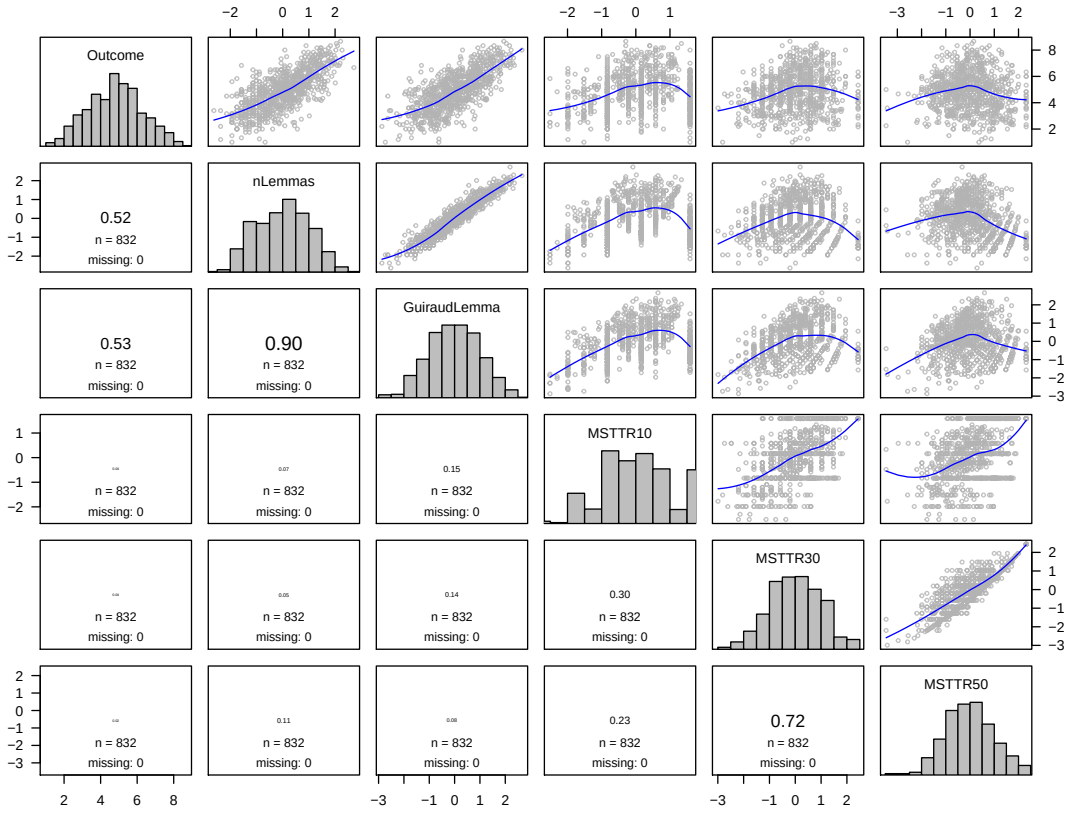

Figure 13.5: Intercorrelation between Portuguese predictors (5): MSTTR. No variables were removed because of their strong intercorrelations with the other variables.

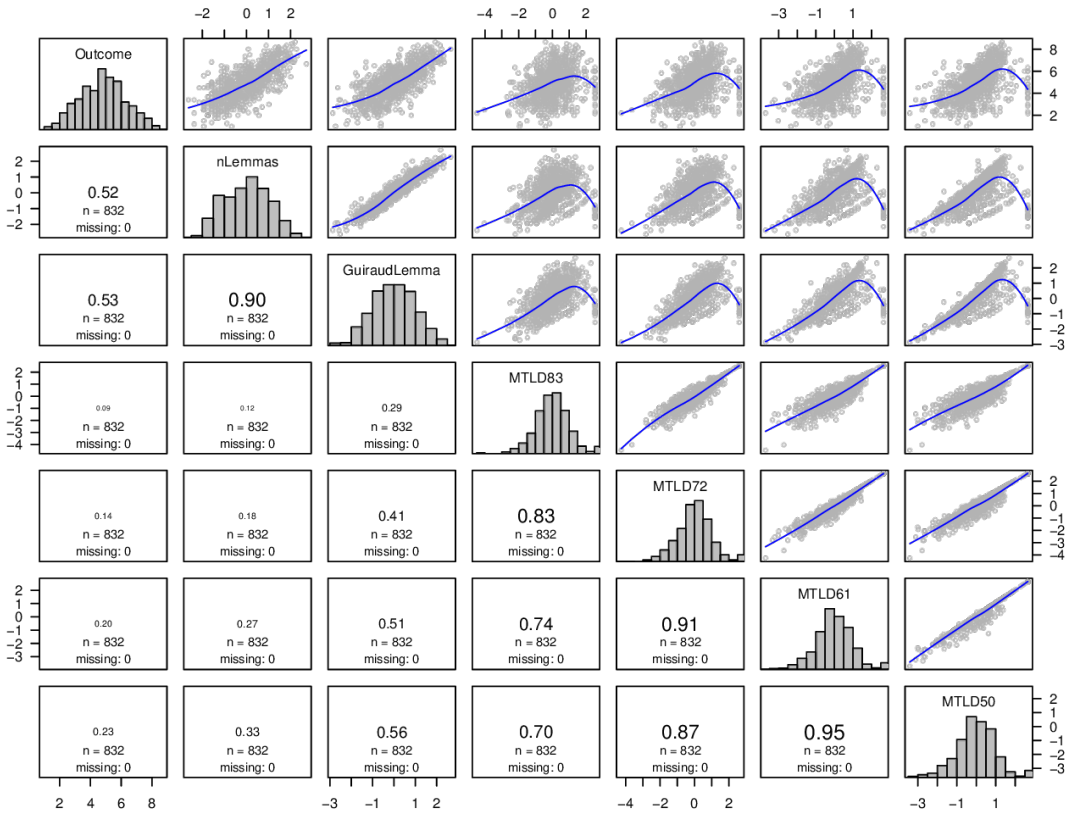

Figure 13.6: Intercorrelation between Portuguese predictors (6): MTLD. `MTLD61` and `MTLD72` were removed because of their strong intercorrelations with the other variables.

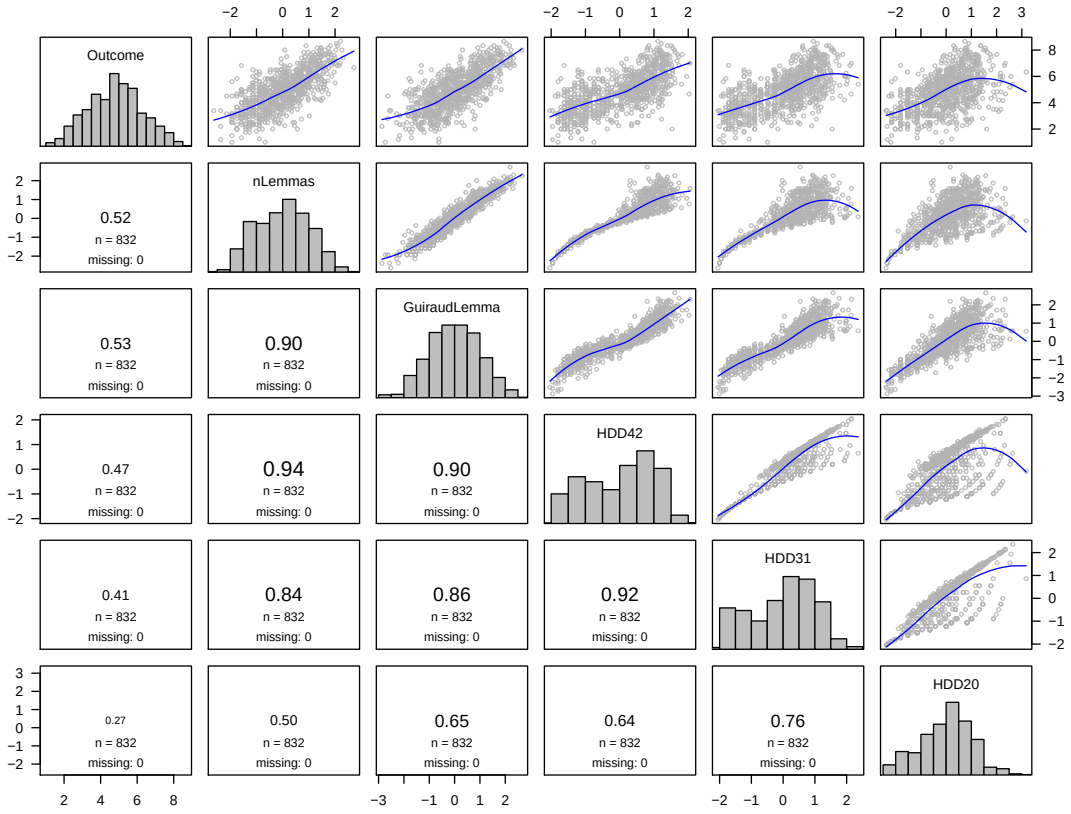

Figure 13.7: Intercorrelation between Portuguese predictors (7): HDD. `MTLD31` was removed because of their strong intercorrelations with the other variables.

