Academic Formalities

Student Information

Information	Description
Name	Christopher Michael Chandler
Matrikelnummer	108017107247
Erstfach	Linguistik B.A, 6. Semester
Zweitfach	Romanische Philologie Franzoesisch B.A, 9. Semester
Titel	From T'es Qui to Qui Es-Tu: A Naïve Bayesian Approach to Assessing Literate and Oral Discourse in Nonstandard French Language Data
Kurs	Schriftliche Hausarbeit für die Bachelorprüfung der Fakultät für Philologie an der Ruhr-Universität Bochum (Gemeinsame Prüfungsordnung für das Bachelor/Master-Studium im Rahmen des 2-Fach-Modells an der RUB vom 03. November 2016)

Preface

The layout of this program was partly inspired by a computational linguistics project called " Erkenner von Unicode-Emojis und ASCII-Emoticons sowie entsprechender Neubildungen" that I did with a partner in the summer semester of 2020 (Chandler & Cultraro, 2020). The goal of that project and its program was to recognize emoticons, both old and new, and emojis within a given text. The inspiration, and to a certain extent the contribution, for this program is therefore limited to the dynamic menu system, file structure and control structure that is used within this program.

As this is a lingistic project, there are files pertaining to protoypical sentences and texts that are representative of the conceptual discourse types ,i.e., literacy and orality (Cook, 2012; Goudailler, 2002; Müller, 1975). There are three primary data sets that are the focus of the linguistic analysis: eBay petites annonces, or EPA, (Gerstenberg & Hewett, 2019), CMR-wikiconflits (Poudat, Grabar, et al., 2015), and 88milsms (Panckhurst et al., 2016).

Please refer to the references for information on how to obtain the specific corpora used within this project.

References

Chandler, C., & Cultraro G. (2020) *Erkenner von Unicode-Emojis und ASCII-Emoticons sowie entsprechender Neubildungen [Gray literature]*.

Cook, J. (2012). Les marques lexicales du français familier dans la traduction polonaise des dialogues romanesques. *Traduire, 226*, 93–107. https://doi.org/10.4000/traduire.162

Gerstenberg, A., & Hewett, F. (2019). *A collection of online auction listings from 2005 to 2018* (anonymised) [Data set]. La-bank: Resources for Research and Teaching. https://www.uni-potsdam.de/langage/la-

bank/ebay.php

Goudailler, J.-P. (2002). De l'argot traditionnel au français contemporain des cités. *La linguistique, 38(1)*, 5–24. https://doi.org/10.3917/ling.381.0005

Müller, B. (1975). *Das Französische der Gegenwart: Varietäten, Strukturen, Tendenzen*. Winter.

Panckhurst, R., Détrie, C., Lopez, C., Moïse, C., Roche, M., & Verine, B. (2016). 88milSMS. *A corpus of authentic text messages in French* (nouvelle version du corpus ISLRN : 024-713-187-947-8) (Cmr-88milsms-tei-v1) [Data set]. Banque de Corpus CoMeRe. https://hdl.handle.net/11403/comere/cmr-88milsms/cmr-88milsms-tei-v1

Poudat, C., Grabar, N., Kun, J., & Paloque-Berges, C. (2015). *TEI-CMC version of wikipedia discussions associated to the article "Quotient intellectuel"* (Cmr-wikiconflits-qi_discu-tei-v1) [Data set]. CoMeRe Corpora Repository. https://hdl.handle.net/11403/comere/cmr-wikiconflits/cmr-wikiconflits-qi_discu-tei-v1

Declaration of Authenticity

```
I hereby declare that the work submitted is my own and
that all passages and ideas that are not mine have been fully and properly
acknowledged.
I am aware that I will fail the entire course should I include passages and
ideas from other sources and present them as if they were my own.

Kamen, 14.12.2021
Christopher Chandler
```

Eigenständigkeitserklärung

```
Hiermit versichere ich, dass ich die Arbeit selbständig angefertigt, außer den im
Quellen- und
Literaturverzeichnis sowie in den Anmerkungen genannten Hilfsmitteln keine
weiteren benutzt und
alle Stellen der Arbeit, die anderen Werken dem Wortlaut oder dem Sinn nach
entnommen sind,
unter Angabe der Quellen als Entlehnung kenntlich gemacht habe.

Kamen, 14.12.2021
Christopher Chandler
```

Téki: A Naïve Bayesian French Discourse Analyzer

What is Teki?

