ULB UNIVERSITÉ LIBRE DE BRUXELLES

STIC-B545 - Traitement automatique de corpus

Etudiant

NGOUFACK NANFAH Angelot

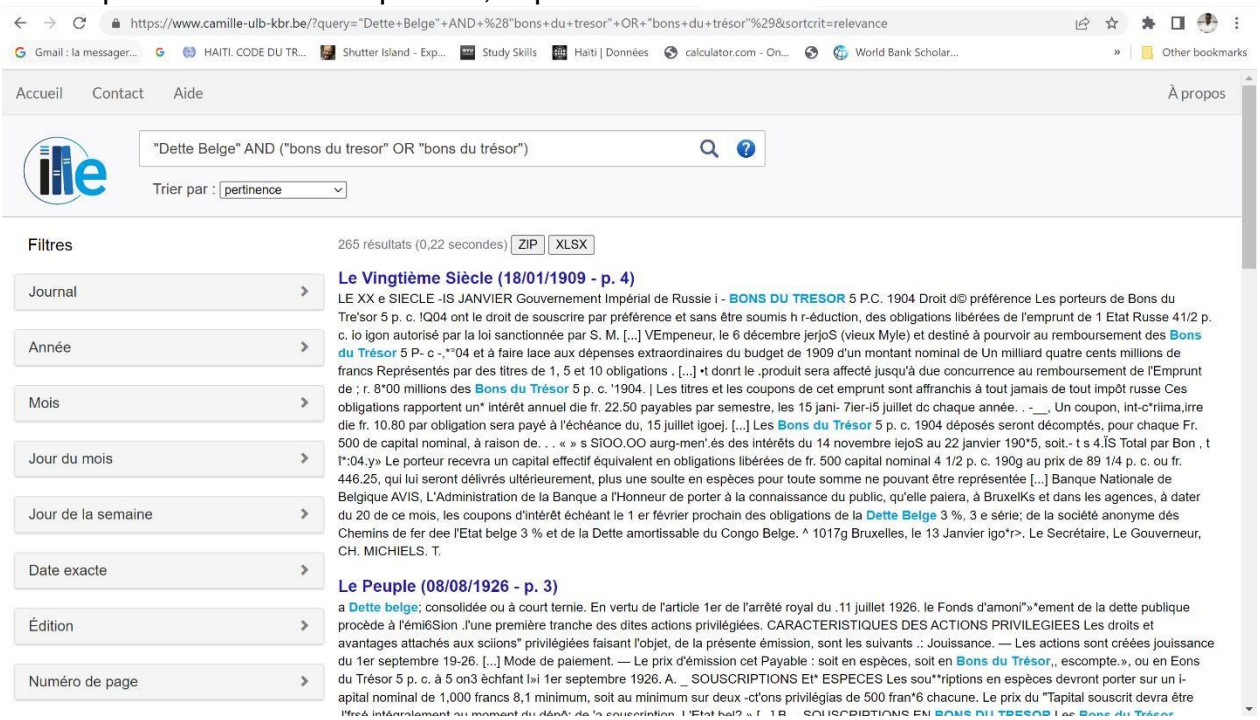
Sur la base des éléments méthodologiques et des enseignements techniques présentés lors du cours théorique, il est demande dans le cadre de ce TP :
— démobiliser les connaissances et compétences acquises tout au long du cours
— d'apporter un regard critique sur le traitement automatique de corpus en général

Les étapes à mettre en œuvre sont les suivantes :

1. Thématique très ciblée qui vous intéresse.

La dette belge, ses dépositaires, son évolution à travers le temps. Une vue globale des bons du trésor émis, les projets financés.

2. Requête suffisamment précise, exportez les résultats en ZIP et/ou XLSX.



De ce grand corpus du projet Camille qui regroupe les journaux des quotidiens de la presse belge à partir de 1831, ont été scannés puis ocrisés les éditions dans le but de faciliter la recherche sur l'histoire du journalisme et d'avoir aussi une plateforme numérique d'archives dédiée à ce sujet. Créé en 2020, CAMille (Centre d'archives sur les médias et l'information, ULB-KBR) entend offert cette plateforme aux études et recherches à des journalistes mais aussi à d'autres fins, dans les domaines variés tels les recherches en humanités numériques qui s'avère être un domaine scientifique en pleine progression.
Le présent rapport fait état d'une question de recherche et des outils techniques et méthodes de traitement automatique de corpus en vue d'obtenir de l'information, faire de la recherche sur de grand corpus d'archives papiers qui serait presqu'impossible d'explorer à la main.