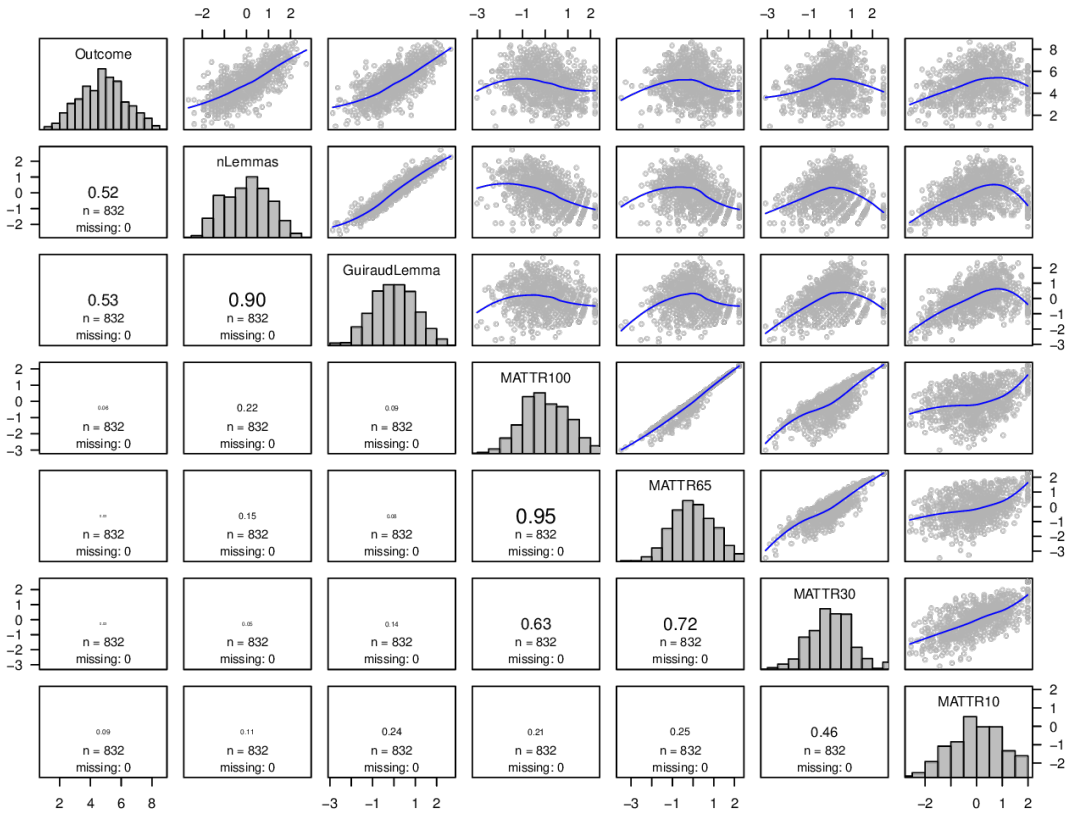

Figure 13.8: Intercorrelation between Portuguese predictors (8): MATTR. `MATTR65` was removed because of their strong intercorrelations with the other variables.

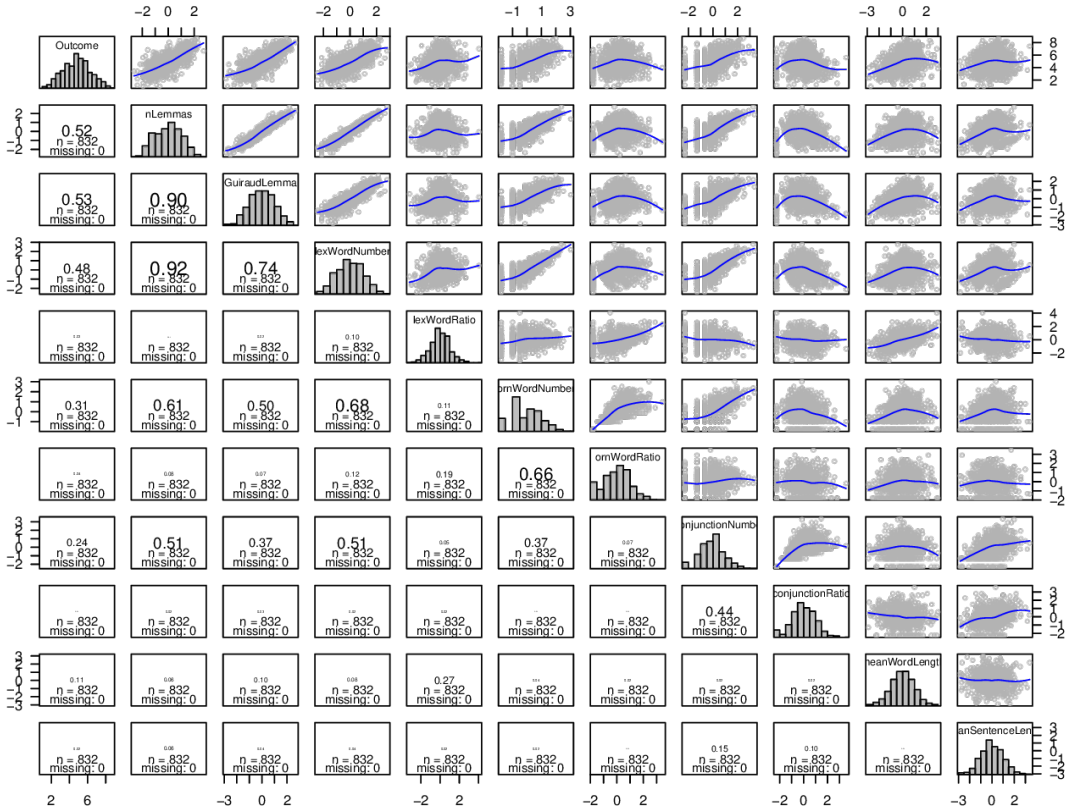

Figure 13.9: Intercorrelation between Portuguese predictors (9): Lexical and syntactic complexity. `lexWordNumber` was removed because of their strong intercorrelations with the other variables.

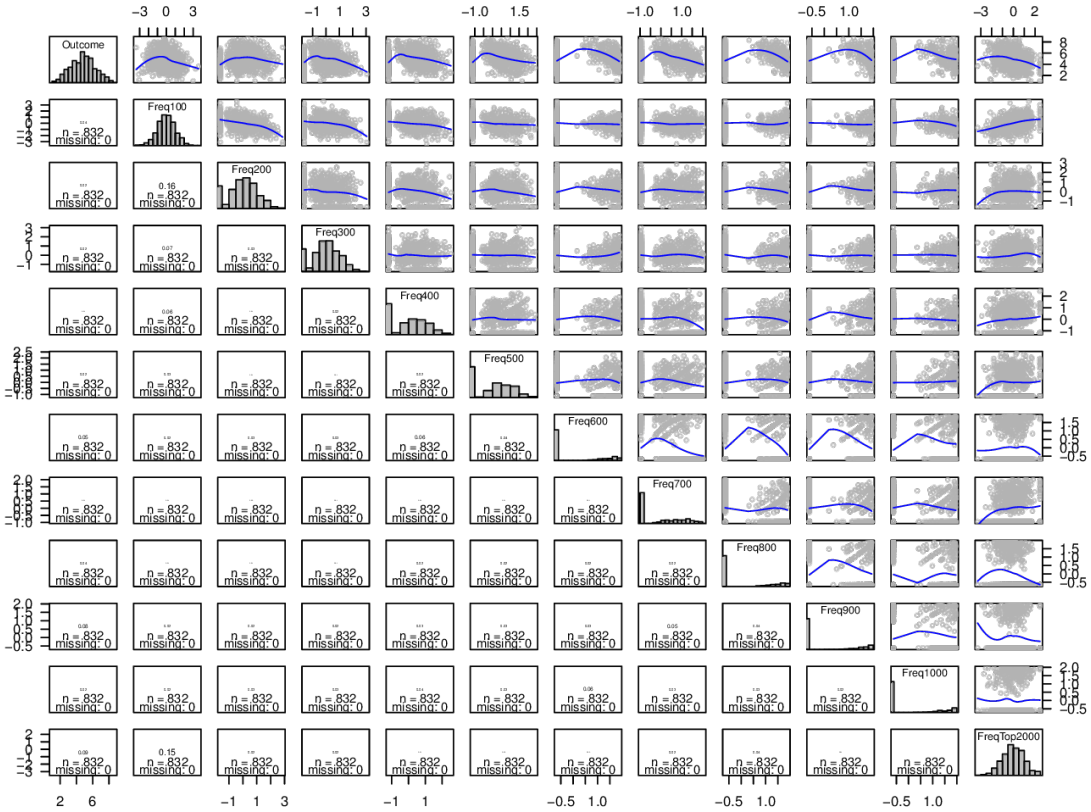

Figure 13.10: Intercorrelation between Portuguese predictors (10): Top frequency bands. No variable was removed because of their strong intercorrelations with the other variables.

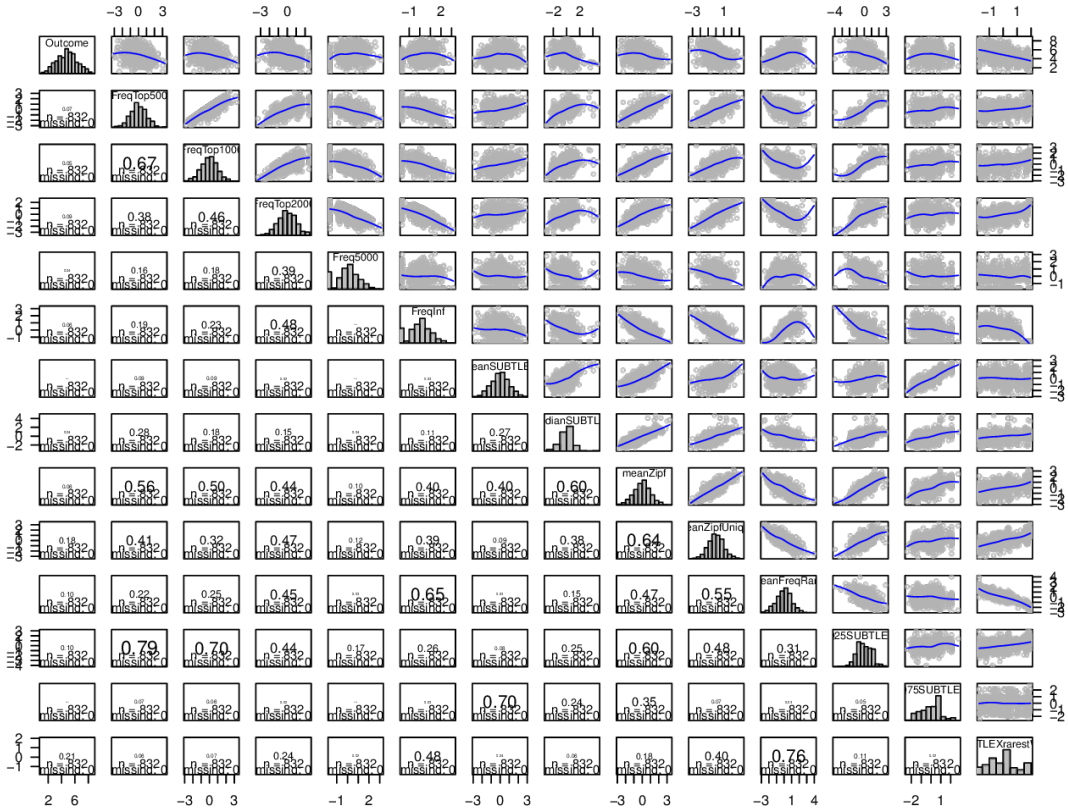

Figure 13.11: Intercorrelation between Portuguese predictors (11): Frequency summaries (token- and type-based). No variable was removed because of their strong intercorrelations with the other variables.