The name 'Teki' pronounced as /Tɛki/ comes from a literal transcription of the informal phrase t'es qui' - 'Who are you'. It is reminiscent of Stromae's 2013 song - Papoutai - Papa ou t'es - lit. Father, you are where? It there contrasts with the more formal phrase "Qui es-tu". The spelling of the name is meant to represent this. This form of topicalization is actually quite common in French even if it is not necessarily exclusive to the language itself.

The name was chosen to reflect the process of researching the conceptual and medial representation of the French language. Message and intent are often communicated either through text or speech. However, these two only represent the medium of the language. The difference between the conceptual and medial representation of language is often overlooked when assessing language. The phrase "someone writes how they speak" reflects this mentality. The train of thought is translated into a written medium, but the words actually reflect a stream of consciousness if you will.

Teki is a naive bayes classifier that is used to tag French chat data according to their literacy and orality. Since there was no training data available for the naive bayes, a scoring system was developped that tagged sentences according to their literacy and orality. Then, using this newly acquiered training data, the naive bayes was trained so that it could probabilistically decide if a sentence was representative of literacy or oralality.

For more information on this topic, please consult the program documentation located in the app_program_documentation folder.

License

MIT License

Program Requirements

Python Version

This program was created and designed with **3.9.6** in mind and is therefore the recommended version. It is possible to run the program using any version **above 3.6**, but program stability cannot be guaranteed.You can download the latest python version at the official website.

Modules and Libraries

standard

This program makes use of the following standard libraries:

```
csv
datetime
json
logging
os
re
shutil
statistics
sys
timeit
tkinter
```

Pip

In addition to these, the following pip packages must be present in order for the program to run properly:

- spacy>=2.3.5
- beautifulsoup4>=4.9.3
- bs4>=0.0.1
- lxml>=4.6.1
- pandas~=1.1.5
- sklearn~=0.0
- scikit-learn~=0.24.2
- future~=0.18.2

There are three ways to install the *requirements.txt* file:

1. Install all modules individually via the pip installer .

```
pip install module_name
```

2. Install all modules with pip using the requirements.txt without specifying the version number:

```
pip install -r requirements.txt
```

3. install with pip using the requirements.txt with specifying the version number:

```
pip3.9 install -r requirements.txt
```

In addition to those, the following for spacy must also be installed separately:

```
python -m spacy download fr_core_news_sm
```

If problems arise with any of the installation, please consult the respective module repositories Please also ensure that you have enough disk space so that the extra modules can be installed.

Program

The python file that is to be run is called ***teki_main_app.py***.

File check

When starting the program, a check is performed to ensure that the directory **app_program_resources** and its files are all present. These files should not be changed in any way. If you decide to add databases or corpora to this program, they should be placed in **app_user_resources**.

```
The current time is 06:17:32.
Please wait while libraries, modules and corpora are being imported...
This should take between 5 - 30 seconds depending on your system resources...
```

Failure to start

If the files have been moved or corrupted, then the following prompt will be displayed to the user.

```
The app resource directory is either missing,
has been renamed or the file has been altered in some other way.

0 yes
1 no

Would you like to continue with the program?
```

The user can continue using the program despite the program not having access to all files. However, program stability cannot be guaranteed. Please consult **teki_error.log**, which is located in the main directory, in this case if program instabiltiy does arise.