Le sous corpus est construit à partir des documents extraits, sur base de la requête formulée en langage naturel avec des opérateurs booléens, accessoire de la plateforme

La formulation de la requête était au départ pour étudier la dette belge des états tiers ou quels états tiers possédaient la dette (titre) de l'état belge à l'époque. Une requête plutôt difficile à trouver.
Car le corpus n'est pas traité de manière à répondre à des requête spécifiques dans le texte. Il n'y a pas de méta donnée.
Donc comme indiqué dans la requête, la recherche s'effectue autour de la **dette belge via les bons du trésor émis**. Et en ce qui a trait aux méthodes, ce sont en principes des outils et techniques en Tac traitement automatique de corpus visant à faciliter l'extraction de l'information de manière intelligible.

D'abord il faut passer par l'étape crucial qui est la collecte de données : Les journaux de 1831 ont été éditées sous forme dure ou papier, donc la numérisation de tonnes de papiers et l'océrisation de ces derniers fut nécessaire à la constitution du corpus textuel. Déjà j'attire votre attention sur les difficultés techniques pour l'exploitation a la base, même pour parler de la qualité des extrants numérisés et ocerisés. Ce dernier désignant *l'Optical Character Recognition* permet la recherche full-text dans un document numérisé.

Après vient un autre étape qui n'est pas la moindre, il s'agit du nettoyage et transformation des données, ceci a été réalisé avec des programmes tel : Extract, transform, load ETL, avec OpenRefine et Python. A cet étape aussi s'ajoute la détection automatique d'anomalies dans le corpus comme une dernière vérification des documents formant le corpus.

Puis vient l'enrichissement à l'aide de techniques de traitement automatique des langues, il s'agit entre autres de méthode (extraction de termes, mots-clés et concepts, analyse de sentiment, résumé automatique, topic modeling) avec Python (NLTK, Gensim…). Pour extraire sous certaines formes de l'information et explorer le corpus.

Et aussi l'apprentissage automatique supervisé et non supervisé : régression linéaire pour la classification, clustering avec K-Means, exercices pratiques avec scikit-learn, deep learning et plongement lexical avec word2vec.

Ce rapport fera état de notre appréciation sous certaines forme des résultats de ces méthodes et techniques dans un but critique. De la question de recherche, explorer le corpus, extraire de l'information, commenter les résultats et tirer conclusion.

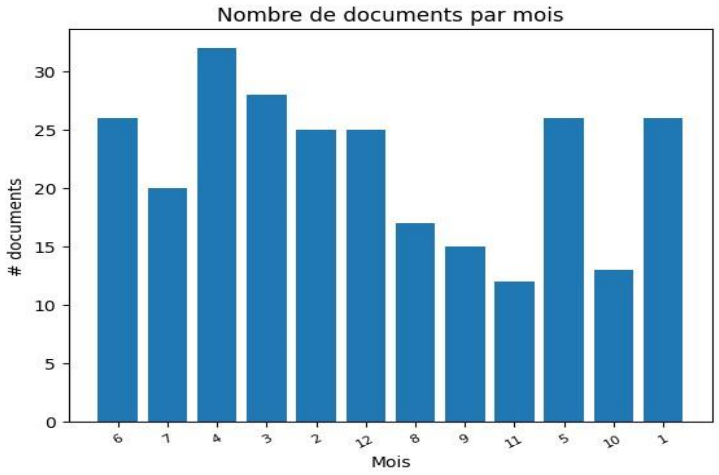
4. Etudiez votre thématique de manière transversale dans votre sous-corpus en vous aidant des différentes techniques vue au cours (exploration, fréquences, mots-clés, entités nommées, sentiment, clustering, word2vec...)

De ces technique ci-dessus j'ai retenu quelque unes dans le cadre de cette recherche.