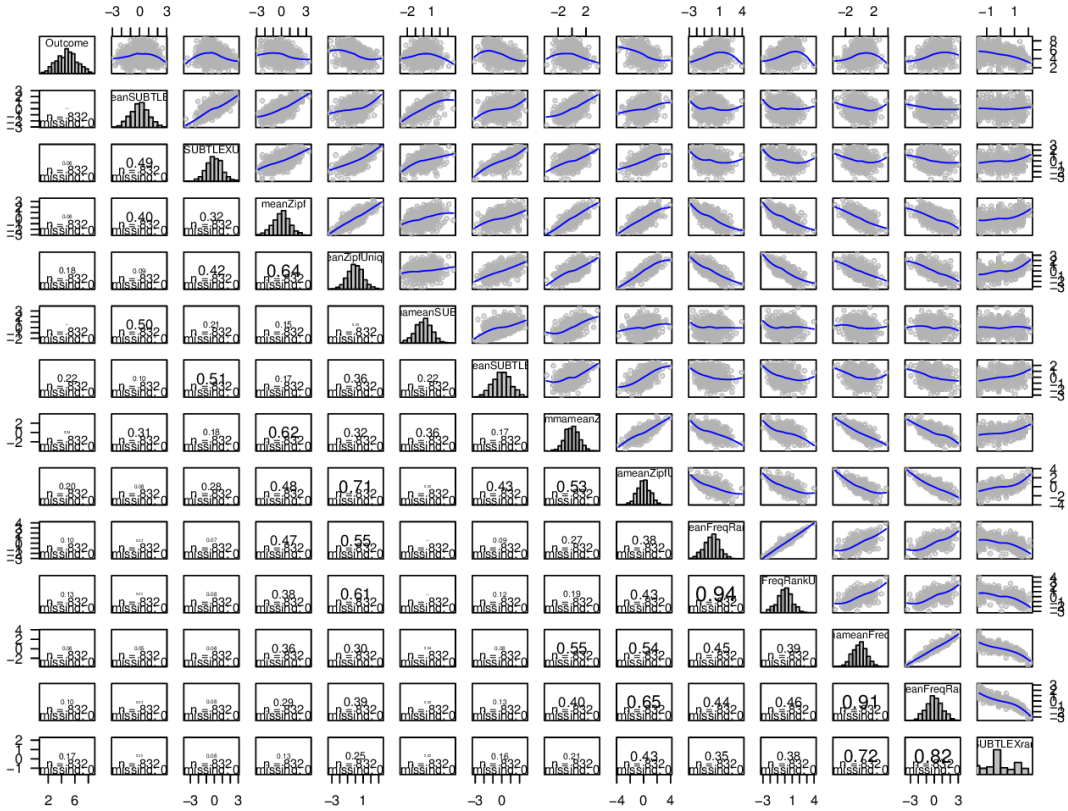

Figure 13.12: Intercorrelation between Portuguese predictors (12): Frequency summaries (lemma-based). `meanFreqRank` and `meanFreqRankUnique` were removed because of their strong intercorrelations with the other variables.

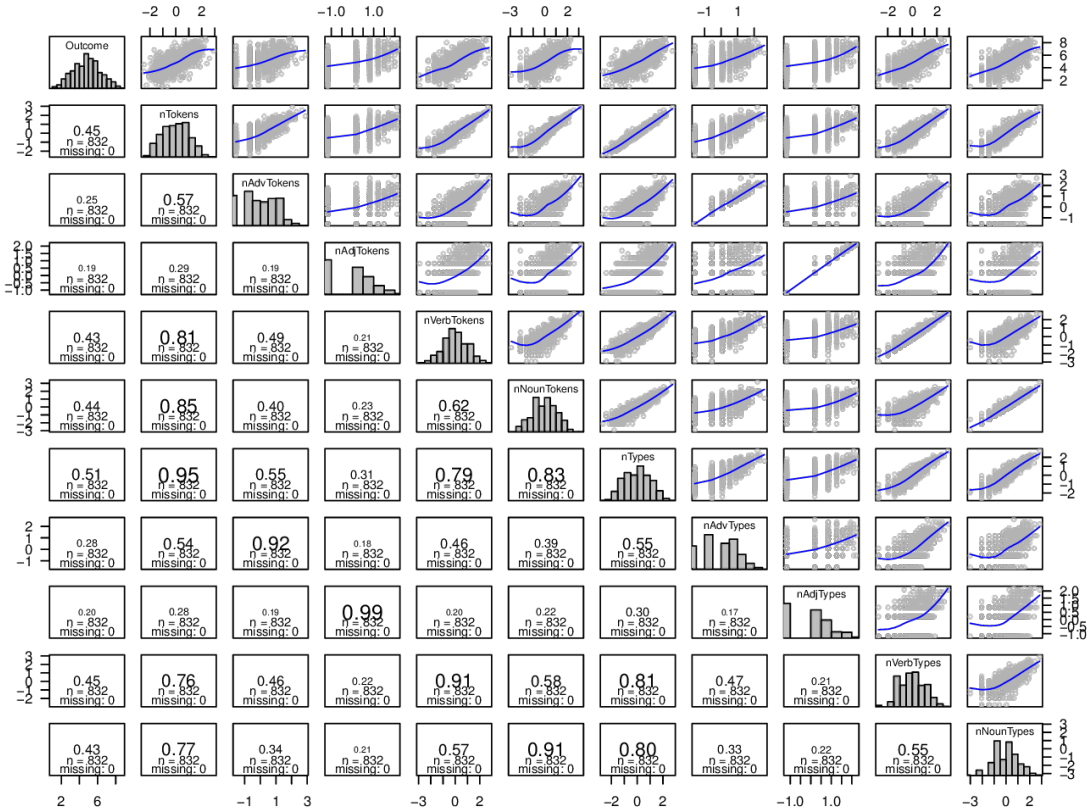

Figure 13.13: Intercorrelation between Portuguese predictors (13): Number of types and tokens by POS.

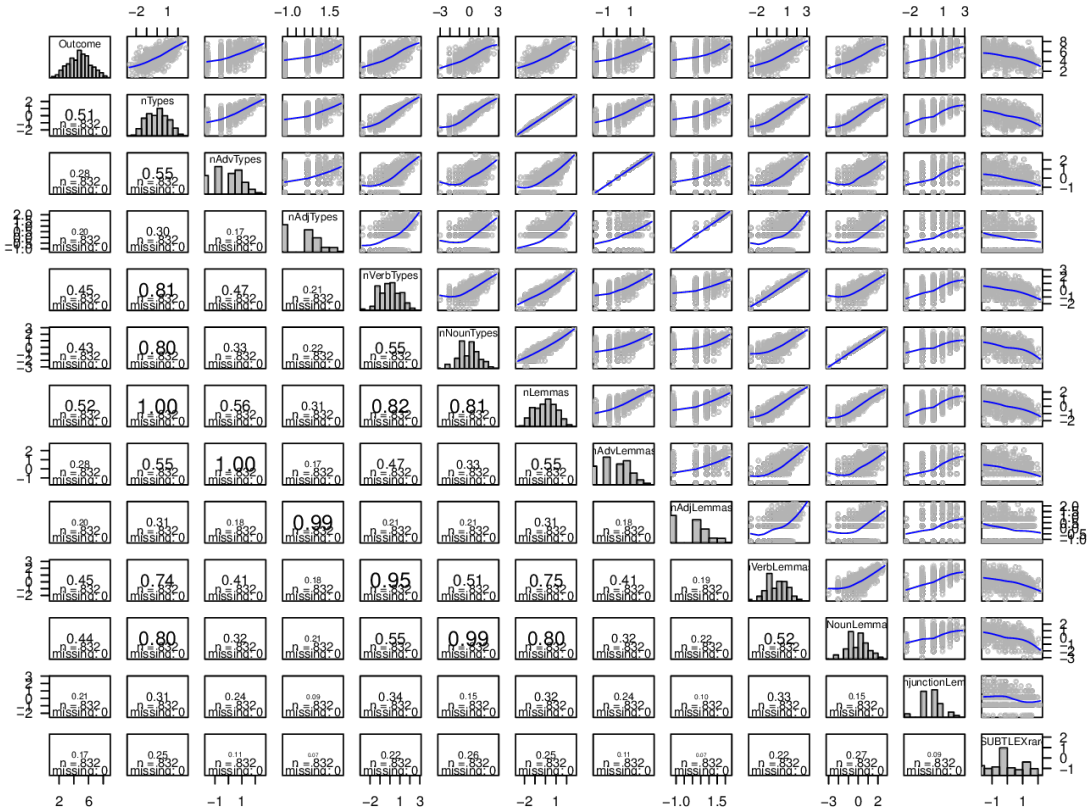

Figure 13.14: Intercorrelation between Portuguese predictors (14): Number of lemmas by POS. `nNounTokens`, `nNounTypes`, `nVerbTokens`, `nVerbTypes`, `nAdjTokens`, `nAdjTypes`, `nAdvTokens` and `nAdjTypes` were removed because of their strong intercorrelations with the other variables.

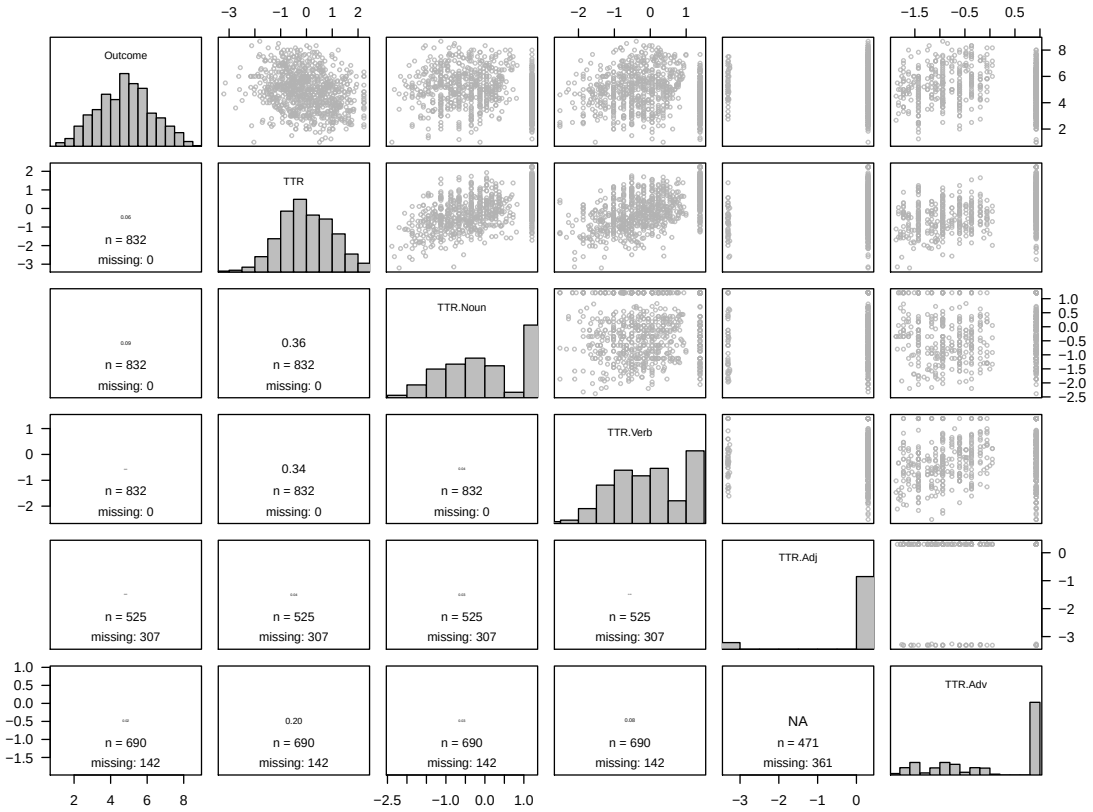

Figure 13.15: Intercorrelation between Portuguese predictors (15): TTR by part of speech. `TTR.Noun`, `TTR.Verb`, `TTR.Adj` and `TTR.Adv` removed due to missing values.

