LINMA2472-Homework 2

Groupe 21
Cédric Antoine, Romain Graux, Lionel Lamy

November 2020

1 Introduction

This work has been realised for the course 'LINMA2472' (Homework 2). For the first part we have continued to use the book 'Harry Potter and the sorcerer' as we have done in Homework 1.

2 Part 1

2.1 Preprocessing steps

We first started by designing a dictionary listing and linking all the nicknames to the real names of the characters. Then we separated the book into sentences and, using the spacy module, we detected each character name and replaced it with the corresponding value in our dictionary. Finally we used several preprocessing methods from the gensim package (line sentence, strip tags, strip shorts,...) to clean up our data and return each sentence as a word list.

2.2 Word2Vec & k-means

With the preprocessing completed, we started building and training Word2Vec models. To do this, we defined for each parameter (size, window, epochs, alpha, as well as the number of training iterations) an interval of values to take and then left our computers running for more than 12 hours straight. In the end, we trained more than 3000 different models (more than 10 GB).

Once the training of all these models is completed, we were able to cluster them in two different ways with the KMeansClusterer class from the NLTK package : using an Euclidean distance or a Cosine distance.

2.3 Comparison of the clusters

With help of the Jaccard similarity algorithm we finally compared all our clustered models to the one obtained by the Louvain algorithm in the first homework.

Unfortunately, we did not manage to achieve a higher similarity than 0.212, which is really low. We think that this is due to an artefact of the Word2Vec classification method, because our data seems to be relatively clean. As a result, when we observe the clusters generated by k-means they seem to be quite distant from the real story of the book.

We can see that in the Figure 2, Harry Potter, who is the main character, does not end up with the people you would expect, such as Hermione, Dudley and Ron, who are his close friends in history. Most of the time in our trials, Harry found himself in a cluster where most of the members were not protagonists.. This could perhaps be explained by the fact that Harry is the link between all the characters and is difficult to classify in a group.

On the other side in the cluster of the Louvain algorithm (Figure 3), 'Harry Potter' is in a cluster with the majority of his friends. This seems to be much more plausible that the ones obtained by k-means.

