DataScientest

Reconnaissance Vocale - Système de traduction adapté aux lunettes connectées

Introduction au projet

Contexte

Les personnes malentendantes souffrent d'un problème auditif et se trouvent donc dans l'incapacité de communiquer aisément avec autrui.
Par ailleurs, toute personne se trouvant dans un pays étranger dont il ne connaît pas la langue se trouve dans la situation d'une personne malentendante.
Les lunettes connectées sont dotées de la technologie de reconnaissance vocale avec des algorithmes de deep learning en intelligence artificielle.
Elles permettent de localiser la voix d'un interlocuteur puis d'afficher sur les verres la transcription textuelle en temps réel. A partir de cette transcription, il est possible d'afficher la traduction dans la langue du porteur de ces lunettes.

Objectifs

L'objectif de ce projet est d'adapter un système de traduction au projet de lunettes connectées. Le système implémenté par ces lunettes permet de localiser, de transcrire la voix d'un interlocuteur et d'afficher la transcription sur des lunettes connectées.
Dans ce projet, notre groupe de projet implémentera un système de traduction qui élargira l'utilisation de ces lunettes à un public plus vaste et permettra à deux individus ne pratiquant pas la même langue de pouvoir communiquer aisément.
Ce projet concentrera ses efforts sur l'implémentation d'un système de traduction plutôt que sur la reconnaissance vocale. Celle-ci nous sera fournie.

Il nous faut prendre en considération quelques contraintes d'usages final, et voir si nous pourrons les respecter :

- Traduction en temps réel d'un dialogue oral -> optimisation sur la rapidité
- Dialogue courant sans expertise particulière (champs sémantique généraliste)
- Prise en compte de la vitesse de lecture de chacun, la traduction doit être synthétique et conserver l'idée clé sans biais. (tout public et/ou design inclusif)

Il est souhaitable que le système puisse rapidement identifier si les phrases fournies sont exprimées dans une des langues connues par le système de traduction, et si c'est le cas, laquelle.
De plus, si le système de reconnaissance vocale n'est pas fiable, il est souhaitable de corriger la phrase en fonction des mots environnants ou des phrases préalablement entendues.
Lors de la traduction, nous prendrons en compte le contexte défini par la phrase précédente ainsi que par le contexte des phrases préalablement traduites.
Nous évaluerons la qualité de nos résultats en les comparant avec des systèmes performants tels que "Google translate" et "Deepl".
Enfin, si le temps, nos compétences et les datasets existants, le permettent, nous intégreront une langue originale, non proposée par ces systèmes, telle qu'une langue régionale ou de l'argot.

Le projet est enregistré sur Github

Compréhension et manipulation des données

Cadre

- **Jeux de données utilisés**

1. Small_vocab

Nous avons étudié le dataset **small_vocab**, proposés par Suzan Li, Chief Data Scientist chez Campaign Research à Toronto.
Celui-ci représente un corpus de phrases simples en anglais et sa traduction (approximative) en français.
Small_vocab contient 137 860 phrases en anglais et français :

- En anglais, ce dataset comprend 1 552 863 mots dont 199 mots uniques
- En français, il comprend 1 728 899 mots dont 330 mots uniques.

Référence:

- *Gitbub "NLP with Python" de Suzan Li*
- *Exploration 6 small_vocab.ipynb*
- *Données du projets*

Ces données sont en accès libre sur github

2. Vectors-Wiki

Nous avons aussi utilisé les datasets **Vectors-Wiki,** qui listent plusieurs millions de mots monolingues, vectorisés dans des espaces vectoriels alignés, issus de corpus parallèles de Wikimedia compilés par Facebook Research. Nous avons concentré nos efforts initiaux sur l'anglais et le français, afin de pouvoir exploiter le dataset **small_vocab.**

Le dataset **Vectors-Wiki** représente :

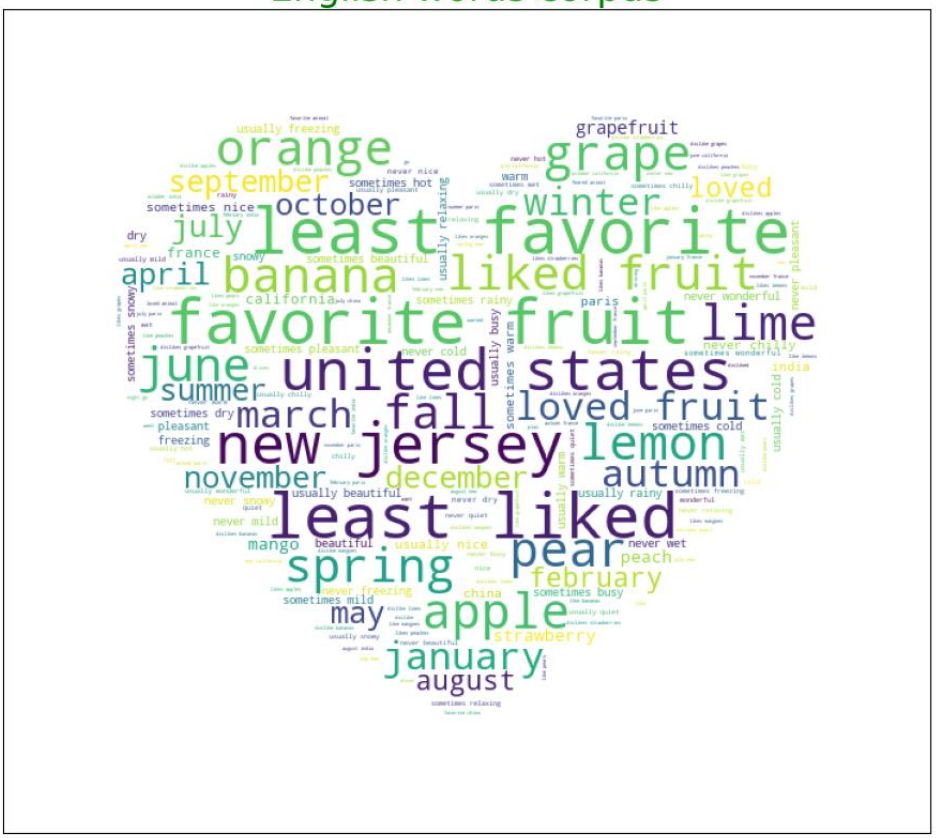
- En anglais, 2 519 370 mots uniques avec ses vecteurs de 300 dimensions.
- En français, 1 152 449 mots uniques.