1. **Explorer le résultat de la requête (corpus extrait) Comparaison avec le fichier (README)** Il est vrai que le fichier (README) contient toutes les informations du sous corpus extrait. N'empêche les outils d'exploration vérifie ces dernières et offre la possibilité d'avoir des statistiques globales avec python l'outil principal de notre exercice.

Nombre de documents	265
Il y a 65 exemplaires du journal Le Soir et 14 exemplaires de La Libre Belgique	
Il y a 21 exemplaires du journal Le Vingtième Siècle et 60 exemplaires de L'Indépendance belge	
Il y a 0 exemplaires du journal L'Indépendance belge (édité en Angleterre) et 37 exemplaires de La Meuse	
Il y a 2 fichiers pour la décennie 1840s , le premier document date de la décennie de 1840 Il y a 14 fichiers pour la décennie 1900s	

Les documents, pages des quotidiens, puisqu'on parle de journaux ont été plus édités en avril et mars et juin et moins de publication en novembre. Mais il n'y a pas grand-chose à dire selon mois.



2. Extraction de Mots clés pour extraire les mots clés du document avec Yake.

Au-delà de titre ou de résumé qui sont quasi impossible à accéder, il y a lieu d'avoir de mots-clés (aka keywords) pour identifier le contenu du corpus étudié. C'est un moyen simple et mais très efficace d'identifier le sujet et les concepts dans ces tonnes de donnés. Ça peut aussi être une bonne façon de catégoriser une série de textes : les identifier et les regrouper par mots-clés. Yake est une méthode payante de l'approche statistique.

Quatre années sélectionnés sur base de plus de documents ou de journaux édités.
Ls résultats sont assez parlants. Les termes extraits ont presque tous une appartenance à la QdR.
A part les jeunes filles/.. arrêtées. Les dix mots qui extraites sont dans le périmètre de sujet. Et pour les Bigrames c'est le même constat.

KB_JB421_1937-05-14_01-00002.txt	KB_JB555_1848-04-12_01-00006.txt	KB_JB837_1932-12-07_01-00002.txt	KB_JB1051_1925-06-03_01-00001.txt
[('millions',)7), 0.012188040953236704), ('Mai', 0.013755509641175917), ('Mais', 0.0160480945813719), ('jeunes filles', 0.0214889882030968 ('jeunes filles arrêtées', 0.021612518104366072), ('Colonie', 0.022078278473111 56), ('Rerum Novarum', 0.022569991071613293), ('Loterie Coloniale', 0.023323285117866514),	[('section', 0.0032483294709135103), ('section centrale', 0.003964495495301273), ("l'emprunt", 0.006086272906031132), ('sections adoptent', 0.008957119624227884), ('sections', 0.009961543710801431), ('seclion', 0.012575920356404568), ('centrale', 0.013214135204515481), ('retenue', 0.013645892891731303), ('section adopte', 0.015373562108412208), ('rentes', 0.015618458027141808),	[('Bruxelles', 0.00797101109338981), ('heures', 0.011814113993735803), ('SAINTE-BARBE EN AUVERGNE', 0.014346780213914), ('salle', 0.014976725803420509), ('décembre', 0.015361639896713134), ('Conseil', 0.026006873394129278), ('cours', 0.027462031105099552), ('jours', 0.028011464962273053), ('artistes', 0.028419019689221784), ('socialiste', 0.03043040761571868), ('parti libéral', 0.03149893265120696),	('DEUXIEME ANNEE', 0.0043512021402876), ('Conseil général', 0.015440309950447357), ('Conseil', 0.019925276001076384), ('socialistes', 0.021344558916232288), ('Etats-Unis', 0.024166741603358712), ('Friedmann', 0.028273443711159182), ('gouvernement', 0.02955076541922086), ('dettes', 0.030707169342652867), ('belge', 0.03535045280516193), ('ouvriers', 0.03863465848227941), ('Belgique', 0.03905056584507936),

