Beubli project

[**Notebooks**](#_if08yw3lyxr4) **3**

[First parse and varied tests](#_9aq46prqogd5) 3

[General Description](#_uo0rnfe6qx2v) 3

[Conclusion](#_alm6rklyhnzo) 3

[First large “iel” replacement](#_yqpyfd3twcb4) 3

[Steps taken](#_q5himxu60kat) 3

[Conclusion](#_gj8n9jekyxxc) 4

[Notebook V3](#_5yp4cp1qj75h) 4

[Steps taken](#_1xb5p56su3lu) 4

[Conclusion](#_gzbu7mxj6iid) 4

[Notebook V4](#_kf0mzrf5090i) 5

[Notebook V5](#_gk5ffnsci2c8) 5

[Steps taken](#_tbwdj0vbxcuo) 5

[Conclusion](#_o4391739sjnn) 5

[Notebook V6 (Camembert V1)](#_qxpg32mjd3g) 5

[Notebook V7 (Camembert V2)](#_2799lq6x5yja) 6

[Steps taken](#_43go78d0fhnh) 6

[Some graphs](#_ny3kubnx6vnn) 7

[Conclusion](#_nh0ibbxu2zo3) 8

[Notebook V8 - Translation transformers](#_4aszox9aub5j) 8

[Steps taken](#_arily5tanxc3) 9

[Conclusion](#_ow67yl15e9xi) 9

[Notebook V9 - Complete translation](#_3m526zdm6f5i) 9

[Step taken](#_q90iqv35h7zp) 9

[Conclusion](#_kfixouoh6g2s) 9

[Notebook V10 - feature extraction from bible translation](#_klr9i7gpjlby) 9

[Steps taken](#_6jgdqip6o3o2) 10

[Conclusion](#_phbch9jjspz4) 10

[**Notebook V11 - gender identification of a proper noun**](#_4p8ardljfs6o) **10**

[Goal](#_6qrj3bcuti76) 11

[Steps taken](#_2cg0g2sep6lx) 11

[Conclusion](#_gaf09vvglq9t) 11

[Notebook V12 - gender identification of a proper noun part 2](#_nbzi22vkyxo) 11

[Goal](#_bleszs99qs5s) 11

[Steps taken](#_nplqs66ij0bl) 11

[Conclusion](#_igryrggfyjfs) 11

[**Fun stuff**](#_hqunppwcdyok) **12**

[Gonorrhée](#_av3d2hnphk3g) 12

[**Sources**](#_amodei9ba4qs) **12**

[Treetagger](#_ap54bt498vml) 12

[tutorial](#_p1aks72rn3a2) 12

[french resources](#_vufwyzhvo2iq) 12

[Word2Vec](#_j915q3hjls4s) 12

[french word2vec embeddings](#_48rbx0fjeib2) 12

[FastText](#_idr6tiz28x94) 12

[french word2vec embeddings](#_gaj33amtq0rb) 12

[CNRS Ressources](#_hhym3wz2bya0) 13

[Feature descriptors for French POS tags](#_a5mvq0ra2ufa) 13

[Misc. Ressources](#_egq7ytz258w7) 13

[Misc. Github and data resources](#_qdhlt2j2ussk) 13

[CamenBERT](#_bahlqbellj82) 13

[Wikipedia (incl. french) POS tagged dataset](#_7k431ctikkl1) 13

[pygrammalecte](#_f2ze1l1rd9kw) 13

[Large french dataset (LeMonde) POS tagged](#_tnzeybs0bpr8) 13

[PRO:PER : Neutral ?](#_x1b7ps34wtjc) 13

[Coreference computation](#_8aig71aghlx0) 13

[Iel PRO : French language rules](#_7ipx912grbn5) 13

[Useless article on the impersonal form](#_greqgqgul8p0) 13

[Interesting NLP “Tools and Techniques” overview](#_9q1bu18f1bu) 14

# Notebooks

## First parse and varied tests

Link to notebook:

<https://colab.research.google.com/drive/1GEq_fCV4e9ieCOXE8Rph-U72QLW8pyKK#scrollTo=swLRczdkDQWt>

Link to related Gdrive folder:

[la beubli](https://drive.google.com/drive/folders/1hngnZsIyXnjCYzfbI1GaJqELjkPRwJGD)

### General Description

We tried the following techniques:

* NLTK for sentence POS tagging
* TreeTagger set to french
* pygrammalecte to check french grammar
* translate text (Google translate) from french to english

### Conclusion

All failed as a single step:

* NLTK default french model doesn’t differentiate between neutral and gendered PRO:PER in french text data.
* TreeTagger doesn’t differentiate either.
* check for grammatical errors when performing the PRO:PER switch in a sentence and detect the error level as a gender score for each sentence: also failed
* translate the sentence: we cannot query all sentences, mapping fails sometimes

However, we combine these techniques in the NotebookV2 below to handle most edge cases.

## First large “iel” replacement

<https://colab.research.google.com/drive/1U2N7ZyeFBnXw78yHORi6MRHk5kO4nLPy#scrollTo=9c8j-KvfQOf6>

### Steps taken

* install and import all the libs tested in previous notebook
* define functions that use the libs (tag sentence, translate sentence, etc…)
* define higher order functions (is\_transform\_possible, cut\_into\_sentences, etc…)
* create one big “main” function that transforms each page

### Conclusion

There are several drawbacks:

* The current system only replaces one pro:per per sentence, when there are sometimes up to 4. This would mean running the system 4 times to have 100% coverage
* Each file needs to be tagged, translated. All intermediary data is lost, so recovery is costly (TreeTagger can be multithreaded on its own).
* The current pipeline stopped after 331 pages due to Google Translate api failure
  + maybe store all api requests and try to approximate results
  + maybe train a NN to find correct pro:per from augmented dataset of tagged gender neutral sentences

The next big step is building a complete french bible annotated dataset.

## Notebook V3

Link to the notebook:

<https://colab.research.google.com/drive/1nmM_9k7vNqIi614MDVsQ7kPfaRT8edNe#scrollTo=8zhf9PVb4i6N>

Link to the related files:

[complete\_french\_bible](https://drive.google.com/drive/folders/1OIKR1aqyA8cJiIrJxS2-QP_QOSlnbQMB)

Link to the GSheets:

[complete\_bible\_french](https://docs.google.com/spreadsheets/d/1JejKmyhmwAp-1vwe5fELu8CQfH6mCmPkowJE1pOV7oM/edit#gid=1292979474)

### Steps taken

We extract data from the text files and metadata from the filenames, then export all text content into a single dataframe, with the following data structure:

* chapter number (int)
* page number (int)
* line number (int)
* content (string of a given sentence)

### Conclusion

We can freely explore the whole text in a Google Sheets file associated with the exported .csv file. There are several replacement scénarios that are listed below:

* elle / elles / Elle / Elles => replace by iels / Iels
* ils / Ils => replace by iel/ Iels
* il => translate to know whether the replacement is appropriate

As the string replacement (from il/elle to iel) will make the final text have a parsing issue, we will apply this change in a later version (or rendered version only).

Next step: extract the most problematic sentences and test a more thorough replacement technique.

## Notebook V4

<https://colab.research.google.com/drive/1Ei3W-pyTf8VI4xfMrqikPoitHFQ1U5EU>

Tag the complete bible into JSON and PIckle format files for later use.

## Notebook V5

Link to notebook:

<https://colab.research.google.com/drive/1CX7-0_FHrstQZFPM97si3Dp2m-K0P4XV#scrollTo=qxJWlDJMwF7d>

Try to translate the complete notebook in English, discover that the free Tier quotas are really low.