Program Start

If the modules, files and libraries have been successfully loaded, the user should be greeted with the following:

```
All libraries were loaded 9 seconds. The program can now start.

You are currently using the app_common_default_docs files:

Default Text: 'app_program_resources/default_files/mueller_oral.txt'
Default Training: 'app_program_resources/default_files/default_training.csv'
```



```
This is a dynamic menu to which the user can return. Should an error arise in the
program,
the user will automatically be redirected to this menu.
```

1 load .XML or .TXT file

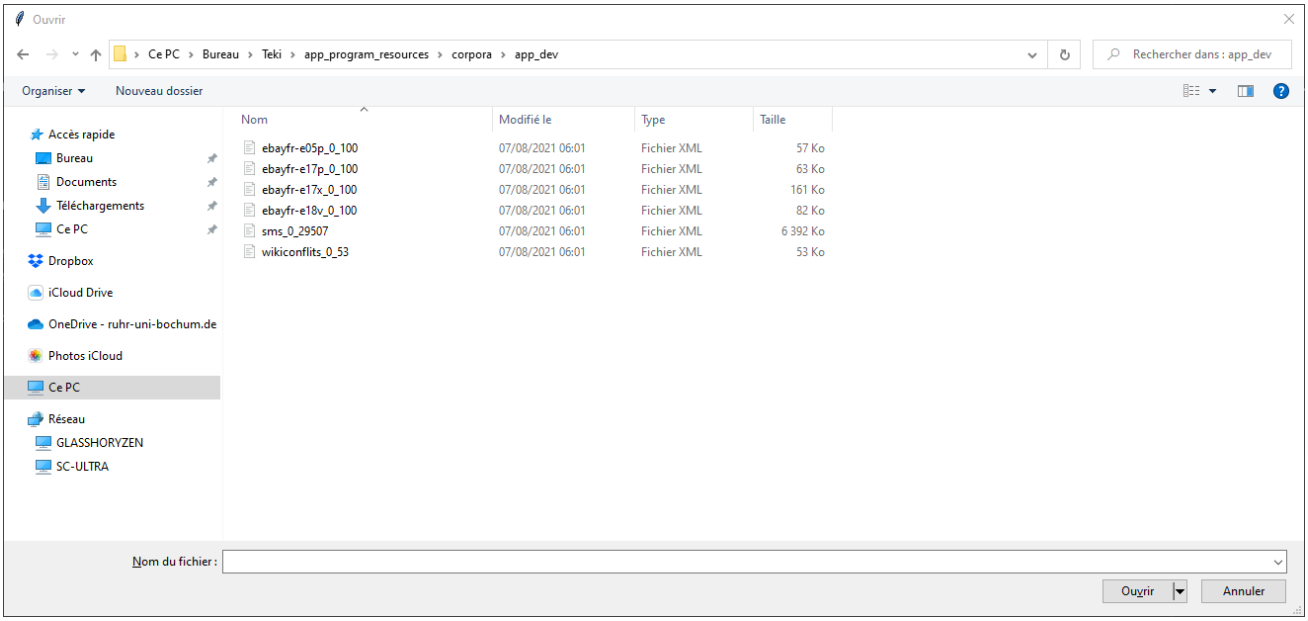
When first starting the program, the program will run using the following default file:

```
Default Text: 'app_program_resources/default_files/mueller_oral.txt'
```

If the user would like to change this, then it is necessary to choose a new file. The program comes with corpora and it is designed to be used with these corpora. Simply click on the .xml file that should be read into the program. The subdirectories containing the .xml files can be found in the **app_program_resources** under **app_corpora**.