2.4 Visualisation

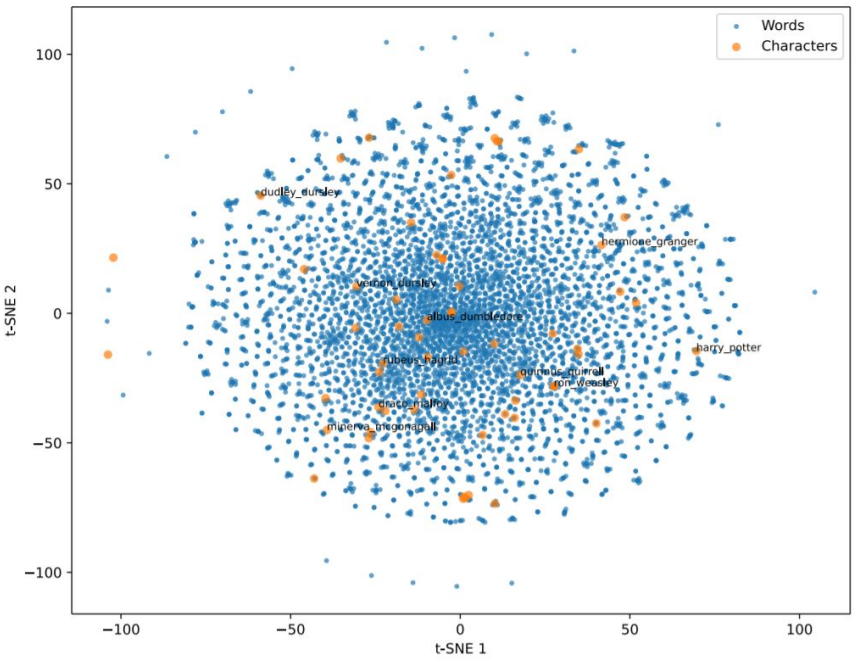

Figure 1: Embedding 2D visualisation

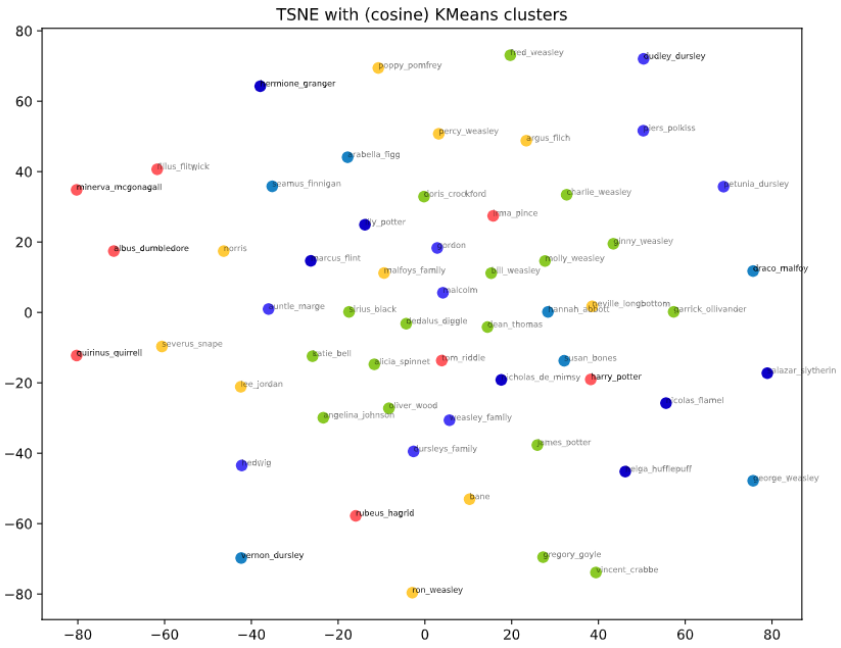

Figure 2: Cosine distance - Reduced to 2D using t-SNE

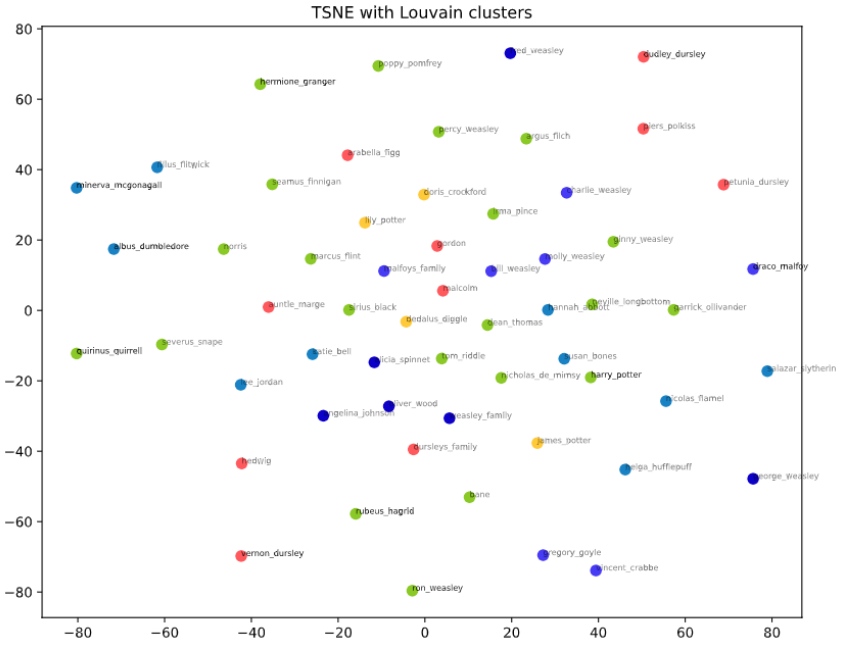

Figure 3: Louvain algorithm best partition - Mapped using t-SNE

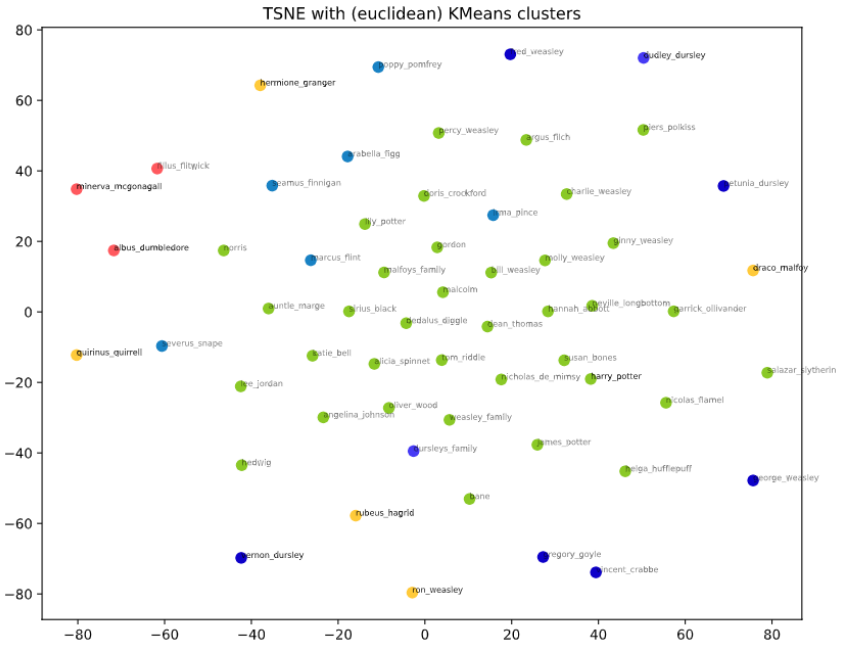

Figure 4: Euclidean distance - Reduced to 2D using t-SNE

3 Part 2

Abstract : all figures are available in html with more interaction, you just have to click on it to view them !

3.1 Sentence embedding

In this part, we will try to find the best embedding representation.
First let's try a Doc2Vec implementation.

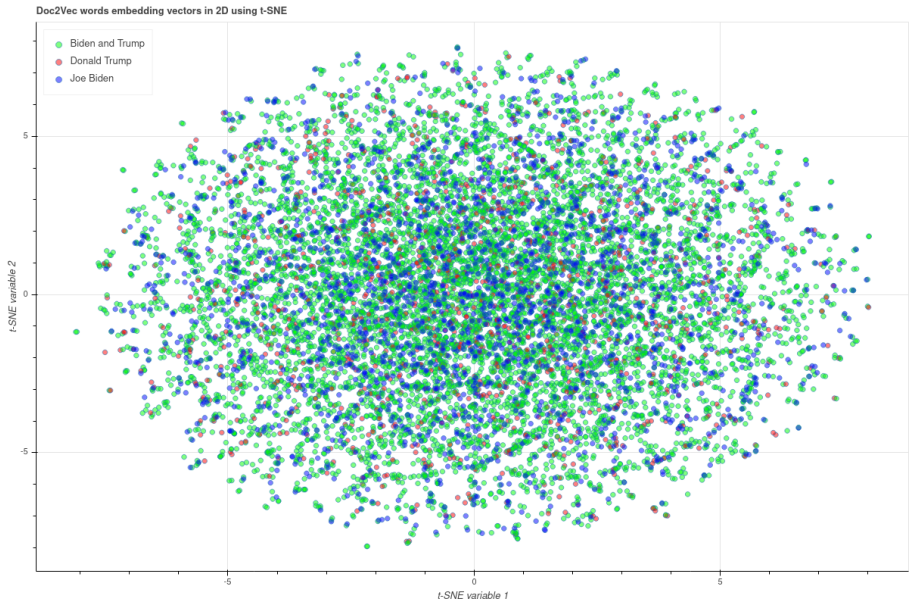

Figure 5: Words embedding with Doc2Vec model reduced to 2D using t-SNE

3.1.1 Doc2Vec

For the preprocessing, we have 3 different steps:

- Retrieve all desired entities with spaCy and merge all words contained in the entity with underscores
- Then clean the string with different filters such as :
  - lower cases
  - strip non alpha numeric values
  - strip punctuation
  - strip multiple withespaces
- Finally we take the lemma of each word with the WordNetLemmatizer from NLTK

And for the hyper parameters we have used:

- dm : 0
- $vector\_size$: 200
- $window$: 2
- min_count : 2
- $epochs$: 50

3.1.2 BERT

For BERT it's more simple because the model tokenizes and produces the sentences embedding, the one hyper parameters is that we have chosen *roberta-base-nli-stsb-mean-tokens* model.