('francs', 0.025329302533918286), ('cour', 0.03021356153895284),			('Friedmann MARCO FRIEDMANN', 0.04557457196558706), ('dollars', 0.045575509948076985),
Bigrame	Bigrame	Bigrame	Bigrame
['jeunes filles', 'Rerum Novarum', 'Loterie Coloniale', 'Chasseurs Ardennais', 'francs mais', 'partie défenderesse', 'filles arrêtées', 'Léon XIII']	['section centrale', 'sections adoptent', 'section adopte', 'section décide', 'contribution personnelle', "D'une retenue", 'contribution foncière', 'seclion cenlrale']	['parti libéral', 'Claire Collet', 'liste socialiste', 'Roger Liévin', 'Comité général']	['DEUXIEME ANNEE', 'Conseil général', 'plan Dawes', 'dettes belges', 'dette belge', 'parti catholique', 'soi-disant socialiste']

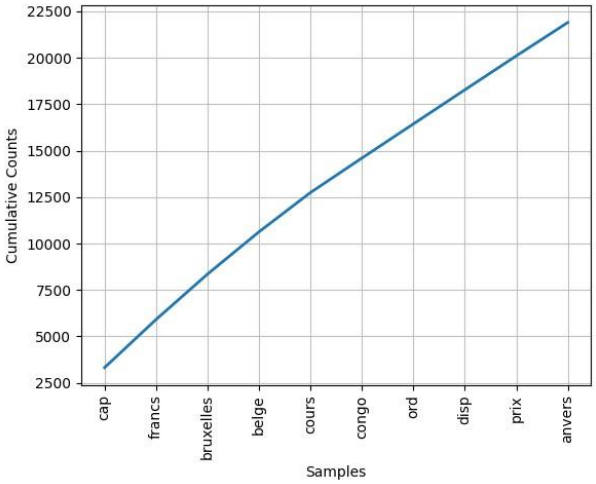
Remarque :
On pourrait le faire pour tous les documents mais qu'est-ce que cela va vraiment rapporter ! selon moi pas grand-chose car les mot clés confirment déjà la composition du corpus. A mois de vouloir chercher une information particulières d'un événement.

3. Analyse de la distribution du vocabulaire
Pour ce faire, on réunit les documents en un seul fichier corpus tel un grand fichier texte.

Le corpus généré des 265 documents contient 2,730,290 mots. Mais parmi ces mots il y en a qui sont comme on a fait plus tôt l'extraction de mots clés, mais aussi des mots vides de sens qu'on appelle des stopwords. En français ces mot vides sont entre autres des articles , prépositions, verbes, adverbes etc... Aussi faut des formes non grammatical lors de l'océrisation. Pour mieux faire l'analyse il faut vider le corpus de ces mots et formes. On obtient la taille du vocabulaire avec 864,917 words kept (124,308 different word forms)

Récupération des mots les plus fréquents
[('cap', 3321), ('francs', 2588),('bruxelles', 2444), ('belge', 2276), ('cours', 2115), ('congo', 1858), ('ord', 1838), ('disp',
1837), ('prix', 1828), ('anvers', 1797)]

['liflercrfcdi', 'drapouge', 'bruxelleô', 'beïge', 'eaens', 'œllel' 'trustera','ageuts', 'peuveut', 'désorganiserait'], 'ïringoler', 'troublerait', 'drainé', 'bénéficos','detti', 'goûv', 'jrtm','piëî',
'beîtfb', 'réclamerait', 'iallut', 'vcuicut', 'prêtée','lappelons',
'ftats','duibois' 'paert', 'formols','brimés', 'coopératistes']



Les mots en rouge sont des happaxes, des mots qui se rencontrent une seule fois dans le corpus.
Après comme j'ai tantôt attiré votre attention, les autres sont des erreurs d'océrisation. Et ce, pour voir les mots les plus long du texte sont tous des erreurs de la sorte.

4. Nuages de mots

Une façon de désigner un ensemble de mots ou expression-clé qui a pour fonction de décrire ou de classer l'information. Les mots qui apparaissent le plus fréquemment dans le texte source se distinguent par la taille des termes .
Il s'agit en grossomodo d'afficher les termes avec plus d'occurrence qui sont présents dans le corpus

Trois années avec le plus de document et appliquer l'algorithme là-dessus et générer le nuage de mots.
Et une année isolée pour voir ce que ça donne.
Comme on peut le constater les termes qui découlent du jargon de la finance et des institutions étatiques bancaires sont afficher en grande lettre, synonyme de représentativité ou de termes qui représentent le document.