Référence :

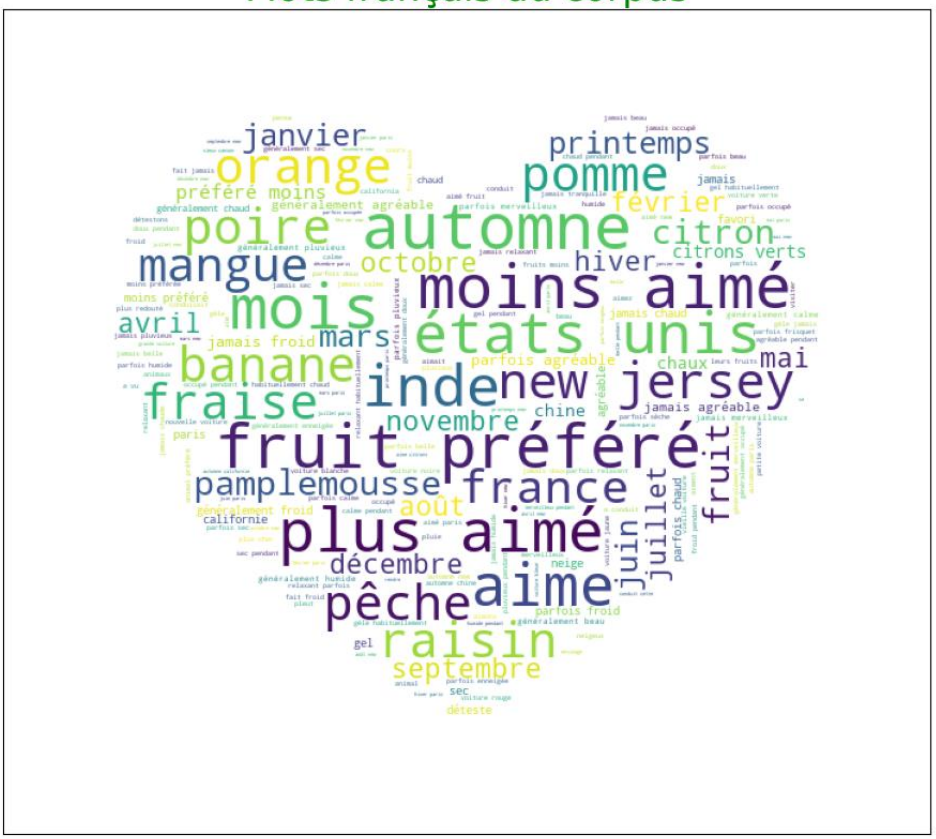
- *https://ai.facebook.com/blog/wikimatrix/*
- *https://opus.nlpl.eu/WikiMatrix.php*
- *Holger Schwenk, Vishrav Chaudhary, Shuo Sun, Hongyu Gong and Paco Guzman, WikiMatrix: Mining 135M Parallel Sentences in 1620 Language Pairs from Wikipedia, arXiv, July 11 2019.*

Ces données sont en accès libre sur github ou sur le site opus.nlpl.eu

English words corpus



Mots français du corpus



Pertinence

- **Variables pertinentes**

Les datasets sont composés de phrases et de mots, donc aucune variable ne nous est explicitement fournie.
Voici les premières phrases des corpus:

```
EN:  new jersey is sometimes quiet during autumn , and it is snowy in april .
FR:  new jersey est parfois calme pendant l' automne , et il est neigeux en avril .

EN:  the united states is usually chilly during july , and it is usually freezing in november .
FR:  les états-unis est généralement froid en juillet , et il gèle habituellement en novembre .

EN:  california is usually quiet during march , and it is usually hot in june .
FR:  california est généralement calme en mars , et il est généralement chaud en juin .

EN:  the united states is sometimes mild during june , and it is cold in september .
FR:  les états-unis est parfois légère en juin , et il fait froid en septembre .
```

Néanmoins, lors de notre pré-traitement, nous produirons de nombreuses variables intermédiaires nécessaires au processus de traduction :

- **Texte « propre »** : *txt_en, txt_fr.* Texte notamment sans ponctuation.

- **Token** : *txt_split_en, txt_split_fr.*
  Ces variable sont des tableaux numpy à 2 dimensions (dim 1 = phrase, dim 2 = 'mot' dans la phrase) :

[['new', 'jersey', 'is', 'sometimes', 'quiet', 'during', 'autumn', 'and', 'it', 'is', 'snowy', 'in', 'april'],
['the', 'united', 'states', 'is', 'usually', 'chilly', 'during', 'july', 'and', 'it', 'is', 'usually', 'freezing', ...], ...]

- **Bag Of Words (BOW)** : *df_count_word_en, df_count_word_fr.*
  Ces dataframes comprennent les phrases en ligne, et les mots en colonnes. Les valeurs sont le nombre d'occurrence du mot dans la phrase (comptage des tokens)

	a	am	and	animal	animals	apple	apples	april	are	aren	...	when	where	white	why	winter	wonderful	would	yellow	you	your
0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
1	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
3	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
4	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
137855	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
137856	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
137857	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0
137858	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
137859	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0

Ainsi, on constate que les mots 'and' et 'april' apparaissent 1 fois dans la 1ere phrase du corpus anglais.

- **Mots uniques (keywords)** : *corpus_en ; corpus_fr* (improprement appelé ainsi) :

```
['new', 'jersey', 'is', 'sometimes', 'quiet', 'during', 'autumn', 'and', 'it',
'snowy', 'in', 'april', 'the', 'united', 'states', 'usually', 'chilly', 'july',
'freezing', 'november', 'california', 'march', 'hot', 'june',…]
```

- **Mots transparents** (stop words)
  Ces mots sont les plus courants dans la langue et portent moins de signification que les autres. Ces mots sont enlevés des phrases lors du pré-processing
  Exemple anglais : {'a', 'about', 'above', 'after', 'again', 'against',…}

- **Longueur des phrases** (en mots) : *sent_len_en, sent_len_fr* ('sent' signifie « sentence ») :
  sent_len_en = [13, 15, 13, 14, 14, 12, 12,…]

  Ainsi la 1ere phrase anglaise comporte 13 mots, la 2eme 15, etc.

- **Modèle d'embedding** : *en_model, fr_model*
  Ces modèles nous permettront de convertir les mots en vecteur de 300 dimensions. Les espaces vectoriels dans les lesquels sont « plongés » ces mots, sont alignés. Ainsi des mots de même signification dans 2 langues différentes doivent avoir des vecteurs proches.

- **Variables cible**

Les variables cibles sont des **DataFrames** « **dictionnaire** », comprenant les mots du langage source en nom de colonne, et la traduction de ces mots en 1ere ligne.
Voici la liste des dictionnaires produits avec les différents algorithmes :

- *dict_FR_EN* et *dict_EN_FR* créés avec un BOW et l'algorithme K-Means
- *knn_dict_FR_EN* et *knn_dict_EN_FR* créés avec un BOW et l'algorithme KNN avec k=1
- *rf_dict_FR_EN* et *rf_dict_EN_FR* créés avec un BOW et l'algorithme Random Forest
- *we_dict_FR_EN* et *we_dict_EN_FR* créés avec Vectors-Woki et la méthode most_similar des modèles de Word Embedding

- **Particularités**

Les corpus de phrases multilingues sont faciles d'accès et nombreux. **Small_vocab** constitue un jeu de données d'une simplicité extrême. Néanmoins ils nécessitent un pré-processing important afin de pouvoir être compris par les algorithmes de traitement.