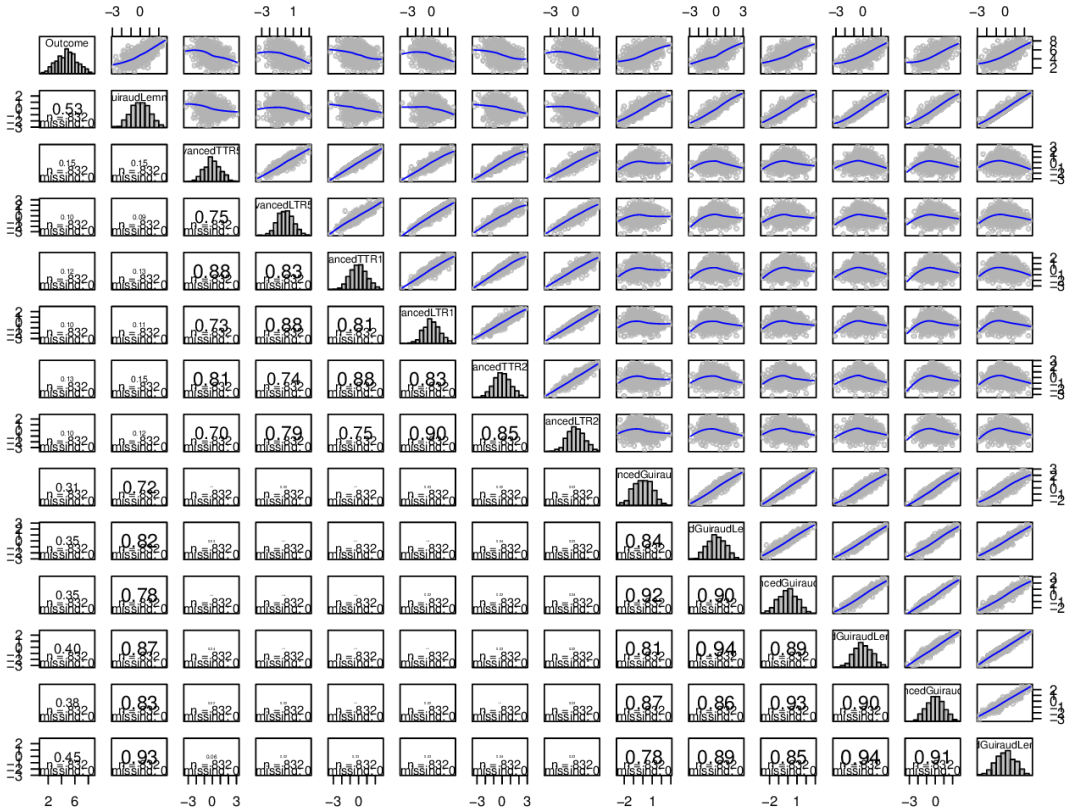

Figure 13.16: Intercorrelation between Portuguese predictors (16): Advanced Guiraud and TTR. `AdvancedLTR1000`, `AdvancedTTR1000`, `AdvancedGuiraudLemma1000`, `AdvancedGuiraud1000`, `AdvancedTTR2000`, `AdvancedGuiraudLemma2000` and `AdvancedGuiraud2000` removed due to high intercorrelations.

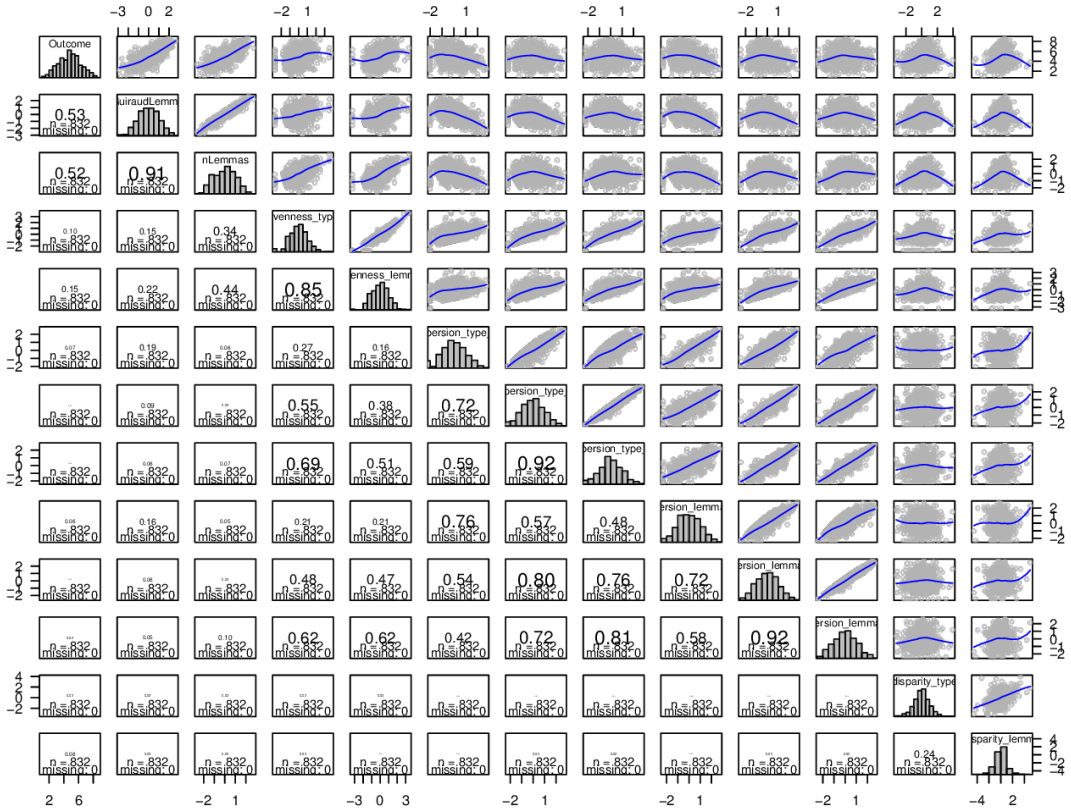

Figure 13.17: Intercorrelation between Portuguese predictors (16): Evenness, disparity and dispersion. `dispersion_type_30` and `dispersion_lemma_30` removed due to high intercorrelations.

13.4 Model performance in cross-validation

Several different algorithms were fitted to the training data and tuned using block cross-validation. For most models, the data were Yeo–Johnson transformed. Figure 12.18 shows the estimated predictive accuracy of 14 tuned models. The algorithm with the greatest predictive power was the elastic net/LASSO, with a mean RMSE of 0.971, followed by MARS and partial least squares with mean RMSE values of 0.971 and 0.972, respectively.

13.5 Model stacking

The out-of-fold predictions for all 14 models were extracted. The Pearson correlations between the out-of-fold predictions of the different models varied between 0.86 and 0.9996 with a median correlation of 0.965. These strong correlations suggest that any gain in predictive accuracy from model stacking will be small.

The out-of-fold predictions of each model were used as predictors in a principal component regression model. This is a linear regression model for which principal component analysis was first applied to the predictors. The predictive accuracy of this model was assessed block cross-validation The RMSE was estimated to be 0.996. This represents a slight *drop* in performance relative to the single best model, but for comparability with French and German, the stacking model was nonetheless used for prediction.

13.6 Does predictive accuracy depend on text length?

Figure 13.19 shows the correlations between text length and the out-of-fold predictions and residuals according to the stacked model. Longer texts are predicted to have better ratings (left). The variance of the residuals seems to be slightly larger for shorter than for longer texts.

13.7 Variable importance in top-2 models

Figure 13.20 shows the variable importances of the 20 most important predictors in the top 3 models (elastic net, MARS, partial least squares). These values were extracted using `caret`'s `varImp()` function, and for each of the three models, the variables were rank-ordered from less to more important. The twenty variables with the highest mean rank across the two models are shown in the plot.