### Steps taken

* count number of words and chars in the complete bible, 4M chars for 30k daily limit
* Create a sentence level dataset with a new feature to identify (sentence number). As the max sentence number is 8, we can split sentences into 8 new columns, then pivot the whole dataframe longer by transforming the 8 additional columns into 8 labels of a single key:value column pair.
  + the idea is that simple sentences will be easier to match to their translation
  + we can drop some duplicates and reduce the total number of translations needed
* We also filter on “il” sentences, as the known cases (elle/ils, etc…) do not need to be translated

### Conclusion

STILL IN PROGRESS/NOT FINISHED - Must filter the dataset to have less than 500k chars to translate, then map impersonal forms of pro:per to sub-sentence words.

## Notebook V6 (Camembert V1)

What about French BERT ? (ie. “CamemBert”) => Could we identify sentence structure and change the sentence root subject gender in an organic fashion (grammar would be compatible, some words might be missing but general sense is preserved).

We should be able to change the gender of a sentence, from masc to fem. If this step is possible, then we could change all genders to masc on one version, then all the genders to fem in a second version. The final version would replace all pro/per changes by the pro:per “iel”, and concatenated word-endings so (“levé”+”levée”=”levé.e”).

To map the changes and concatenate words, we need to generate text that is very close to the original, maybe only predict words from certain POS, in certain positions.

Link to notebook:

<https://colab.research.google.com/drive/1916f2S6VvhIjFXF5NS98-Ej83vcBX2bg#scrollTo=mFKWe-CFpT9I>

## Notebook V7 (Camembert V2)

Link to notebook:

<https://colab.research.google.com/drive/1DmoUwpIFNO5zKXQPS3t-Fe1Zuu9zZgLx#scrollTo=6AESED_aENMQ>

### Steps taken

1. libs and models import
2. funcs to add masks to sentences on PRO:PER
3. funcs to create all possible variations of masked sentences
4. camemBert func to guess a masked word
5. compute 5 guesses per mask
6. apply to 1000 samples
7. try Kmeans on:
   1. several types of dimension reduction/manifolds:
      1. raw TF-IDF
      2. PCA on TF-IDF
      3. t-sne on TF-IDF
   2. compare results between clusters and dim reduced data (cluster first, reduce and plot labels as hue after to confirm)
8. Generalize silhouette analysis function and test K values from 2 to 24
   1. no K Means solution is optimal, always issues
   2. another clustering algorithm should be tested
   3. dimension reduction produces vastly different results, we should analyze explained variance per dimension to get a sense of loss
9. Manual sorting of neutral sentences
   1. add TreeTagger POS to CamemBert output tags and count Nouns VS PRO:PER occurrences
   2. count points per certain PRO:PER( +1 for “il”, -1 for “elle”) and compute derived score of neutrality
      1. doesn’t separate well
      2. isn’t really working (80% match only)

We currently do not have a real way to measure neutrality in a sentence.

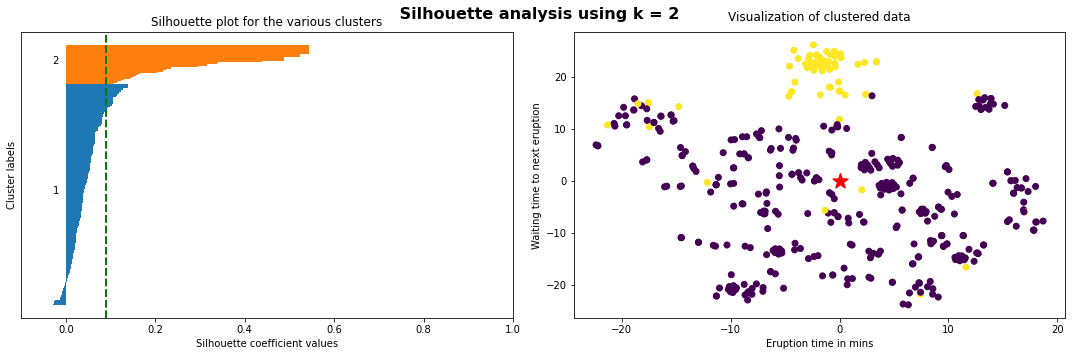
The dataset is exported here:

[camenbert\_masked\_dataset](https://drive.google.com/drive/folders/1V9FewSgc54dQgiiafHrycY0-IDk6QnKa)

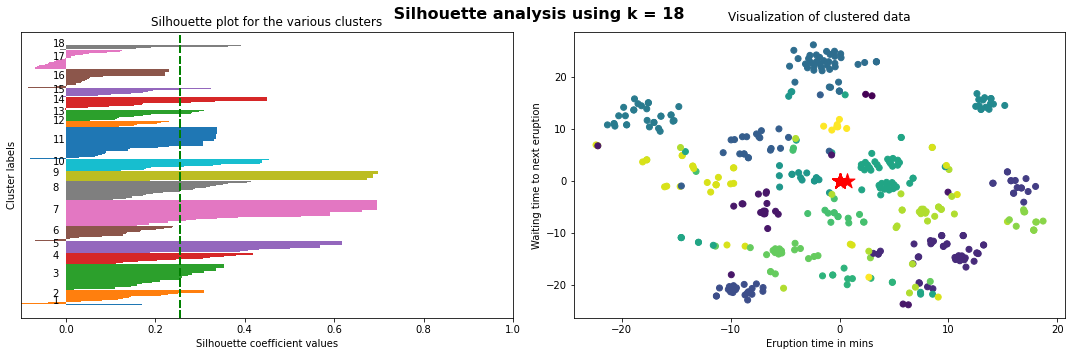
### Some graphs

In respect to Kmeans, the silhouette analysis shows that the most stable configuration is either K=2 (with a simple minority cluster) or K=18, with very small and specialized clusters. Either way, Kmeans is not well suited for this task as the dataset should be balanced: Silhouette is expecting homogenous blobs with roughly equal cluster cardinality.

We will either change metrics, or change the clustering algorithm.

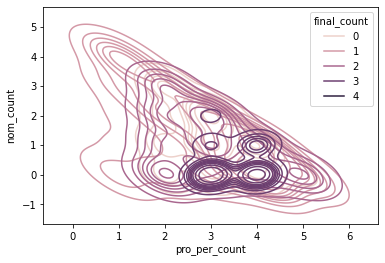


*Kmeans silhouette analysis for K=2*

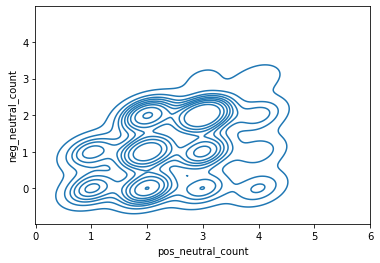


*Kmeans silhouette analysis for K=18*

When we compare the KDE (Kernel Density Estimate) plots of our handcrafted variables against the POS parsed by TreeTagger, we can see that our score doesn’t represent anything, while the POS parse reveals concentration of POS tags.



*GOOD : TreeTagger POS PRO:PER and NOUN counts*



*BAD : Derived score from handcrafted rules doesn’t discriminate*

### Conclusion

CamenBert offered a new way to guess grammatical gender by offering more neutral PRO:PER in the case of a neutral sentence, and more gendered propositions, as well as alternative noun subjects, when a gendered sentence is detected.

Bert&Co models do not seem to understand grammatical gender as a concept but will suggest the correct gender if a sentence is gendered.

## Notebook V8 - Translation transformers

Link to the notebook:

<https://colab.research.google.com/drive/1SbkhBr3ZrARDPjZAN6ZlgwLAjCIkqrsb#scrollTo=iUkCcxMgKLJu>

Link to fr -> en transformer:

<https://huggingface.co/Helsinki-NLP/opus-mt-fr-en>

### Steps taken

1. loaded fr to en transformer
2. translated 100 sentences from the french bible
3. loaded en to fr transformer
4. translated the 100 sentences back to french
5. compared results => some inconsistencies
   1. punct is preserved
   2. misplaced words changing sentence meaning
   3. bad tense use
6. create gender swap function (simple regex + double translation)

### Conclusion

Takes too long to process. On gender swap takes around 2 to 4 seconds.