3.2 Classification

3.2.1 Random forest classifier

For the random forest classifier, we have chosen a random forest with 100 estimators. With a random forest, we obtain these results based on Doc2Vec and BERT vectors :

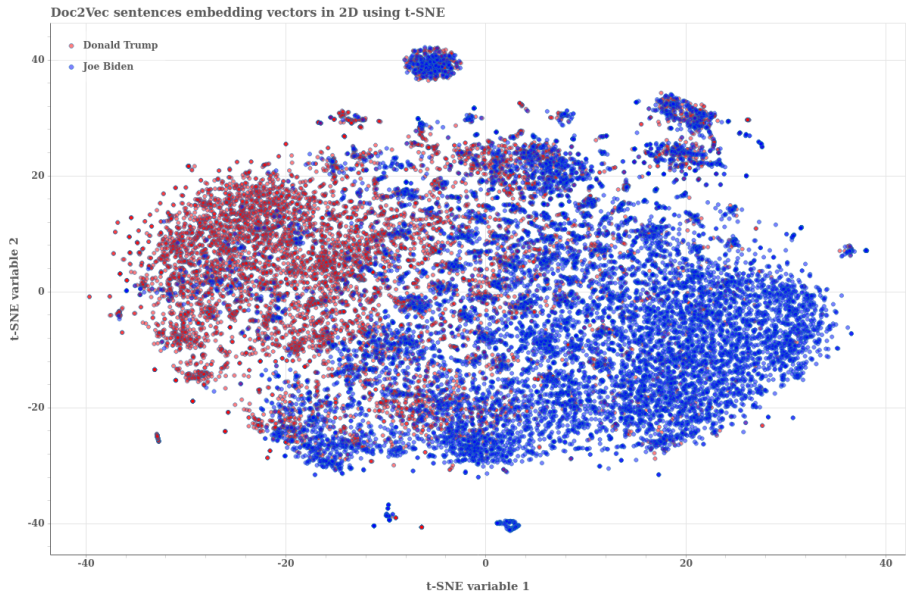

Figure 6: Sentences embedding with Doc2Vec model reduced to 2D using t-SNE

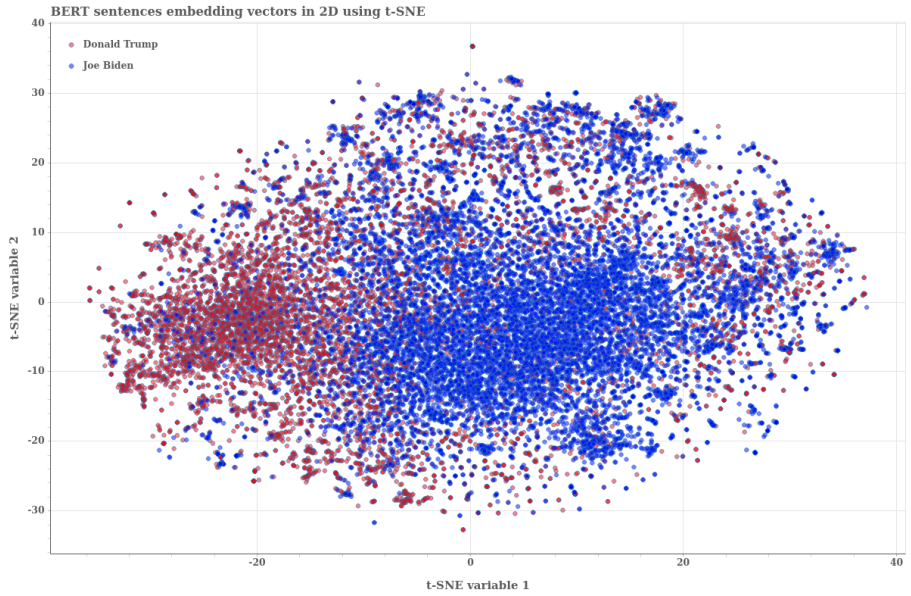

Figure 7: Sentences embedding with BERT model reduced to 2D using t-SNE

	Doc2Vec	BERT
Accuracy	.81	.79
F1	.81	.78
Precision	.83	.78
Recall	.79	.79

Table 1: Random forest scores on Doc2Vec and BERT vectors

3.2.2 Deep neural network classifier

The neural network is as follow :

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
dense (Dense)                (None, 512)               102912
_________________________________________________________________
dense_1 (Dense)              (None, 256)               131328
_________________________________________________________________
dense_2 (Dense)              (None, 128)               32896
_________________________________________________________________
dense_3 (Dense)              (None, 1)                 129
=================================================================
Total params: 267,265
Trainable params: 267,265
Non-trainable params: 0
```

And with this network, we obtain scores for both Doc2Vec and BERT of:

	Doc2Vec	BERT
Accuracy	.78	.78
F1	.75	.75
Precision	.65	.64
Recall	.88	.89

Table 2: DNN scores on Doc2Vec and BERT vectors