The single most important variable in both models is `GuiraudLemma`, that is, $\frac{\text{number of lemmata}}{\sqrt{\text{number of tokens}}}$. After this two variable, the models show different pictures. (The

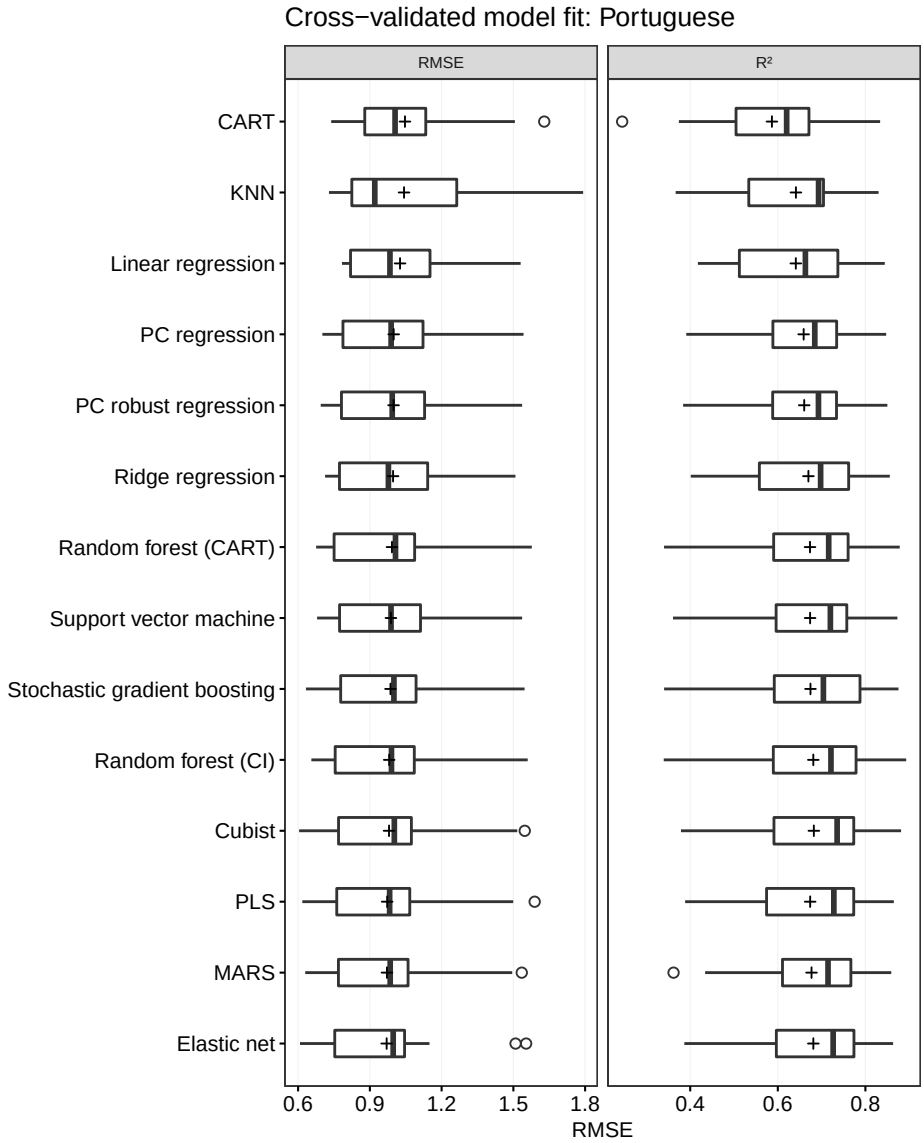

Figure 13.18: Performance of 14 tuned predictive models for the Portuguese training data in block cross-validation (with 16 blocks). The crosses mark the mean of each distribution.

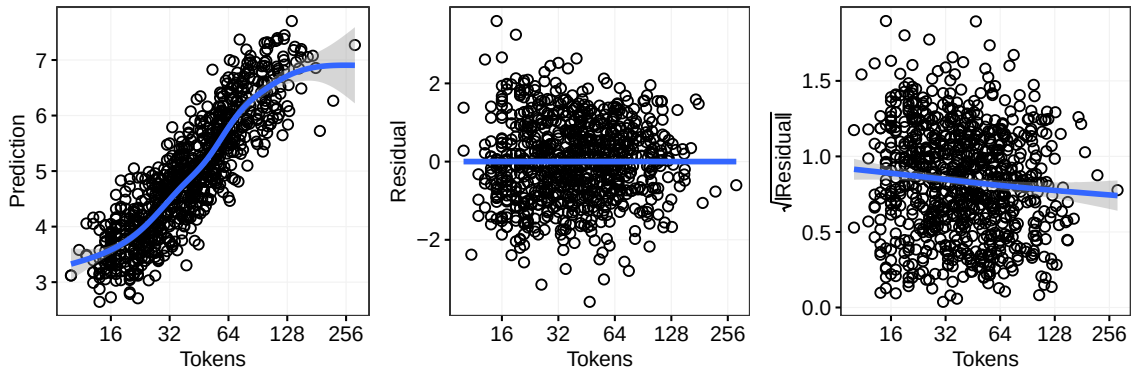

Figure 13.19: *Left:* Text length and out-of-fold prediction according to the stacked model for all 832 training texts. *Middle:* Text length and residuals (actual value − average out-of-fold prediction). *Right:* Text length and root absolute residuals.

other variables identified as important by enet and PLS are essentially all correlates of `GuiraudLemma`.)

13.8 A 6-dimensional model

A more interpretable 6-dimensional model based on the framework proposed by Jarvis (2013a,b) was fitted to the training data. This model contained the following predictors. For Portuguese, these choices were not based on data exploration but on the fact that these variables had turned out to be useful for French and German.

- Volume: The number of tokens. (Log-transformed)
- Variability: MTLD with a TTR setting of 0.83. The MTLD was chosen as it is not systematically affected by the texts' length. (Log-transformed)
- Evenness: The lemma-based evenness index. (Square-root transformed)
- Rarity: The mean Zipf value of the unique lemmata occurring in the texts.
- Disparity: The disparity index computed with respect to lemmata.
- Dispersion: The dispersion index computed with respect to lemmata and $k = 20$. (Square-root transformed)

As Figure 13.21 shows, these predictors weren't entirely independent of one another.

These predictors were fitted in a generalised additive model whose RMSE was estimated to be 0.985 ± 0.068 in block cross-validation, respectively.

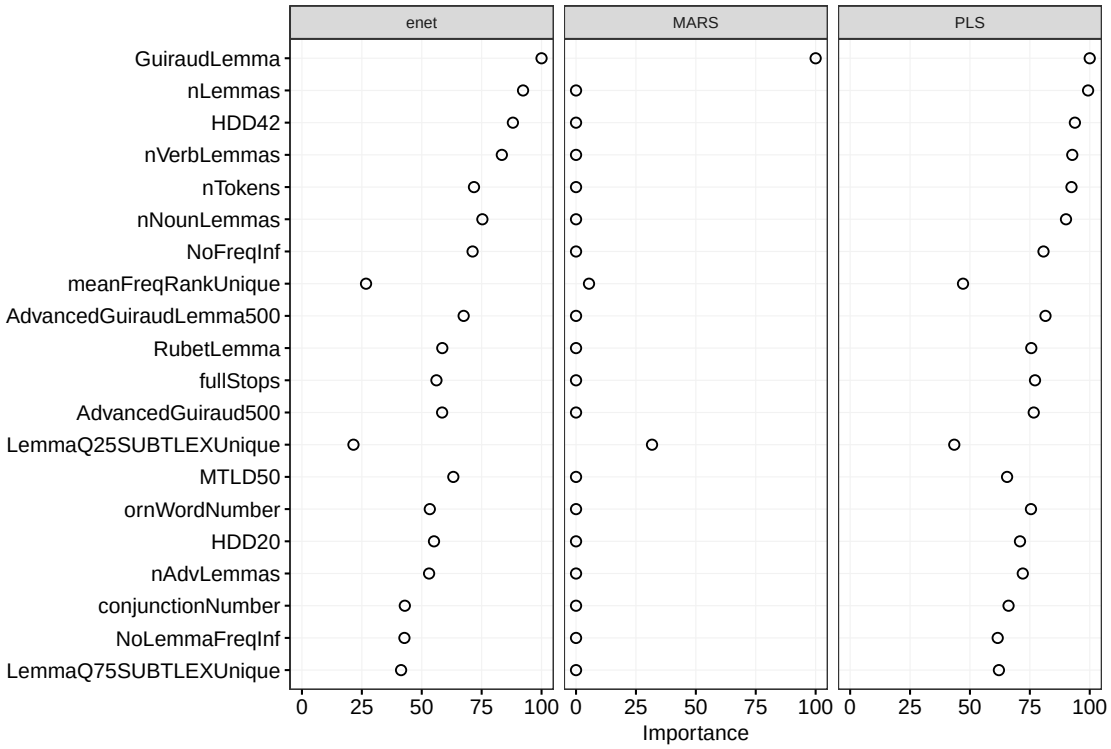

Figure 13.20: Variable importances of the 20 predictors with the highest mean rank in the three best performing predictive models.

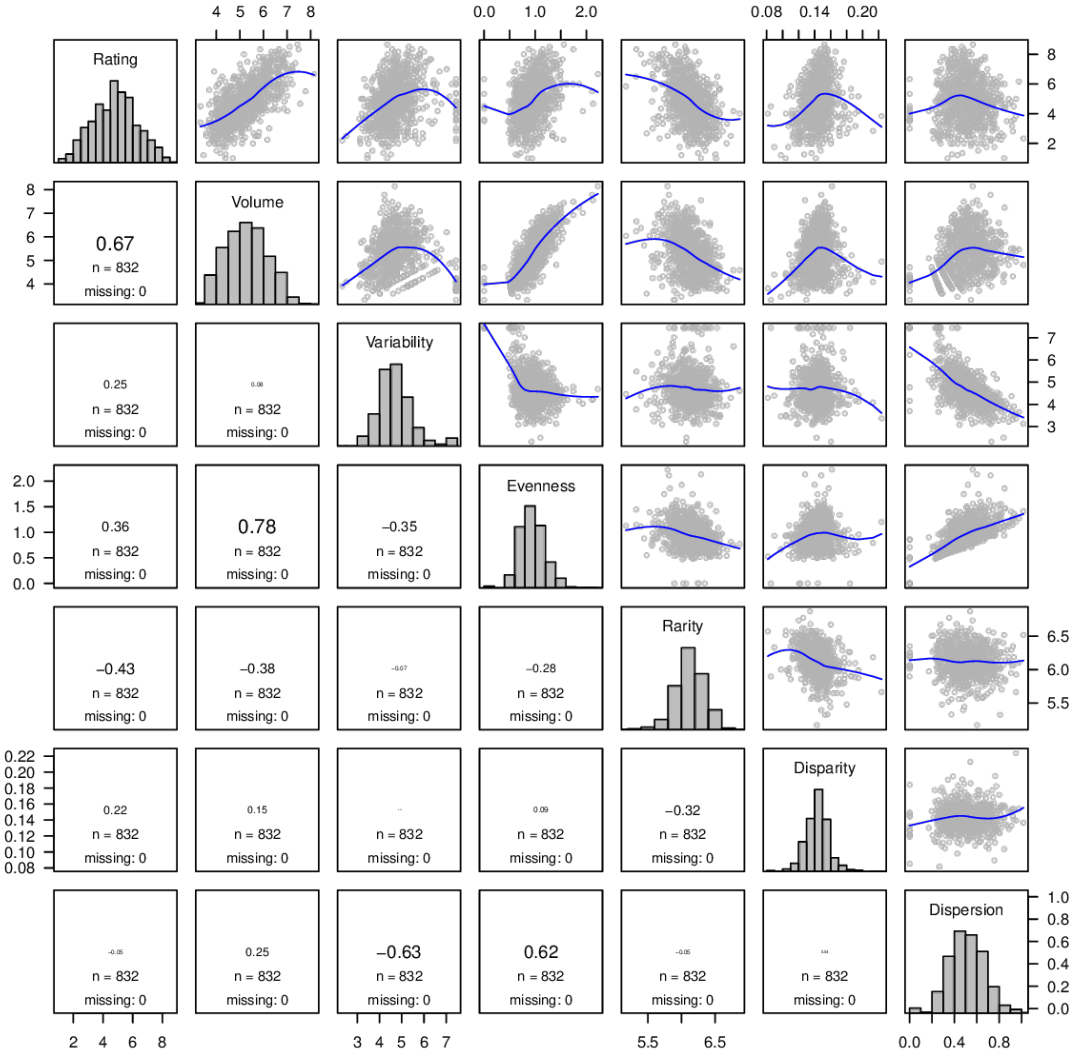

Figure 13.21: Bivariate relationships between the six predictors in the generalised additive model predicting the Portuguese ratings. The numbers in the bottom triangle are Pearson correlations.