We will have to find a faster way (batch process, multi-threading, etc…) to translate the complete dataset.

## Notebook V9 - Complete translation

Link to the notebook :

<https://colab.research.google.com/drive/1RMcXHOW1D0764UrTORZW5jD7x8XMNeN0#scrollTo=z68D0LUP_Fte>

### Step taken

1. Import transformer and usual libs
2. Import sentence dataset
3. Use transformer pipeline to make a faster translate function
4. Translate the complete french bible

### Conclusion

The default translation function would have taken around 14h to translate the bible, whereas the faster version, using a pipeline, only took 5 hours.

## Notebook V10 - feature extraction from bible translation

Link to the notebook:

<https://colab.research.google.com/drive/1tuB_ixmiigdxXTV4ZyThI2Sx6HSV77KE#scrollTo=m6d_Xr_8ht7H>

### Steps taken

1. Import and load Flair POS and NER models
   1. POS is part of speech, same job as TreeTagger and NLTK, but faster and multi-language
   2. NER is named entity recognition, we used it to extract named persons and places for later use
2. Add NER and POS extractions as new features to the sentence dataset
   1. One dedicated POS extraction, then a NER extraction
   2. 30 mins run time each
3. Add the following features:
   1. PRO:PER count for masc and others in French
   2. PRO:PER count for masc and neutral, as well as others, in English
4. Export new sentence dataset to Gdrive

### Conclusion

The newly exported dataset now contains the following features:

* Several keys :
  + Index (global page count)
  + Chapter number
  + Page number
  + Text line number
  + Sentence number
* Original french sentence
* English translation
* NER extraction
* PRO:PER features in FR/EN
  + il/he/there count
  + other PRO:PER counters (elle/s, ils, it, she, etc…)

The next steps will consist of handling the proper nouns (identify gender to swap later) and flagging the sentences where an ambiguous PRO:PER splits into neutral forms in the translated sentences (“il” -> “there” instead of the expected “il” -> “he”).

## Notebook V11 - gender identification of a proper noun

Link to the notebook:

<https://colab.research.google.com/drive/1OXbOOs67GqdgeJvp5k6rkrlMbiVhzHs7#scrollTo=odnNikUM40Tg>

### Goal

Identify the gender of a given proper noun.

### Steps taken

1. Import libs and dataset, as well as CamenBert (useless import)
2. Create a short list of proper nouns from the previous NER export
   1. Keep PER, MISC, and LOC names
   2. Filter on number of words and remove unwanted chars
3. Try to run NER a second time, as well as clean up again (steps 2a and 2b)
4. Try to run NER a third time, on the output of the second time
   1. More junk in the output
   2. Cleanup erases important data while second NER step cuts real words into made-up half-words

### Conclusion

NER is too dirty, we might be better off by re-doing the extraction step with a better NER transformer. Also, re-running a NER on a NER output performs very badly as the models do not recognize single tokens very well. Instead, the self-attention process heavily relies on global context, which is missing when the first extraction is performed.

## Notebook V12 - gender identification of a proper noun part 2

Link to the notebook:

<https://colab.research.google.com/drive/17-52um3yW0TCeTdb0B7pgN7XqdT_XXyR#scrollTo=bJSIqrS9OU6B>

### Goal

Find proper nouns in the bible and analyze their respective gender.

We couldn’t do it in the previous notebook because the NER was too dirty to be properly cleaned. This time, we will use a CamenBert based NER which is slower but more precise on french words. This will prevent the sentence pice spam we previously experienced. However, the miscategorization risk is still present (ig. God is tagged “MISC” instead of “PER”).