Pre-processing et feature engineering

- **Nettoyage**
  La première étape est de nettoyer le texte afin de travailler un texte «propre».
  - Mettre le texte en minuscule
  - Enlever les urls, la ponctuation ("." , "," , ";" , ":" , "?" , "-"), les chiffres et les espaces superflus.
  - Corriger les fautes d'orthographe (étape non exécutée car longue et inutile sur ce corpus)
  - Lemmatiser, c'est-à-dire remplacer les mots par leur forme neutre canonique que l'on trouve dans un dictionnaire. Ainsi en anglais, le lemme du mot 'is' est 'be', et celui du mot 'bananas' est 'banana'.
  - Enlever les stop words (Cf. ci-dessus)
  - Il aurait fallu aussi identifier les noms propres (ce que nous n'avons pas fait)

  Note : Dans la suite du projet, nous n'utiliserons pas le texte lemmatisé, ni le texte sans stop words, car nous allons effectuer une traduction mot à mot, et nous avons besoin de tous les mots sous leur forme originelle.

- **Tokenisation**
  Le texte est segmenté en mots que nous appellerons "tokens". Nous n'avons pas eu besoin de tokeniser en N-Gram, c'est-à-dire une combinaison de N mots.

- **Parts of Speech (POS) tagging)**
  Ce processus traite une séquence de mots et attache un marqueur de parties du discours à chaque mot. La bibliothèque 'nltks poc_tag' est utilisée pour l'étiquetage POS. Voici quelques exemples de balises POS : VBZ -> Verbe, NN-> Nom, PRP -> Préposition, IN -> Interjection.
  Bien que nous ayons réalisé ce processus pour l'anglais, il ne nous a pas été utile à ce stade.

- **Text To Numeric (Term To Digit)**
  À ce stade du projet et avec les algorithmes que nous avons utilisés, nous n'avons pas eu besoin de convertir les mots en nombre.

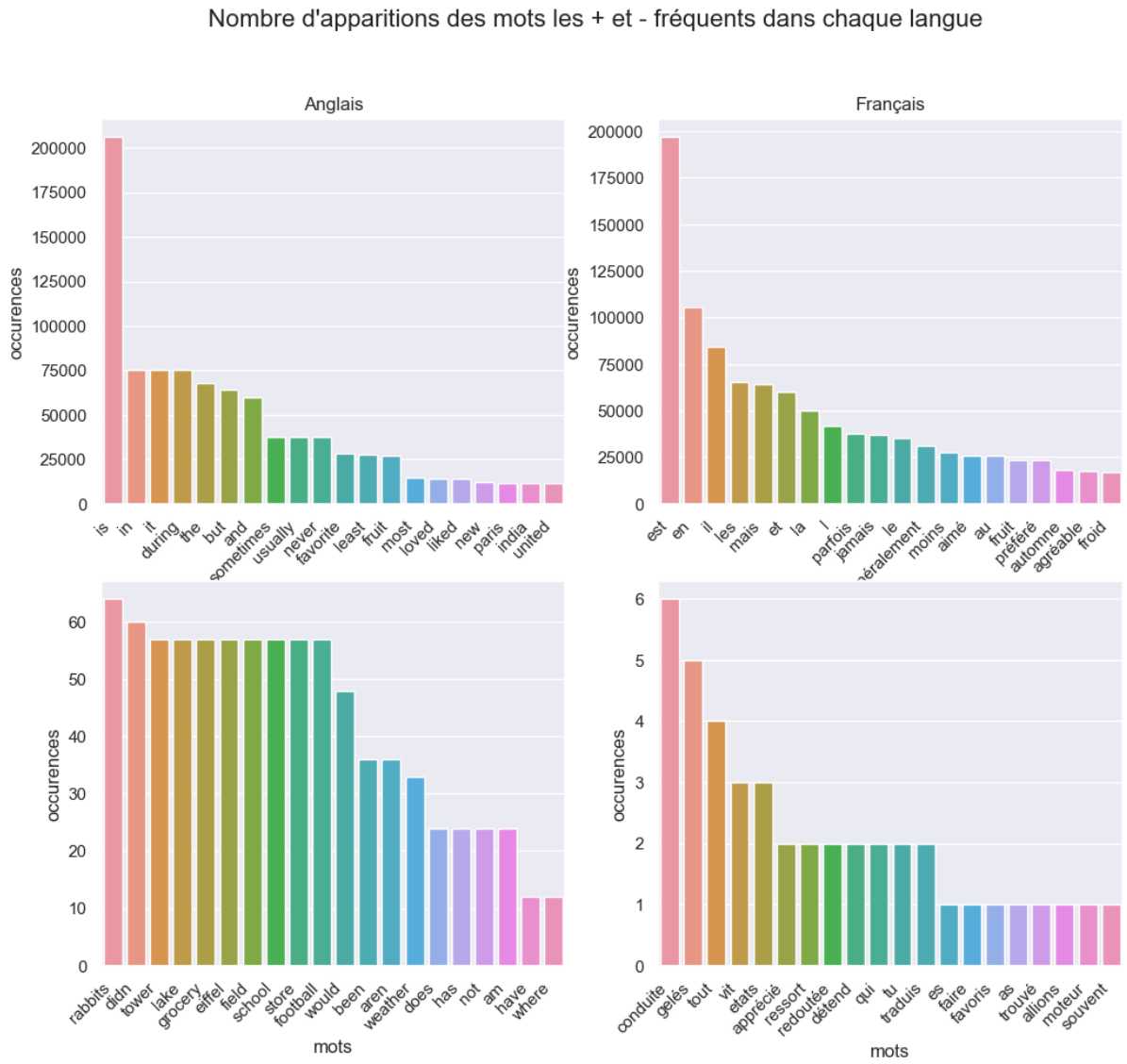
- **Création d'un Bag Of Words (BOW) ou Vectorization TF-IDF**
  Une fois le texte nettoyé et tokenisé, nous pouvons compter les mots dans chaque phrase et ainsi créer les variables BOW (Cf. ci-dessus), grâce à la méthode CountVectorizer de Sklearn. En fait, nous avons utilisé une forme particulière du BOW dans laquelle on enregistre que la présence du mot dans la phrase (mot présent = 1, absent = 0).

  Nous avons aussi réalisé une vectorisation TF-IDF, grâce à la méthode TfidfVectorizer de Sklearn. TF-IDF signifie "Term Frequency-Inverse Document Frequency". Il mesure l'importance d'un terme (mot) par rapport à un document (phrase) dans un corpus. Chaque terme d'un document se voit attribuer un poids après avoir multiplié sa fréquence de terme (tf) et sa fréquence inverse de document (idf). Lors de son utilisation, cette vectorisation n'a pas donné de meilleurs résultats que le BOW. Nous l'avons donc abandonné.