The partial effects of the six predictors in this GAM are shown in Figure 13.22. Table 13.1 summarises the model numerically.

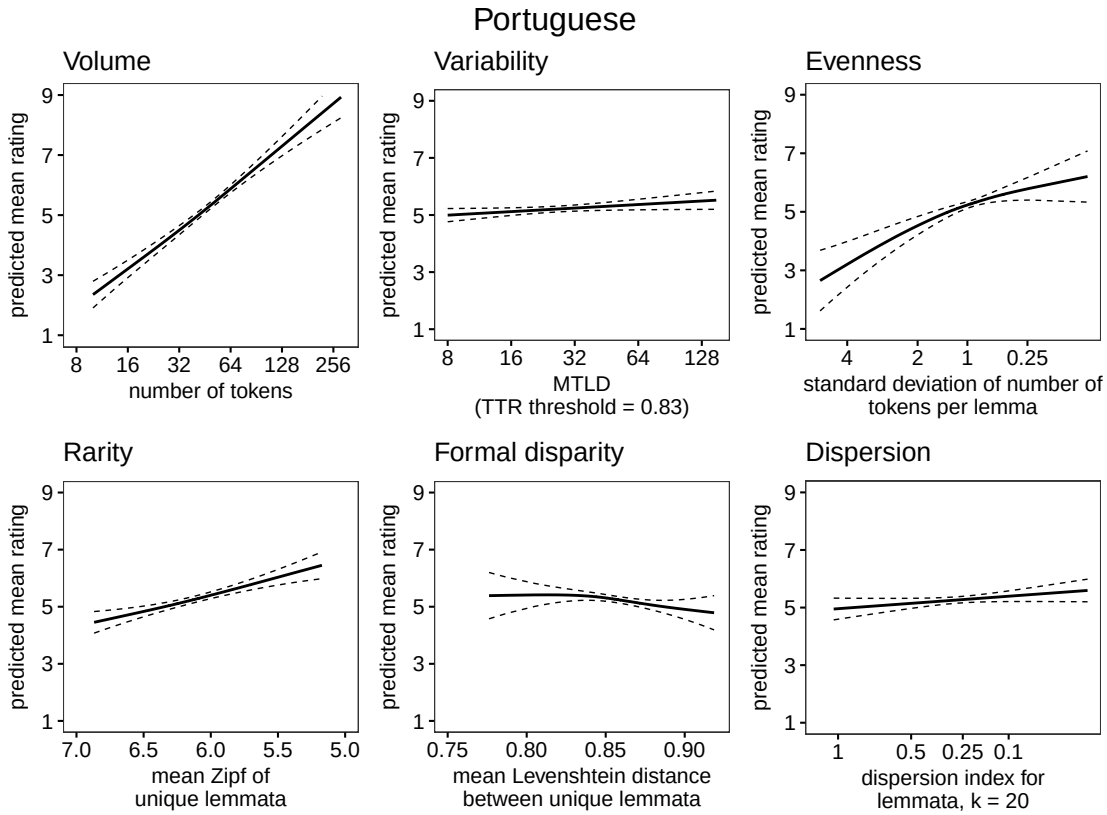

Figure 13.22: Partial effects of a generalised additive model fitted on the Portuguese training data using six predictors corresponding to Jarvis' 6 dimensions.

13.9 A single-predictor model

`GuiraudLemma` emerged as the single best predictor in the black boxes. A linear regression model with it as its sole predictor achieves a RSME of 1.01 in block cross-validation. The predictive function is:

$$\text{Predicted mean rating} = 0.11 + 1.04 \times \texttt{GuiraudLemma} \tag{13.1}$$

Since `GuiraudLemma` and `Guiraud` (type-based) were strongly correlated, a regression model with `Guiraud` as its sole predictor has virtually the same predictive accuracy. Its regression

Table 13.1: Summary of a generalised additive model fitted on the Portuguese training data.

Term	Type	Estimate / edf	Test statistic	p
Intercept	parametric	4.6	$t = 12$	< 0.001
Number of tokens (log2)	smooth	1.7	$F = 116$	< 0.001
MTLD 0.83 (log2)	parametric	0.12	$t = 2.0$	0.044
Evenness lemmata (sqrt)	smooth	2.5	$F = 8.8$	< 0.001
Mean Zipf, unique lemmata	smooth	1.3	$F = 30$	< 0.001
Disparity, lemmata	smooth	2.0	$F = 3.3$	0.024
Dispersion, lemmata, $k = 20$ (sqrt)	parametric	-0.63	$t = 1.7$	0.087

equation is:

$$\text{Predicted mean rating} = 0.26 + 0.97 \times \texttt{Guiraud} \tag{13.2}$$

13.10 Comparison of the three approaches

The predictive accuracy of the three approaches was directly compared using a series of paired t-tests ran on the 16 cross-validation estimates. All comparisons are RMSE-based:

- Black-box versus 6 dimensions: 6 dimensions 0.011 points better on average ($t(15) = 1.1$, $p = 0.29$).
- Black-box versus Guiraud: Black-box 0.011 points better on average ($t(15) = 0.8$, $p = 0.45$).
- 6 dimensions versus Guiraud: 6 dimensions 0.022 points better on average ($t(15) = 1.9$, $p = 0.074$).

Chapter 14

Test set performance

Finally, the models presented in the previous chapters were applied to the test sets. Crucially, the models were not re-estimated on the basis of the test sets, nor were they in any way changed after seeing the their performance on the test sets.

For the black-box approach, the stacked models were used for all three languages. For Portuguese, this model performed somewhat worse in cross-validation than did some of the individual models, but stacking was used in this case, too, to simplify the comparison with French and German. In order to generate predictions from the stacked models, predictions based on the base models were first generated, which were then used as predictors for the stacked models.

Figure 14.1 shows the observed test set values and the predicted test set values according to all three modelling approaches.

Table 14.1 summarises the models' performance in terms of their RMSE and R^2. Standard errors are reported to highlight the fact that the RMSE and R^2 values are point estimates subject to sampling variation. These standard errors were estimated using bootstrapping: 10,000 new datasets were created by resampling (with replacement) the pairs of observed and predicted values, and the RMSE and R^2 values were computed for each newly created dataset. The standard deviations of the distributions of the 10,000 RMSE and R^2 values serves as the SE estimate.

14.1 Comparison of the three approaches

For the sake of completeness, the predictive accuracy of the three modelling approaches was directly compared. This was done by computing the absolute prediction error per text

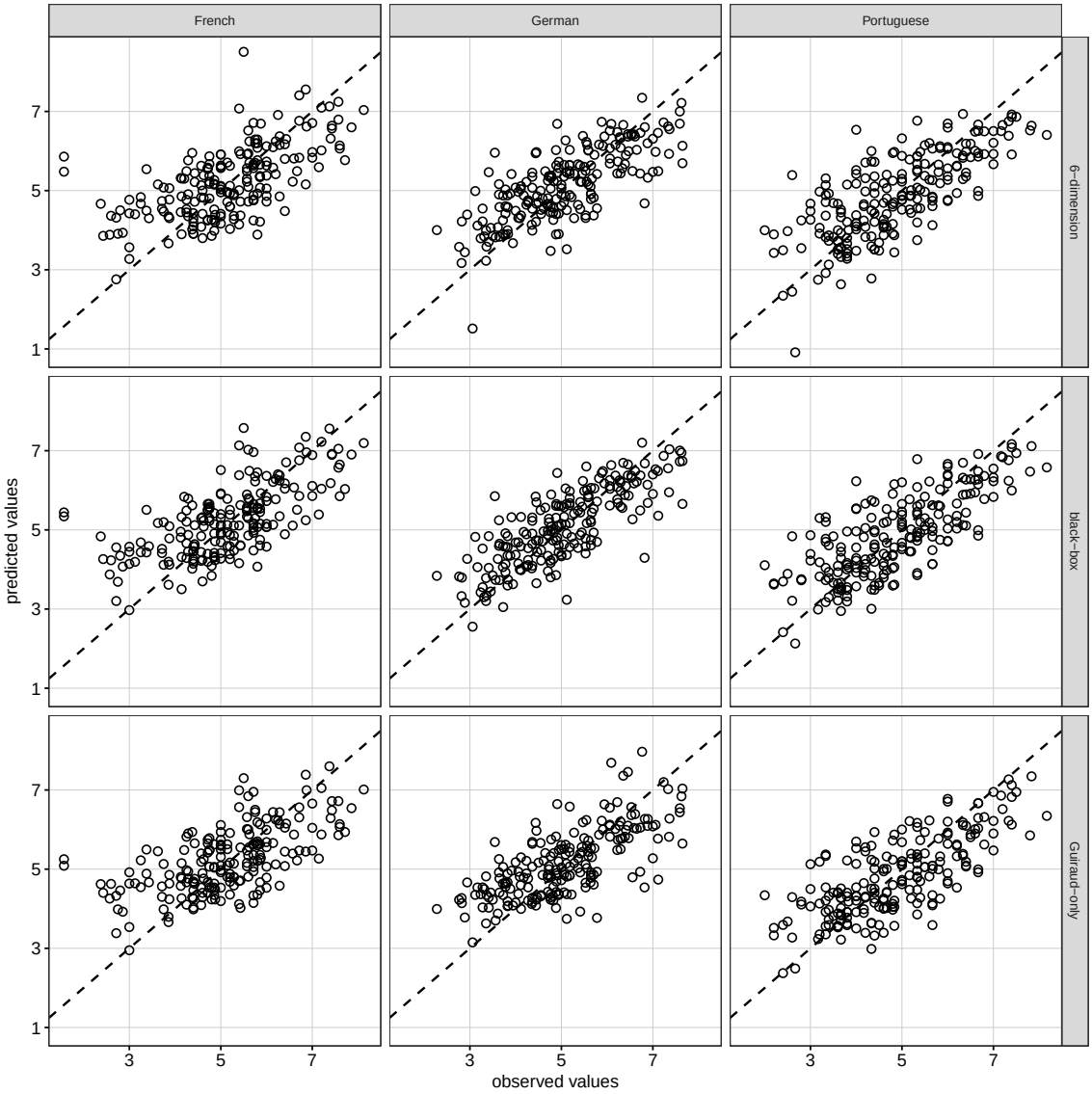

Figure 14.1: The observed test set values and the predictions according to the three approaches. The diagonal line is the line of equalit ($y = x$); points above this line are overpredicted, points below it are underpredicted.