### Steps taken

1. Import libs and dataset, as well as two CamemBert models (NER transformer and mask transformer)
2. Extract NER from sentence once again (with GPU)
3. Select only PER tags (discard LOC, MISC, etc..)
4. Drop duplicates
5. Plot a few graphs to show that scores above 99% certainty are the most prevalent by far
6. Filter on scores above 90%

### Conclusion

This time, the extract went well and actually ran faster than the multilingual “fast” version. It seems that the GPU acceleration has a varied impact depending on the model chosen (from radical change to slight improvement over the default GPU).

We managed to run the complete extract in 4 minutes (instead of the previous 30 minutes). We then chose to filter the words that have a tag equal to “PER”, de-facto excluding mis-categorized words from further study. Same bias for the low scores and “strange chars” containing surnames.

In a further notebook, we analyze the proper noun shortlist (as it is now clean) to find the actual gender.

## Notebook V13 - gender identification of a proper noun part 3

Link to the notebook:

<https://colab.research.google.com/drive/19DySKOBVdqR3RuMNrWRISWgjZmAQ9khp>

### Goal

TODO

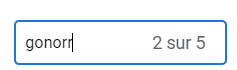
### Steps taken

1. Import libs and dataset, as well as CamemBert masking transformer

### Conclusion

# Fun stuff

### Gonorrhée





# Sources

## Treetagger

### tutorial

Install base + wrapper

[TreeTagger Python Wrapper's documentation!](https://treetaggerwrapper.readthedocs.io/en/latest/)

### french resources

Needs french resources to parse french POS.

[TreeTagger](https://cis.lmu.de/~schmid/tools/TreeTagger/)

<https://cis.lmu.de/~schmid/tools/TreeTagger/data/french.par.gz>

## Word2Vec

### french word2vec embeddings

[Jean-Philippe Fauconnier](https://fauconnier.github.io/#data)

## FastText

### french word2vec embeddings

[Word vectors for 157 languages · fastText](https://fasttext.cc/docs/en/crawl-vectors.html)

## CNRS Ressources

### Feature descriptors for French POS tags

<http://www.llf.cnrs.fr/Gens/Abeille/French-Treebank-fr.php>

### Misc. Ressources

<http://www.llf.cnrs.fr/en/resources>

## Misc. Github and data resources

### CamenBERT

[camembert-base · Hugging Face](https://huggingface.co/camembert-base)

### Wikipedia (incl. french) POS tagged dataset

[GitHub - mcoavoux/wiki\_parse: Tools for multilingual constituency / dependency parsing of wiki\*.](https://github.com/mcoavoux/wiki_parse)

### pygrammalecte

French grammatical python check lib

[GitHub - vpoulailleau/pygrammalecte: Grammalecte, le correcteur grammatical en Python](https://github.com/vpoulailleau/pygrammalecte)

### Large french dataset (LeMonde) POS tagged

[French Treebank](http://ftb.linguist.univ-paris-diderot.fr/)

## PRO:PER : Neutral ?

### Coreference computation

[Le calcul de la référence](https://halshs.archives-ouvertes.fr/halshs-00009780/file/Le_calcul_de_la_reference.pdf)

### Iel PRO : French language rules

[Le langage neutre en français : pronoms et accords à l'écrit et à l'oral – Genre !](https://entousgenresblog.wordpress.com/2017/04/19/quels-pronoms-neutres-en-francais-et-comment-les-utiliser/)

### Useless article on the impersonal form

This article discusses the issues but neither links source code, nor pseudo-code.

[ILIMP : Outil pour repérer les occurrences du pronom impersonnel il](http://www.linguist.univ-paris-diderot.fr/~danlos/Dossier%20publis/TALN'05.pdf)

### Interesting NLP “Tools and Techniques” overview

[Outils et plateformes pour le TAL](https://afia.asso.fr/wp-content/uploads/2018/01/Hamon_PDIA2017_TAL.pdf)