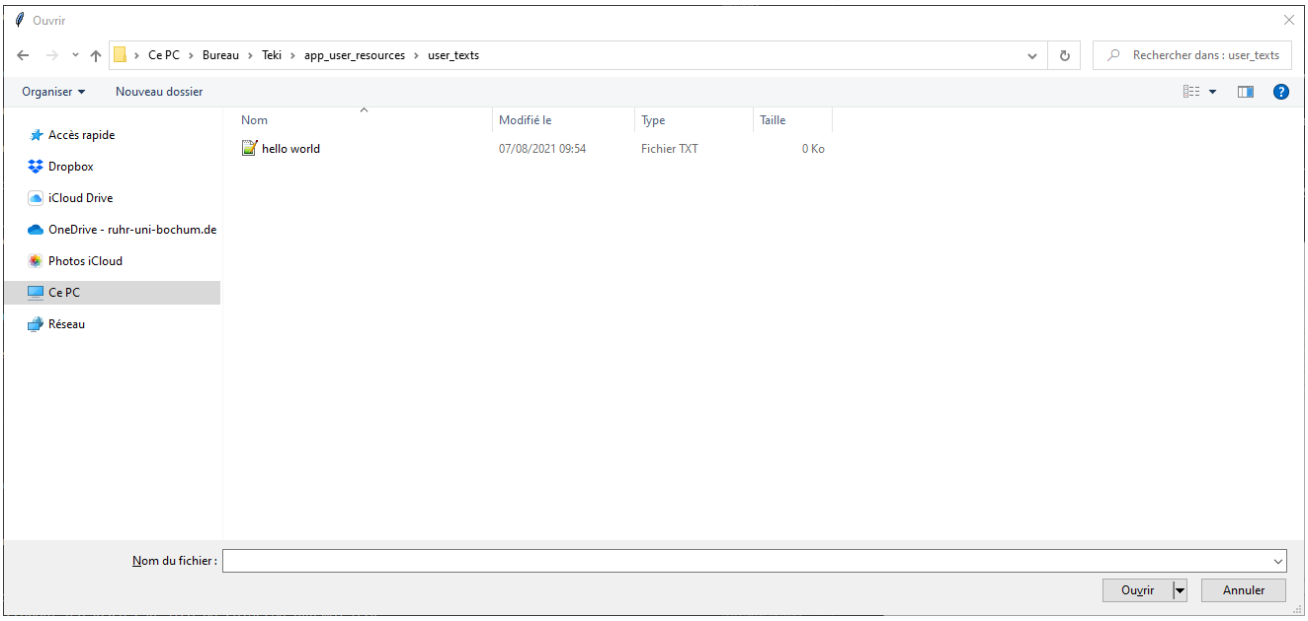
.Xml

The user will be presented with a file prompt to select the desired file:



.Txt

The user can select a desired .txt file from the **muller_corpora**, which is also located under **app_corpora**.



Alternatively, a user can input any .txt file regardless of its location.

⚠ **Note** The file must have .txt or .xml ending. Also, it must only contain text and tags (if .xml) or only text (if .txt)

Other file formats such as .pdf, .doc, .xml, etc. are not supported and could lead to the program unexpectedly crashing.
⚠

After having selected a file, the file will be loaded into memory and the user will be returned to the main menu.

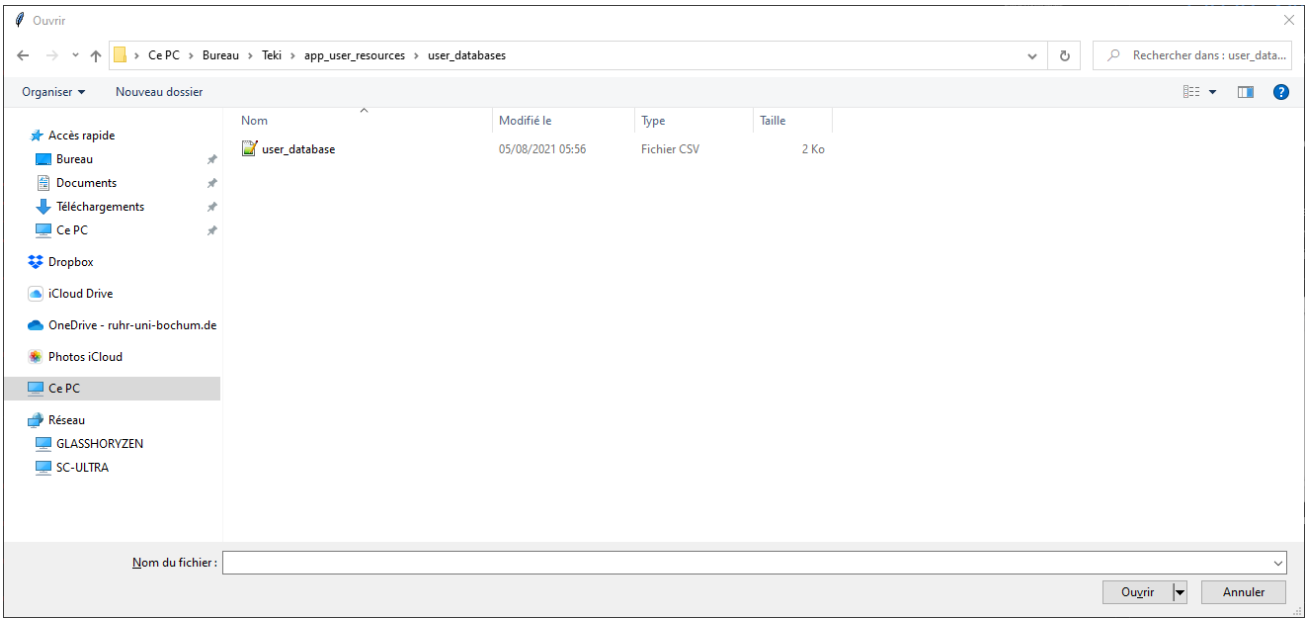
2: load training file

```
Default Training: 'app_program_resources/default_files/default_training.csv'
```