Table 14.1: Performance of the three models on the independent test sets. The standard errors were estimated using bootstrapping.

Language	Approach	RMSE $\pm$ SE	$R^2 \pm$ SE
French	Black-box	0.937 ± 0.063	0.437 ± 0.058
	6 dimensions	0.983 ± 0.070	0.381 ± 0.070
	Guiraud only	0.974 ± 0.058	0.391 ± 0.052
German	Black-box	0.724 ± 0.041	0.623 ± 0.045
	6 dimensions	0.793 ± 0.040	0.547 ± 0.047
	Guiraud only	0.846 ± 0.041	0.485 ± 0.050
Portuguese	Black-box	0.844 ± 0.040	0.576 ± 0.045
	6 dimensions	0.877 ± 0.043	0.542 ± 0.048
	Guiraud only	0.866 ± 0.040	0.554 ± 0.046

for each method and comparing these using paired t-tests. The results are reported in Table 14.2.

14.2 Predictability and text length

Figure 14.2 shows how the prediction errors (for the black-box approach) in the test sets vary according to the texts' length.

Table 14.2: Comparison of the test set performance of the three approaches

Language	Comparison ($A - B$)	$\lvert\text{error}(A)\rvert < \lvert\text{error}(B)\rvert$	t-test
French	Black-box − 6 dimensions	114 out of 200 (57%)	$t(199) = 2.1$, $p = 0.04$
	Black-box − Guiraud only	109 out of 200 (55%)	$t(199) = 1.6$, $p = 0.12$
	6 dimension − Guiraud only	106 out of 200 (53%)	$t(199) = 0.3$, $p = 0.76$
German	Black-box − 6 dimensions	125 out of 204 (61%)	$t(203) = 3.3$, $p = 0.001$
	Black-box − Guiraud only	115 out of 204 (56%)	$t(203) = 3.8$, $p < 0.001$
	6 dimension − Guiraud only	114 out of 204 (56%)	$t(203) = 1.6$, $p = 0.12$
Portuguese	Black-box − 6 dimensions	104 out of 208 (50%)	$t(207) = 1.6$, $p = 0.12$
	Black-box − Guiraud only	108 out of 208 (52%)	$t(207) = 0.6$, $p = 0.58$
	6 dimension − Guiraud only	103 out of 208 (50%)	$t(207) = 0.6$, $p = 0.56$

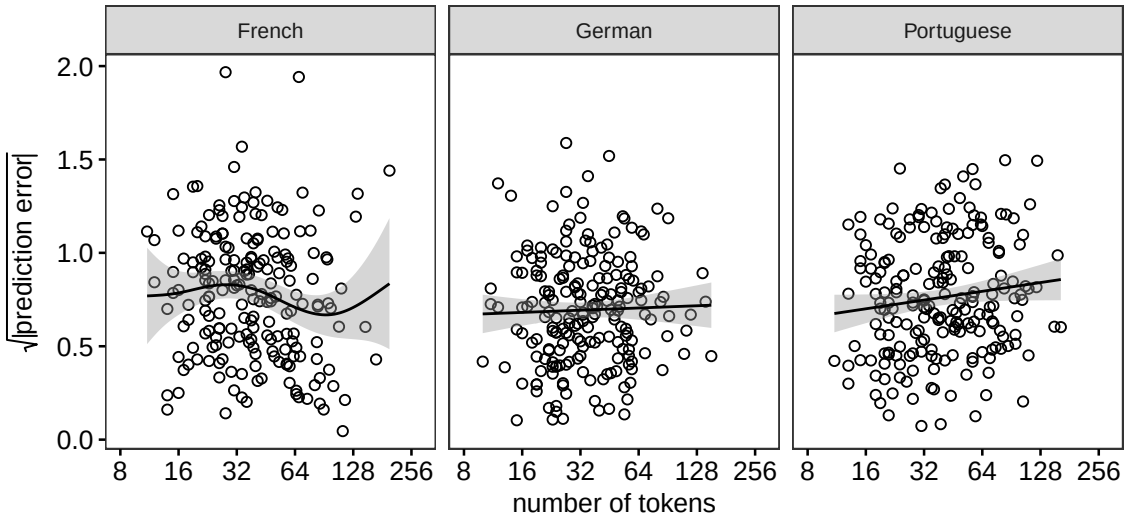

Figure 14.2: The squared absolute prediction errors in the test sets for the black-box approach plotted against the length of the texts.

Chapter 15

Predictive modelling: Summary

Figure 15.1 summarises the predictive power of the three modelling approaches in both cross-validation and in independent test sets that were not used during data exploration and model fitting. For reference, the performance (in terms of RMSE) of a model without any predictors is shown as well. Such a model predicts all test data to be equal to the mean of the training data. (Because it produces no variation in the predicted values, R^2 cannot be computed for this model.)

Figure 15.2 shows how the prediction error of the black-box approach co-varies with the texts' length in both cross-validation and in the independent test sets. Shorter texts are not generally less well predicted than are lower texts.

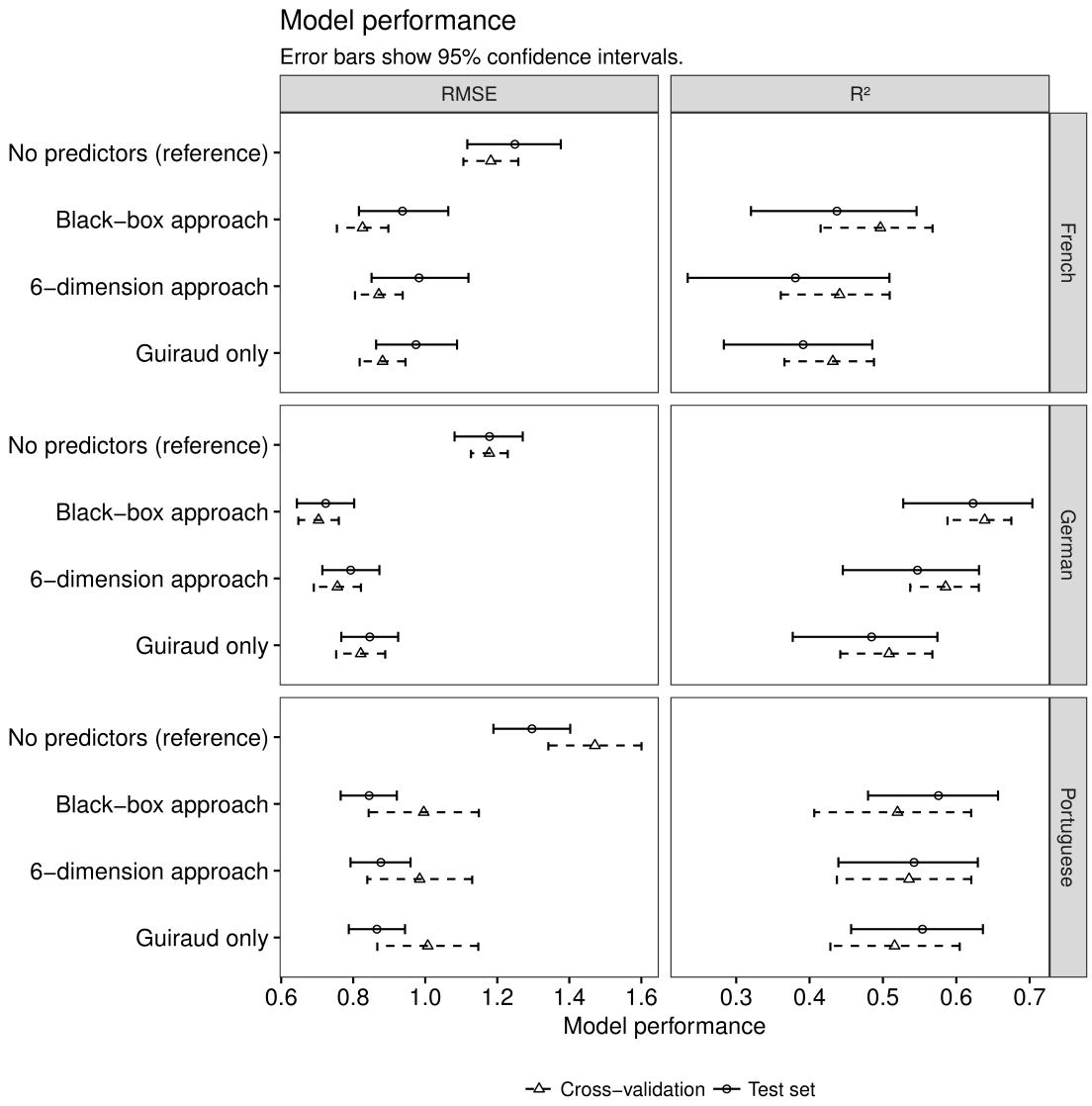

Figure 15.1: The root mean squared errors (RMSE) and the coefficients of determination (R^2) of the different approaches to predicting average lexical richness ratings in both cross-validation and independent test sets. Lower RMSEs indicate greater predictive accuracy; higher R^2 values indicate a stronger decrease in the residual sum of squares of the predictive model relative to the reference model. No R^2 was computed for the reference model as it is 0 by definition. The confidence intervals were computed using bootstrapping (R^2 and RMSE for test set data; percentile method) or based on $t(15)$-distributions (RMSE for cross-validation data).