Visualisations et Statistiques

- **Relation entre les mots**

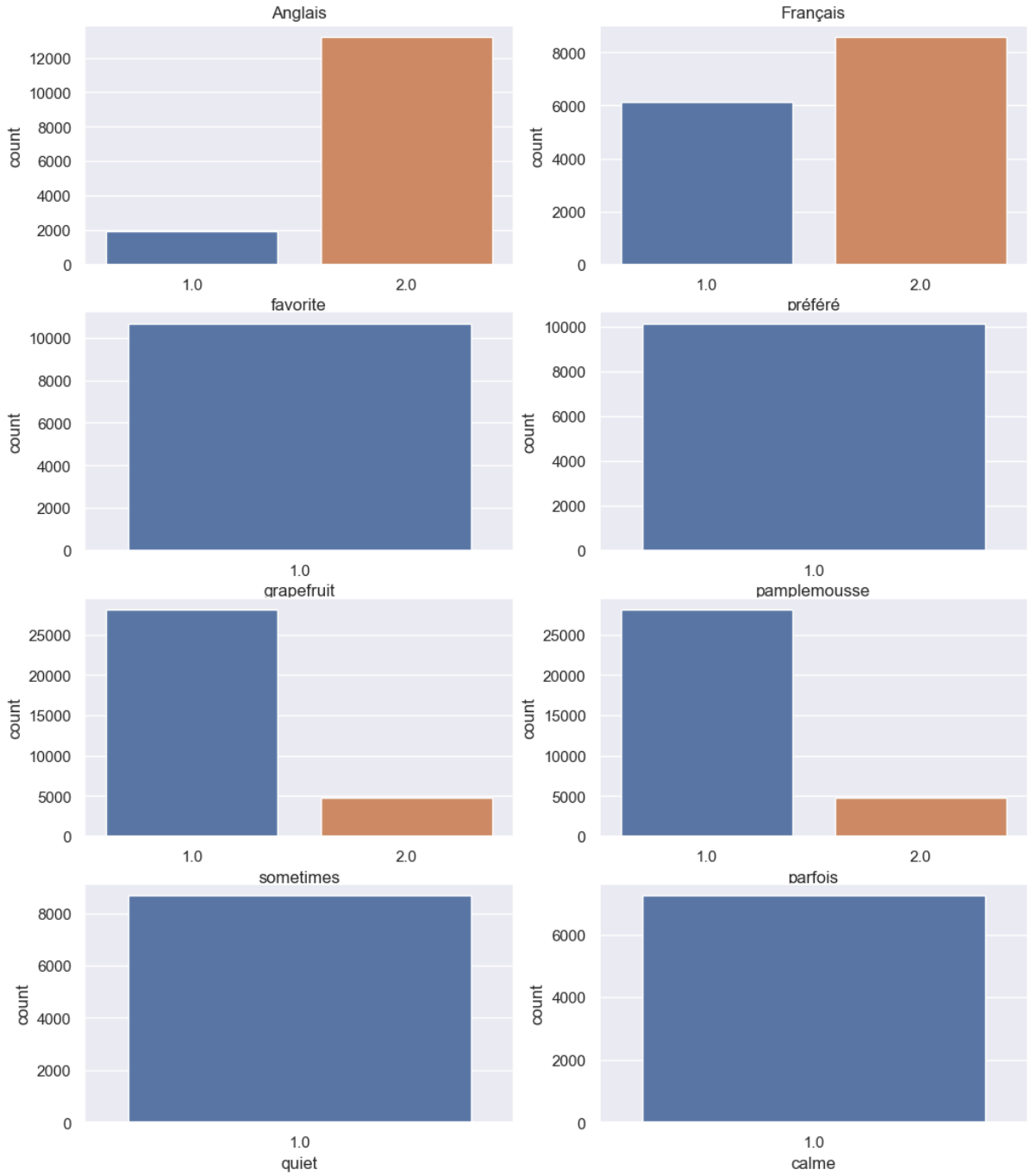
  Si nous regardons le nombre d'apparitions des mots dans le corpus, nous voyons une certaine analogie entre les corpus anglais et français, notamment pour les mots les plus fréquents (graphes ci-dessous de la 1er ligne).



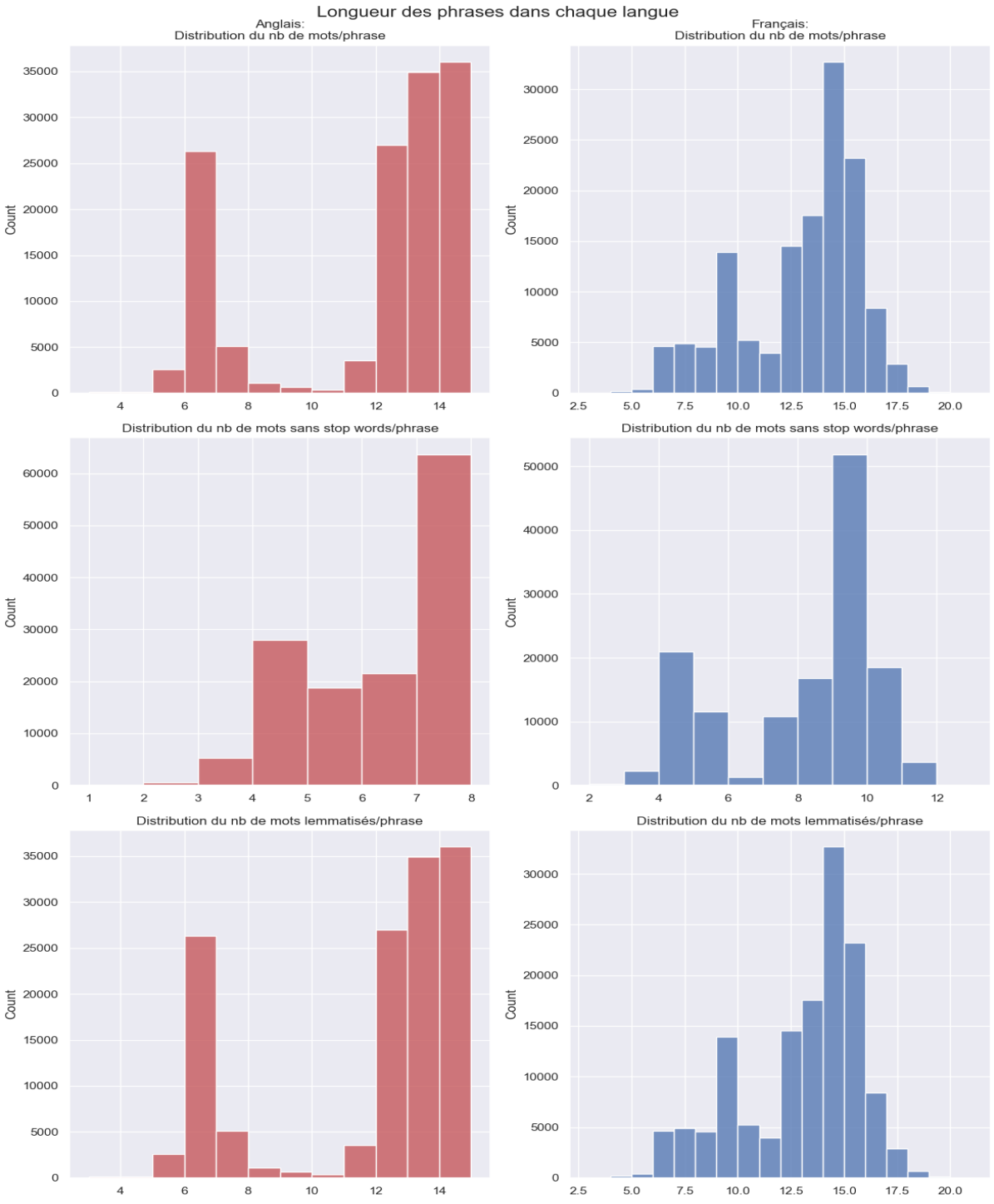
Ainsi, la fréquence d'apparition pourrait permettre d'associer un mot avec sa traduction.