As with loading a .txt or .xml file, if the user does not wish to use the default database, a different database can also be selected dynamically. Default databases can be found under **app_program_resources** under **default_files**

If the user has trained and saved databases, then they should be under **app_user_resource** under **user_databases**. If no custom databases currently exist, simply create an empty .csv file and then load it. The results of the training will be saved there.

After having selected a file, the file will be loaded into memory and the user will be returned to the main menu.



All database files must be saved in a .csv format and have the following structure:

Token	POS	Dependency	Sentence Id	Corpus Id	Feature
ENSEMBLE	NOUN	ROOT	SEN:0	e05p-001	LIT

After having selected a file, the user will be returned to the main menu.

3: analyze contents

After having loaded in the desired files or progressing with the default files, the user is greeted with the following submenu:

Content Analysis
How would you like to proceed with the file?

- 1: read file contents
- 2: analyze .XML data
- 3: analyze .TXT data

- 4: return to main menu

1: read file contents

This displays the content of the .xml or .txt file to the user. After which the user is brought back to the submenu.

2: analyze .XML data

There are three main corpora available within the program:

```
1 eBay
2 SMS
3 Wikiconflict
```

The program could theoretically work with any .xml file. The programm has been desigend to work with these three data sets in mind. Working knowledge of python would allow for one to easily alter the .xml parsing process within the programm. To do so, please refer to function xml_analysis within the python code. However, it should be noted that these three corporora provide at least ca. 100,000 sentences and offer a wide arrange of possibilties with respect to linguistic analysis. Therefore, they should suffice for the average user.

Once a data set has been chosen, the user will have option of inputting a range of tags to be extracted from the corpus.

```
There are 100 tags. Please enter a selection range from 0 - 100.
 A range should be specified as follows with a single space between both numbers:
start stop.
```

The speed and efficacy of this process is highly dependent on the system resources of the user, the python version installed and the range as specified by the user. The longer the range, the more time it will take for the program to parse and analyze everything correctly and efficiently. It is therefore recommend that **the max range should be no more than 100**.

Example Range

start	stop
0	10

It is possible to enter a range that spans the entire corpus, but this is not advised as it could lead to program instability.

Proceeding with the corpus

- 1: process sentences

- 2: save unprocessed
- 3: return to menu

1: process sentences

Processing the sentences means that the spacy tagger will analyze them to retrieve linguistic relevant information. If the user wishes to train or build up their own database, then they must proceed with this option.

Once a range has been entered, the sentnece will then parsed, analyzed and assigend a unique identifier. Afterwards, the following option selection will appear:

```
The sentences have been successfully processed.
Please press enter to continue...
1 automatically
2 manually
```

In both cases, a unique identifier will be assigend to each sentence so that it can be found in the corpus, should the user wish to manually inspect the database. This identifier is also available to the system and necessary so that it it can correctly read in the databases later.

1 manually

If the user already knows the features or wants to specify the features that should be assigned to the respective sentences, the user may do so here by using the following features.

1. LIT
2. ORAL

The feature will then be applied to every sentence within the corpus as previously selected by the user. Therefore, this feature should only be used if the user knows that all sentneces are of the same kind, i.e. all sentence should receive the same feature.

2 automtically

This is the prefered method as the program using as a scoring system to guage the most appropriate feature and assign it to each sentence respectively. The feature LIT or ORAL will the n be assigned my by the program itself.

For more information on the scoring system and how this works, please consult the documentation.