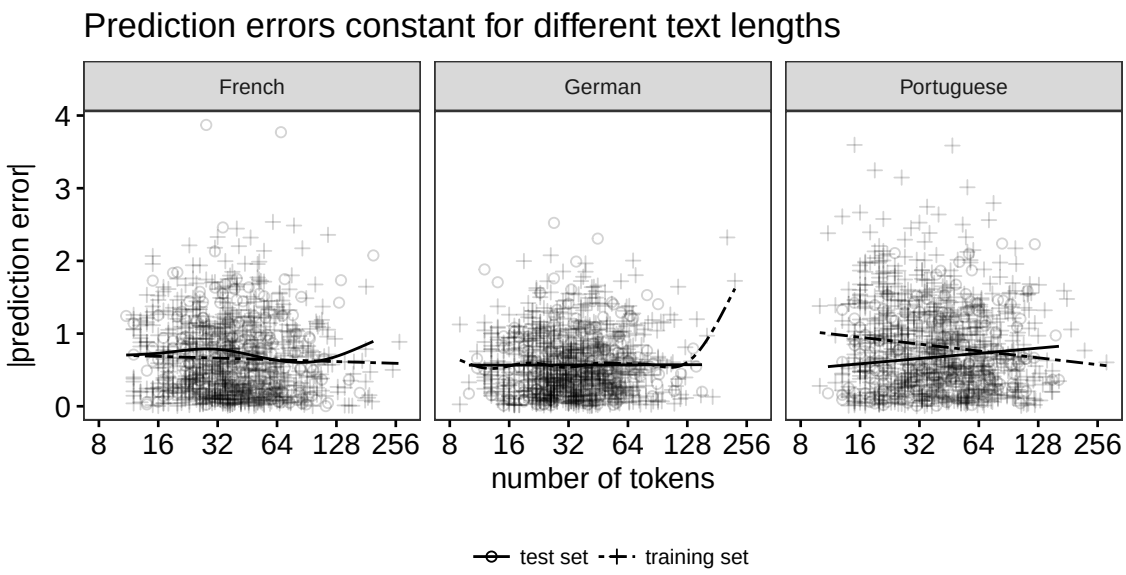

Figure 15.2: The absolute prediction errors in both cross-validation and independent test sets according to the black-box approach plotted against the texts' length.

Bibliography

Baayen, R. Harald. 2008. *Analyzing linguistic data: A practical introduction to statistics using R.* Cambridge: Cambridge University Press.

Box, G. E. P. & D. R. Cox. 1964. An analysis of transformations. *Journal of the Royal Statistical Society. Series B (Methodological)* 26(2). 211–252.

Breiman, Leo. 1996. Stacked regressions. *Machine Learning* 24. 49–64. doi:10.1007/BF00117832.

Breiman, Leo. 2001. Statistical modeling: The two cultures. *Statistical Science* 16(3). 199–231. doi:10.1214/ss/1009213726.

Brysbaert, Marc, Matthias Buchmeier, Markus Conrad, Arthur M. Jacobs, Jens Bölte & Andrea Böhl. 2011. The word frequency effect: A review of recent developments and implications for the choice of frequency estimates in German. *Experimental Psychology* 58. 412–424. doi:10.1027/1618-3169/a000123.

Brysbaert, Marc & Kevin Diependaele. 2013. Dealing with zero word frequencies: A review of the existing rules of thumb and a suggestion for an evidence-based choice. *Behavior Research Methods* 45(2). 422–430. doi:10.3758/s13428-012-0270-5.

Covington, Michael A. & Joe D. McFall. 2010. Cutting the Gordian knot: The moving-average type–token ratio (MATTR). *Journal of Quantitative Linguistics* 17(2). doi:10.1080/09296171003643098.

Daller, Helmut, Roeland van Hout & Jeanine Treffers-Daller. 2003. Lexical richness in the spontaneous speech of bilinguals. *Applied Linguistics* 24(2). 197–222. doi:10.1093/applin/24.2.197.

Desgrippes, Magalie & Amelia Lambelet. 2017. On the sociolinguistic embedding of Portuguese heritage language speakers in Switzerland: Socioeconomic status and home literacy environment (HELASCOT project). In Raphael Berthele & Amelia Lambelet (eds.), *Heritage and school language literacy development in migrant children: Interdependence*

or independence?, 34–57. Bristol: Multilingual Matters. doi:10.21832/9781783099054-004.

Desgrippes, Magalie, Amelia Lambelet & Jan Vanhove. 2017. The development of argumentative and narrative writing skills in Portuguese heritage speakers in Switzerland (HELASCOT project). In Raphael Berthele & Amelia Lambelet (eds.), *Heritage and school language literacy development in migrant children: Interdependence or independence?*, 83–96. Bristol: Multilingual Matters. doi:10.21832/9781783099054-006.

Goldstein, Alex, Adam Kapelner, Justin Bleich & Emil Pitkin. 2015. Peeking inside the black box: Visualizing statistical learning with plots of individual conditional expectation. *Journal of Computational and Graphical Statistics* 24(1). 44–65. doi:10.1080/10618600.2014.907095.

Guiraud, Pierre. 1954. *Les charactères statistiques du vocabulaire. Essai de méthodologie.* Paris: Presses Universitaires de France.

Heeringa, Wilbert. 2004. *Measuring dialect pronunciation differences using Levenshtein distance*: University of Groningen dissertation.

Jarvis, Scott. 2013a. Capturing the diversity in lexical diversity. *Language Learning* 63(Supplement 1). 87–106. doi:10.1111/j.1467-9922.2012.00739.x.

Jarvis, Scott. 2013b. Defining and measuring lexical diversity. In Scott Jarvis & Michael Daller (eds.), *Vocabulary knowledge: Human ratings and automated measures*, 13–43. Amsterdam: John Benjamins. doi:10.1075/sibil.47.

Jarvis, Scott. 2017. Grounding lexical diversity in human judgments. *Language Testing* 34(4). 537–553. doi:10.1177/0265532217710632.

Johnson, Wendell. 1944. Studies in language behavior: I. A program of research. *Psychological Monographs* 56. 1–15.

Kim, Ji-young. 2014. Predicting L2 writing proficiency using linguistic complexity measures: A corpus-based study. *English Teaching* 69(4). 27–51. doi:10.15858/engtea.69.4.201412.27.

Kuhn, Max & Kjell Johnson. 2013. *Applied predictive modeling.* New York: Springer. doi:10.1007/978-1-4614-6849-3.

Kvålseth, Tarald O. 1985. Cautionary note about R^2. *The American Statistician* 4(1). doi:10.2307/2683704.

Kyle, Kristopher & Scott A. Crossley. 2015. Automatically assessing lexical sophistication: Indices, tools, findings, and application. *TESOL Quarterly* 49(4). doi:10.1002/tesq.194.

Lambelet, Amelia, Raphael Berthele, Magalie Desgrippes, Carlos Pestana & Jan Vanhove. 2017. Testing interdependence in Portuguese heritage speakers in Switzerland: the HELASCOT project. In Raphael Berthele & Amelia Lambelet (eds.), *Heritage and school language literacy development in migrant children: Interdependence or independence?*, 26–33. Bristol: Multilingual Matters. doi:10.21832/9781783099054-003.

Levenshtein, Vladimir Iosifovich. 1966. Binary codes capable of correcting deletions, insertions, and reversals. *Soviet Physics Doklady* 10. 707–710.

McCarthy, Philip M. & Scott Jarvis. 2007. vocd: A theoretical and empirical evaluation. *Language Testing* 24(4). 459–488. doi:10.1177/0265532207080767.

McCarthy, Philip M. & Scott Jarvis. 2010. MTLD, voc-D, and HD-D: A validation study of sophisticated approaches to lexical diversity assessment. *Behavior Research Methods* 42(2). 381–392. doi:10.3758/BRM.42.2.381.

Michalke, Meik. 2017. *koRpus: An R package for text analysis.* http://reaktanz.de/?c=hacking&s=koRpus. Version 0.10-1.

New, Boris, Marc Brysbaert, Jean Veronis & Christophe Pallier. 2007. The use of film subtitles to estimate word frequencies. *Applied Psycholinguistics* 28. 661–677. doi:10.1017/S014271640707035X.

Revelle, William. 2021. *psych: Procedures for psychological, psychometric, and personality research.* https://personality-project.org/r/psych/https://personality-project.org/r/psych-manual.pdf. R package version 2.1.9.

Roberts, David R., Volker Bahn, Simone Ciuti, Mark S. Boyce, Jane Elith, Gurutzeta Guillera-Arroita, Severin Hauenstein, José J. Lahoz-Monfort, Boris Schröder, Wilfried Thuiller, David I. Warton, Brendan A. Wintle, Florian Hartig & Carsten F. Dormann. 2017. Cross-validation strategies for data with temporal, spatial, hierarchical, or phylogenetic structure. *Ecography* 40(8). 913–929. doi:10.1111/ecog.02881.

Schmid, Helmut. 1994. Probabilistic part-of-speech tagging using decision trees. In *Proceedings of international conference on new methods in language processing*, Manchester.

Shrout, Patrick E. & Joseph L. Fleiss. 1979. Intraclass correlations: Uses in assessing rater reliability. *Psychological Bulletin* 86(2). 420–428. doi:10.1037/0033-2909.86.2.420.

Soares, Ana Paula, João Machado, Ana Costa, Álvaro Iriarte, Alberto Simes, José João de Almeida, Montserrat Comesaña & Manuel Perea. 2015. On the advantages of word frequency and contextual diversity measures extracted from subtitles: The case of Portuguese. *The Quarterly Journal of Experimental Psychology* 68(4). 680–696. doi:10.1080/17470218.2014.964271.

Tagliamonte, Sali A. & R. Harald Baayen. 2012. Models, forests, and trees of York English: *Was/were* variation as a case study for statistical practice. *Language Variation and Change* 24(2). 135–178. doi:10.1017/S0954394512000129.

Tweedie, Fiona J. & R. Harald Baayen. 1998. How variable may a constant be? Measures of lexical richness in perspective. *Computers and the Humanities* 32(5). 323–352. doi:10.1023/A:1001749303137.

van Heuven, Walter J. B., Pawel Mandera, Emmanuel Keuleers & Marc Brysbaert. 2014. SUBTLEX-UK: A new and improved frequency database for British English. *Quarterly Journal of Experimental Psychology* 67(6). 1176–1190. doi:10.1080/17470218.2013.850521.

Vanhove, Jan & Raphael Berthele. 2017. Testing the interdependence of languages (HELASCOT project). In Raphael Berthele & Amelia Lambelet (eds.), *Heritage and school language literacy development in migrant children: Interdependence or independence?*, 97–118. Bristol: Multilingual Matters. doi:10.21832/9781783099054-007.

Wolpert, David H. 1992. Stacked generalization. *Neural Networks* 5. 241–259. doi:10.1016/S0893-6080(05)80023-1.

Yeo, In-Kwon & Richard Johnson. 2000. A new family of power transformations to improve normality or symmetry. *Biometrika* 87. 954–959. doi:10.1093/biomet/87.4.954.