Voici la distribution de fréquence d'apparition dans une phrase de certains mots sélectionnés : 'favorite' et 'préféré' (respectivement en anglais et français), 'grapefruit' et 'pamplemousse', 'sometimes' et 'parfois', 'quiet' et 'calme'. On constate une similitude dans les distributions anglaises et françaises.

Nombre de phrases (y) où l'on trouve certains mots avec une certaine occurence (x)



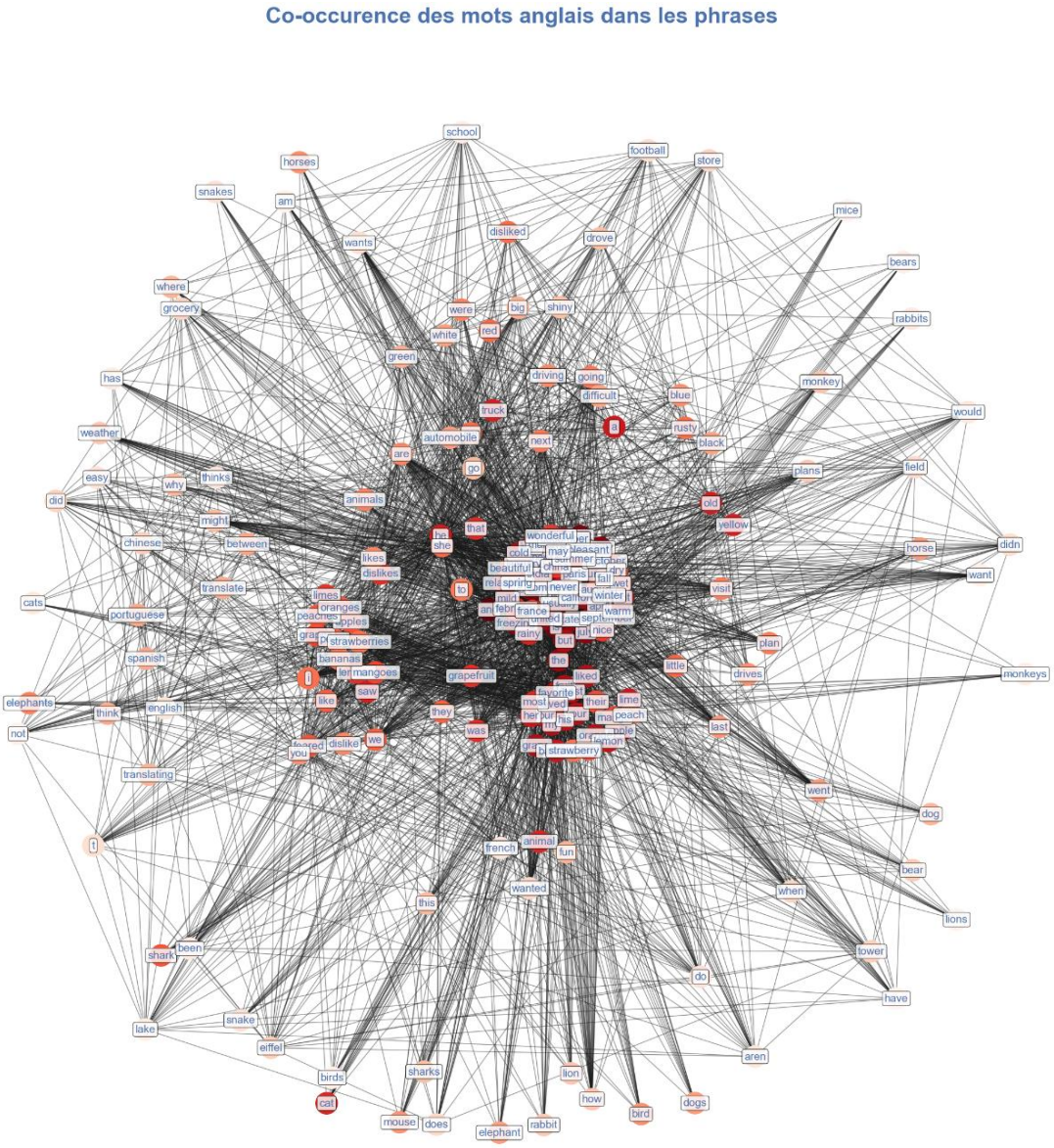
Analyse des longueurs de phrase en mot



On constate une certaine similitude entre les 2 langues, même si les phrases françaises ont plus de mots que les phrases anglaises. Cela nous conforte dans l'idée d'utiliser le BOW pour associer les mots des 2 langues

- **Analyse de co-occurrences de mots dans une phrase**

  Le graphe est difficilement lisible. On constate, bien sûr, que chaque mot apparaît avec beaucoup d'autres mots du corpus.



Exemple sur le mot anglais « fruit » :
Ce mot apparaît dans 23 500 phrases sur les 137 860 du corpus. Voici les premières phrases dans lesquelles ce mot apparaît.

text_en:	your least liked fruit is the grape
Tokens & Tags:	least | like | fruit | grape
text_en:	his favorite fruit is the orange
Tokens & Tags:	favorite | fruit | orange
text_en:	our least liked fruit is the lemon
Tokens & Tags:	least | like | fruit | lemon
text_en:	the lime is her least liked fruit
Tokens & Tags:	lime | least | like | fruit
text_en:	he dislikes grapefruit
Tokens & Tags:	dislike | grapefruit
text_en:	her least liked fruit is the lemon
Tokens & Tags:	least | like | fruit | lemon
text_en:	their favorite fruit is the mango
Tokens & Tags:	favorite | fruit | mango
text_en:	the grapefruit is my most loved fruit
Tokens & Tags:	grapefruit | love | fruit
text_en:	the orange is her least liked fruit
Tokens & Tags:	orange | least | like | fruit
text_en:	the lemon is my most loved fruit
Tokens & Tags:	lemon | love | fruit
text_en:	the apple is our least favorite fruit
Tokens & Tags:	apple | least | favorite | fruit

- **Analyse du Word Embedding des mots des 2 corpus**

Nous avons récupéré les vecteurs de chaque mot des Corpus dans « Vectors-Wiki ». Si l'on affiche une « Analyse en Composante Principale » des mots les plus fréquents des 2 corpus, on constate un certaine « proximité » entre les mots anglais et leur traduction.