2: save unprocessed sentences

If the user is simply interested in retrieving the sentences retrieved from the corpus, then the user should proceed with this option. The user will be promoted by file dialog to select the file in which the results should be saved.

If file does not exist, the user can create the file from within the dialog window. Once it has been created, the user must click on this file and then press 'open'. This will save the path file name and pass it on to the program.

The user will then be returned to the main menu, where the user can safely exit the program. The results will then be in the file specified by the user.

3: Return to menu

This will bring the user back to the main menu.

3: analyze .TXT data

This process is largely similar to that of the .xml process as described above. The only real difference here is that the user must specific a unique indentifier that is to be assigned to this text. Note that this identifier is not the same as the feature, but rather an indicitator so that the user and the program can identify from corpus the sentence came from.

3: return to main menu

If the user chooses to analyze an .xml file, then the user has three options available

4: document classification

Once a databse has been trained, it possible for a document to be classified by the naive bayes formula:

```
The database contains the following documents:
LIT: 30
ORAL: 24
```

- 1: enter a sentence
- 2: enter a document
- 3: return to main menu

1: enter a sentence

If the user wishes to simply analyze just one sentence, then please use this function. The user will then be prompted to enter a **French sentence**. It is possible to enter something that is not a French sentence. However, these answers are to be ignored because they simply rely on default values when the program does not know which exact response it should give.

After having successfully entering a sentence, the program will return an answer based on the training data available to it. The more reliable the training data, the more reliable the answer will be.

The user will be returned to the submenu and will have the option of entering a sentence again.

2: enter a document

This will pull up the analyze contents menu and it follows the same process. The user should consult the chapter if the user has not already done so.

Due to the problem of underflow, a document as a whole cannot be properly anaylzed. That is why the document will only be analyzed on a sentence by sentence level. The program will return the most approrpraite feature per sentence, not necessary per document.

After the analysis, the files will be saved in **naive_bayes_results**. the user can chose to exit the program or continue with other sentences/documents.

3: return to main menu

This will return the user back to the main menu.

5: clear error log file

Should unforseen errors arise within the program, they will be saved within **teki_error.log**. Errors can occur due to :

1. not reading in files
2. reading unsupported files such as .pdf, .doc(x)
3. altering the app program directory
4. altering the programs source code
5. Missing libarries or modules

⚠ **Note:** this function will erase your error report and the data of which cannot be recovered. ⚠

6: restore default database

The user may build upon the default database as provided with the program. However, if this databsse becomes corrupt or becomes inbalanced due to improper training, the user may reset the database using this function.

⚠ **Note:** this function will erase all of your progress and the results of which cannot be recovered. ⚠

7: evaluation

The user has the option of ascertaining the evaluation metrics of the naive bayes or cross validating the data used to train the naive bayes.

Which files would you like to evaluate:

- 1: evaluate naive bayes
- 2: cross validation
- 3: return to main menu

1: evaluate naive bayes

This evaluator takes in two files: a system file and gold file.

The system file are the sentences that were produced by the program. This is not the same as the database files created by the system.

The format of both files should be as follows:

Sentence	Sentence Number	Corpus ID	Feat
A noter que le film n' est pas encore sorti en dvd en France !	SEN:4	e05p-005-4	ORAL

The gold file must be annotated by hand as the program is currently incapable of generating one.*

Once both files are present, the user will be prompted to first select the system file and then gold file. The programm will compare the two files and return the following values:

```
Accuracy
Error Rate
Precision
Recall
F-Score
```

2: cross validation

The cross validation works the same way as the evaluate bayes, but it only requires one file to work. using the sentence file generated by the document classification or analyze content, you can validate the data for overfitting.

Once a file has been seleteced, the user must place

```
sentence,sentence_number,corpus_id,feat
```

at the top of this file in order for it to be read properly. Once having done so, the results of the cross validation will be shown within the program.

3: return to main menu

This brings the user back to the main menu.

8: about program

This function provides the user with this readme file. The user can therefore access the program information from the menu.

9: end program

You can force exit the program by killing the script, but this could lead to file corruption. Therefore, using this function is the preferred way of exiting program because the results from the other functions are only saved to the respective files once you have properly exited the program.