Modalités d'évaluation :

- Nous avons vu que l'utilisation d'un BOW pour la traduction mot à mot était prometteuse.
- Cependant, il sera intéressant d'étudier l'utilisation du Word Embedding, qui présente l'avantage d'être indépendant des corpus de mots.

Rendu 2 : rapport de modélisation

Étapes de réalisation du projet

Classification du problème

Dans une première approche naïve, nous allons implémenter un système de traduction mot à mot (phase 1 et 2).
Pour cela nous allons créer automatiquement (sans intervention manuelle), 2 dictionnaires (FR->EN, EN->FR), en utilisant les textes et leur traduction pour la phase d'apprentissage.

1. Lors d'une première phase, nous allons associer un mot d'une langue avec un mot d'une autre langue (catégorie). Cela peut être réalisé par des **algorithmes de classification** (supervisés) **ou de clustering** (non supervisés).

2. Dans une deuxième phase, nous allons « plonger » nos mots dans une **vectorisation « FasText »**, afin de trouver la traduction d'un mot par similitude de vecteur.

3. Enfin, dans une troisième phase, nous utiliserons, des algorithmes de **Deep Learning** de réseaux neuronaux pour obtenir des traductions de meilleure qualité qui s'affranchisse d'une relation « **one** (word) **to one** (word) ».

Choix du modèle et optimisation pour les phases 1 et 2

Afin de pouvoir mesurer la qualité de la traduction des algorithmes de clustering et classification, nous avons constitué un dictionnaire de référence « idéale » à la main, afin de connaître la meilleure traduction de chaque mot dans le corpus de l'autre langue (dict_FR_EN_ref, dict_EN_FR_ref). Cela permettra de définir une fonction approximative de **précision** la traduction (% de mots correctement traduits)

1. Clustering et classification

a. Clustering non supervisé, **K-Means**

L'idée est de créer une classe 'mot' et un label pour chaque mot d'une langue. Chaque mot représentera donc un cluster. Les caractéristiques (« features ») sont les vecteurs du BOW, c'est-à-dire des vecteurs ayant une dimension = 137 860, contenant des 0 ou des 1 en fonction de la présence du mot dans une phrase. Ce vecteur constitue une empreinte unique du mot.
Il n'est pas nécessaire d'entraîner longtemps le modèle puisque le centroïde du cluster est le vecteur du mot (1 itération suffit)

Exemple : Ensemble d'entraînement pour un dictionnaire FR ->EN

label	0	1	2	3	4	5	6	7	8	9	...	137850	137851	137852	137853	137854	137855	137856	137857	137858	137859
	X_train																		N° de phrase		
a	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
am	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
and	1.0	1.0	1.0	1.0	0.0	0.0	0.0	1.0	0.0	1.0	...	0.0	1.0	0.0	0.0	0.0	1.0	1.0	0.0	0.0	0.0
animal	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
animals	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
wonderful	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
would	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
yellow	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
you	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
your	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0

Présence dans ces phrases

Enfin, considérons les mots de l'autre langue, avec ses vecteurs BOW) comme étant l'ensemble de test.
Il suffit de prédire le label des mots de la 2eme langue dans la première langue.

Voici le résultat :

```
Dictionnaire Anglais -> Français:
169  mots corrects /  199
Précision du dictionnaire = 84.92%
```

Anglais	a	am	and	animal	animals	apple	apples	april	are	aren	...	when	where	white	why	winter	wonderful	would	yellow	you	your
Francais	une	vais	et	animal	animaux	pomme	pommes	avril	sont	allions	...	quand	où	blanche	pourquoi	hiver	merveilleux	voudrait	jaune	vous	votre

```
Dictionnaire Français -> Anglais:
152  mots corrects /  330
Précision du dictionnaire = 46.06%
```

Francais	a	agréable	aimait	aime	aiment	aimeraient	aimez	aimons	aimé	aimée	...	à	école	éléphant	éléphants	épicerie	étaient	était	états	été	êtes
Anglais	drove	pleasant	disliked	likes	they	have	you	have	loved	have	...	fall	school	elephant	elephants	grocery	were	was	states	summer	have

b. Classification, **K-NN** et **Random Forest**

Là aussi nous allons créer une classe 'mot', puis prédire le label d'un mot de l'autre langue. Dans le cas de l'algorithme du K-NN, on utilise k=1, puisque l'on veut associer un mot cible à un mot source. L'avantage de K-NN sur le K-Means est que l'on peut utiliser une métrique différente de la métrique euclidienne imposée par le K-Means. Dans notre projet, nous avons utilisé la métrique 'minkowski' pour le dictionnaire EN ->FR et la métrique ' cosine' pour le dictionnaire FR->EN (métriques qui donnent la meilleur précision)

Voici les résultats :

K-NN

```
Dictionnaire Anglais -> Français:
171  mots corrects /  199
Précision du dictionnaire = 85.93%
```

Anglais	a	am	and	animal	animals	apple	apples	april	are	aren	...	when	where	white	why	winter	wonderful	would	yellow	you	your
Francais	une	vais	et	animal	animaux	pomme	pommes	avril	sont	allions	...	quand	où	blanche	pourquoi	hiver	merveilleux	voudrait	jaune	vous	votre

```
Dictionnaire Français -> Anglais:
240  mots corrects /  330
Précision du dictionnaire = 72.73%
```

Francais	a	agréable	aimait	aime	aiment	aimeraient	aimez	aimons	aimé	aimée	...	à	école	éléphant	éléphants	épicerie	étaient	était	états	été	êtes
Anglais	drove	pleasant	disliked	likes	they	would	you	we	loved	loved	...	fall	school	elephant	elephants	grocery	were	was	states	summer	did

Random Forest

```
Dictionnaire Anglais -> Français:
163  mots corrects /  199
Précision du dictionnaire = 81.91%
```

Anglais	a	am	and	animal	animals	apple	apples	april	are	aren	...	when	where	white	why	winter	wonderful	would	yellow	you	your
Francais	une	vais	et	animal	animaux	pomme	pommes	avril	sont	allez	...	quand	où	blanc	pourquoi	hiver	merveilleux	voudrait	jaune	vous	votre

```
Dictionnaire Anglais -> Français:
173  mots corrects /  330
Précision du dictionnaire = 52.42%
```

Francais	a	agréable	aimait	aime	aiment	aimeraient	aimez	aimons	aimé	aimée	...	à	école	éléphant	éléphants	épicerie	étaient	était	états	été	êtes
Anglais	saw	nice	disliked	likes	they	where	you	we	loved	where	...	fall	school	elephant	elephants	store	were	was	states	summer	where

.

2. Vectorisation « FastText »

Nous avons utilisé **Vectors-Wiki** pour obtenir la vectorisation de chaque mot du corpus (dimension = 300)
Ainsi il est possible de comparer la similitude des mots en utilisant la méthode 'most_similar' (métrique 'cosine').

Exemple des mots les plus proches de 'hiver', avec leur 'similarity scores' :

```
fr_model.most_similar("hiver")
```

```
[('automne', 0.6660692095756531),
 ('printemps', 0.6183713674545288),
 ('neigeux', 0.5664983987808228),
 ('neige', 0.5257566571235657),
 ('enneigée', 0.5075902938842773),
 ('pluvieux', 0.503300130367279),
 ('enneigé', 0.4926919639110565),
 ('froid', 0.47274085879325867),
 ('gelé', 0.4528149366378784),
 ('gèle', 0.45185184478759766)]
```

Le « word2vec embedding » capture efficacement les propriétés sémantiques et arithmétiques d'un mot.
Ainsi, il est possible de faire de l'arithmétique avec les vecteurs de mot, et de faire directement la traduction du résultat.

Exemple : 'King' + 'Man' – 'Woman' = en français à 'Reine'

```
# traduction de : 'king' + 'man' - 'woman' = 'reine'
vect1 = en_model.get_vector("king")
vect2 = en_model.get_vector("man")
vect3 = en_model.get_vector("woman")
print("Traduction en français de ('king' - 'man' + 'woman') = ",fr_model.most_similar(vect1-vect2+vect3)[0][0])
```

```
Traduction en français de ('king' - 'man' + 'woman') =  reine
```

En utilisant une liste de plusieurs milliers de mots les plus courants, nous avons créé un sous dictionnaire dans chaque langue, sous-ensemble de Vectors-Wiki, **mini.wiki.en.align.vec** et **mini.wiki.fr.align.vec**, dictionnaires qui permettent d'accélérer les calculs. En effet, trouver les mots les plus similaires « cross-langues » parmi plusieurs millions de mots prend énormément de temps.

Finalement, voici les dictionnaires calculés avec cette méthode :

Dictionnaire Anglais -> Français:

Anglais	a	am	and	animal	animals	apple	apples	april	are	aren	...	when	where	white	why	winter	wonderful	would	yellow	yo
Francais	une	je	et	animaux	animaux	pomme	pommes	février	sont	sont	...	lorsque	où	blanc	pourquoi	hiver	merveilleux	pourrait	jaune	m

1 rows × 199 columns

Dictionnaire Français -> Anglais :

Francais	a	agréable	aimait	aime	aiment	aimeraient	aimez	aimons	aimé	aimée	...	à	école	éléphant	éléphants	épicerie	étaient	était	états
Anglais	has	pleasant	loved	love	enjoy	want	dare	come	loved	loved	...	to	school	elephant	elephants	grocery	were	was	states

1 rows × 330 columns

Interprétation des résultats

- Voici quelques exemples de traduction avec les différents algorithmes :

1. EN -> FR

Traduction à l'aide du dictionnaires de reference:

Anglais	**paris**	**is**	**never**	**freezing**	**during**	**november**	**but**	**it**	**is**	**wonderful**	**in**	**october**
Francais	paris	est	jamais	gel	en	novembre	mais	il	est	merveilleux	en	octobre

Anglais	**the**	**banana**	**is**	**their**	**favorite**	**fruit**	**but**	**the**	**grapefruit**	**is**	**your**	**favorite**
Francais	le	banane	est	leur	préféré	fruit	mais	le	pamplemousse	est	votre	préféré

Anglais	**that**	**cat**	**was**	**my**	**most**	**loved**	**animal**
Francais	cette	chat	était	mon	plus	cher	animal

Traduction à l'aide du dictionnaires KMeans calculés:

Anglais	**paris**	**is**	**never**	**freezing**	**during**	**november**	**but**	**it**	**is**	**wonderful**	**in**	**october**
Francais	paris	est	jamais	gel	en	novembre	mais	en	est	merveilleux	en	octobre

Anglais	**the**	**banana**	**is**	**their**	**favorite**	**fruit**	**but**	**the**	**grapefruit**	**is**	**your**	**favorite**
Francais	fruit	banane	est	leur	préféré	fruit	mais	fruit	pamplemousse	est	votre	préféré

Anglais	**that**	**cat**	**was**	**my**	**most**	**loved**	**animal**
Francais	cette	chat	était	mon	plus	plus	animal

Traduction à l'aide du dictionnaires KNN calculés:

Anglais	**paris**	**is**	**never**	**freezing**	**during**	**november**	**but**	**it**	**is**	**wonderful**	**in**	**october**
Francais	paris	est	jamais	gel	en	novembre	mais	en	est	merveilleux	en	octobre

Anglais	**the**	**banana**	**is**	**their**	**favorite**	**fruit**	**but**	**the**	**grapefruit**	**is**	**your**	**favorite**
Francais	fruit	banane	est	leur	préféré	fruit	mais	fruit	pamplemousse	est	votre	préféré

Anglais	**that**	**cat**	**was**	**my**	**most**	**loved**	**animal**
Francais	cette	chat	était	mon	plus	plus	animal

Traduction à l'aide du dictionnaires RF calculés:

Anglais	**paris**	**is**	**never**	**freezing**	**during**	**november**	**but**	**it**	**is**	**wonderful**	**in**	**october**
Francais	paris	est	jamais	gel	en	novembre	mais	en	est	merveilleux	en	octobre

Anglais	**the**	**banana**	**is**	**their**	**favorite**	**fruit**	**but**	**the**	**grapefruit**	**is**	**your**	**favorite**
Francais	fruit	banane	est	leur	préféré	fruit	mais	fruit	pamplemousse	est	votre	préféré

Anglais	**that**	**cat**	**was**	**my**	**most**	**loved**	**animal**
Francais	cette	chat	était	mon	plus	plus	animal

Traduction à l'aide du dictionnaires Word Embedding FastText :

Anglais	**paris**	**is**	**never**	**freezing**	**during**	**november**	**but**	**it**	**is**	**wonderful**	**in**	**october**
Francais	paris	est	jamais	froid	durant	février	mais	elle	est	merveilleux	dans	février

Anglais	**the**	**banana**	**is**	**their**	**favorite**	**fruit**	**but**	**the**	**grapefruit**	**is**	**your**	**favorite**
Francais	la	bananes	est	leurs	préférés	fruits	mais	la	pamplemousse	est	votre	préférés

Anglais	**that**	**cat**	**was**	**my**	**most**	**loved**	**animal**
Francais	que	chat	était	mon	plus	aimait	animaux

2. FR -> EN

Traduction à l'aide du dictionnaire de reference:

Francais	**paris**	**est**	**jamais**	**le**	**gel**	**en**	**novembre**	**mais**	**il**	**est**	**merveilleux**	**en**	**octobre**
Anglais	paris	is	never	the	freezing	in	november	but	it	is	wonderful	in	october

Francais	**la**	**banane**	**est**	**leur**	**fruit**	**préféré**	**mais**	**le**	**pamplemousse**	**est**	**votre**	**favori**	
Anglais	the	banana	is	their	fruit	favorite	but	the	grapefruit	is	your	favorite	

Francais	**ce**	**chat**	**était**	**mon**	**animal**	**animal**	**le**	**plus**	**aimé**
Anglais	this	cat	was	my	animal	animal	the	most	loved

Traduction à l'aide du dictionnaire Kmeans calculés:

Francais	**paris**	**est**	**jamais**	**le**	**gel**	**en**	**novembre**	**mais**	**il**	**est**	**merveilleux**	**en**	**octobre**
Anglais	paris	is	never	grapefruit	freezing	in	november	but	it	is	wonderful	in	october

Francais	**la**	**banane**	**est**	**leur**	**fruit**	**préféré**	**mais**	**le**	**pamplemousse**	**est**	**votre**	**favori**	
Anglais	fruit	banana	is	their	fruit	favorite	but	grapefruit	grapefruit	is	football	have	

Francais	**ce**	**chat**	**était**	**mon**	**animal**	**le**	**plus**	**aimé**	
Anglais	this	cat	was	my	animal	grapefruit	most	loved	

Traduction à l'aide du dictionnaire KNN calculés:

Francais	**paris**	**est**	**jamais**	**le**	**gel**	**en**	**novembre**	**mais**	**il**	**est**	**merveilleux**	**en**	**octobre**
Anglais	paris	is	never	fruit	freezing	in	november	but	it	is	wonderful	in	october

Francais	**la**	**banane**	**est**	**leur**	**fruit**	**préféré**	**mais**	**le**	**pamplemousse**	**est**	**votre**	**favori**	
Anglais	fruit	banana	is	their	fruit	favorite	but	fruit	grapefruit	is	your	favorite	

Francais	**ce**	**chat**	**était**	**mon**	**animal**	**le**	**plus**	**aimé**	
Anglais	this	cat	was	my	animal	fruit	most	loved	

Traduction à l'aide du dictionnaire RF calculés:

Francais	**paris**	**est**	**jamais**	**le**	**gel**	**en**	**novembre**	**mais**	**il**	**est**	**merveilleux**	**en**	**octobre**
Anglais	paris	is	never	most	freezing	in	november	but	it	is	wonderful	in	october

Francais	**la**	**banane**	**est**	**leur**	**fruit**	**préféré**	**mais**	**le**	**pamplemousse**	**est**	**votre**	**favori**	
Anglais	fruit	banana	is	their	fruit	favorite	but	most	grapefruit	is	your	where	

Francais	**ce**	**chat**	**était**	**mon**	**animal**	**le**	**plus**	**aimé**	
Anglais	this	cat	was	my	animal	most	most	loved	

Traduction à l'aide du dictionnaires Word Embedding FastText :

Francais	**paris**	**est**	**jamais**	**le**	**gel**	**en**	**novembre**	**mais**	**il**	**est**	**merveilleux**	**en**	**octobre**
Anglais	paris	is	never	the	freeze	in	june	but	he	is	wonderful	in	june

Francais	**la**	**banane**	**est**	**leur**	**fruit**	**préféré**	**mais**	**le**	**pamplemousse**	**est**	**votre**	**favori**	
Anglais	the	banana	is	their	fruit	favorite	but	the	grapefruit	is	your	favorite	

Francais	**ce**	**chat**	**était**	**mon**	**animal**	**le**	**plus**	**aimé**	
Anglais	that	cat	was	my	animal	the	less	loved	

Conclusion (temporaire) :

- Les résultats de création de dictionnaire sont relativement bons, notamment dans le sens EN->FR.

- Les méthodes de clustering et de classification ne sont malheureusement pas généralisables à tout type de corpus. Nous avons vu que **small_vocab** comprend une grande redondance de mots, condition 'sine qua non' du fonctionnement des algorithmes de classification et clustering sur le BOW.

- La méthode de Word Embedding avec FastText semble plus prometteuse car elle n'a pas besoin de corpus symétrique, ni de redondance de mots. Néanmoins la traduction mot à mot montre ses limites, et nous n'avons pas la garantie sur la qualité de la traduction, comme le montre les exemples ci-dessus, avec la confusion des mois 'octobre', 'novembre', 'juin', 'février'.

- L'implémentation mot à mot, donne donc une piètre traduction. De plus, elle ne permet pas de prendre en compte les particularités du langage. Par exemple :
    - Le genre des articles en français n'apparaît pas en anglais
    - La disparition des articles dans une langue et pas dans l'autres (« des » en FR)
    - Les expressions de longueur inégale. Ainsi "is going to" en anglais peut se traduire par "va" en français.

- A ce stade du projet, il est souhaitable d'aborder la phase 3 avec le **Deep Learning**.

Bibliographie

1. ***Tutorials***
   a. *Kaggle*
      - Comprehensive NLP Tutorial-1:ML Perspective | Kaggle
      - ComprehensiveNLP Tutorial-2:DL Prespective | Kaggle
      - ComprehensiveNLP Tutorial-2:DL Prespective | Kaggle

   b. *Ekino*
      - Introduction au NLP (Partie I) - Ekino FR
      - Introduction au NLP (Partie I) - Ekino FR

   c. *TensorFlow*
      - Natural Language Processing (NLP) Zero to Hero - YouTube
      - Découvrez l'univers du machine learning

   d. *Hugging Face*
      - Introduction - Hugging Face NLP Course

2. ***DataScientest***
   1. Natural Language Processing (NLP) : Définition et principes
   2. Word2vec : NLP & Word Embedding - DataScientest
   3. NLP- Word translation - Formation Data Science | DataScientest.com
   4. NLP Twitter - Analyse de Sentiment - DataScientest
   5. Réseau de neurones : définition et fonctionnement
   6. Word Embedding et Systèmes de traduction - Script Video - Google Docs
      -

3. ***Pré-processing***
   7. Comprehensive NLP Tutorial-1:ML Perspective | Kaggle
   8. Natural Language Processing: Text Data Vectorization | by Paritosh Pantola | Medium
   9. 10+ Examples for Using CountVectorizer - Kavita Ganesan, PhD

4. ***Neural Machine Translation***
   10. Neural Machine Translation. Machine Translation using Recurrent… | by Quinn Lanners | Towards Data Science
   11. Using RNNs for Machine Translation | by Aryan Misra | Towards Data Science

5. ***Librairies***
   12. fastText
   13. John Snow Labs - State of the Art NLP in Python

Fichier de code

- *Exploration 6 small_vocab.ipynb*