Quant au dernier la présence du mot « aujourd » et « hui » et « meeus » ne sont pas trop pertinentes , du côté de meeus, soit le seul document donc par trop de significatif de le faire sur une petite quantité de document et le terme aujourd hui, pourrait bien être écarté du corpus comme mots vides.

Figure 2. 1927

Figure 1. 1933

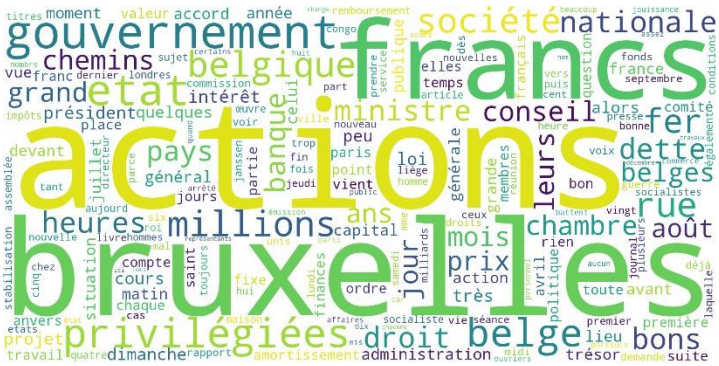


Figure. 3 1926

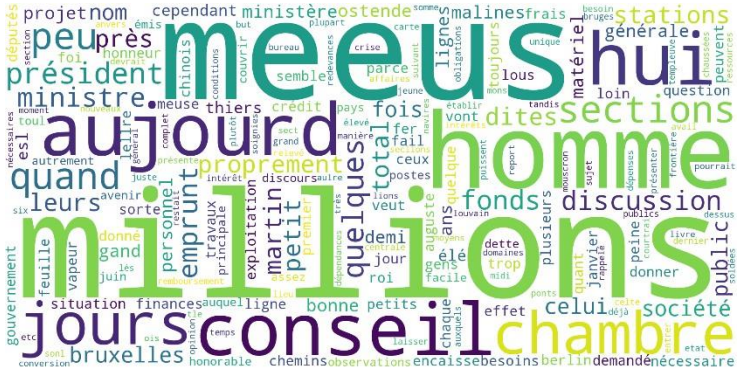


Figure 4. 1840

5. Appliquer la reconnaissance d'entités nommées sur notre corpus

Reconnaissance d'entités nommées avec SpaCy (NER) , un outil très puissant.

Personnalité	Organisation	Lieu
Friedmann apparait 7 fois dans le corpus Poullet apparait 3 fois dans le corpus P. O. B. apparait 3 fois dans le corpus Janssen apparait 3 fois dans le corpus Coppens apparait 3 fois dans le corpus MM. Vandervelde apparait 2 fois dans le corpus Wauters apparait 2 fois dans le corpus M. Poullet apparait 2 fois dans le corpus M. Segers apparait 2 fois dans le corpus	Conseil apparait 7 fois dans le corpus Etats- Unis apparait 4 fois dans le corpus Aveo apparait 2 fois dans le corpus Internationale des Employés apparait 2 fois dans le corpus BANDITS apparait 2 fois dans le corpus Congrès apparait 2 fois dans le corpus Sénat apparait 2 fois dans le corpus Solvay apparait 2 fois dans le corpus	Belgique apparait 12 fois dans le corpus Etat apparait 10 fois dans le corpus Etats-Unis apparait 9 fois dans le corpus Bruxelles apparait 7 fois dans le corpus Allemagne apparait 5 fois dans le corpus Londres apparait 5 fois dans le corpus Russie apparait 5 fois dans le corpus Europe apparait 4 fois dans le corpus

M. Theunis apparait 2 fois dans le corpus	Chambre apparait 2 fois dans le corpus Parti Communiste Beïge apparait 1 fois dans le corpus	Amérique apparait 3 fois dans le corpus Vandervelde apparait 3 fois dans le corpus

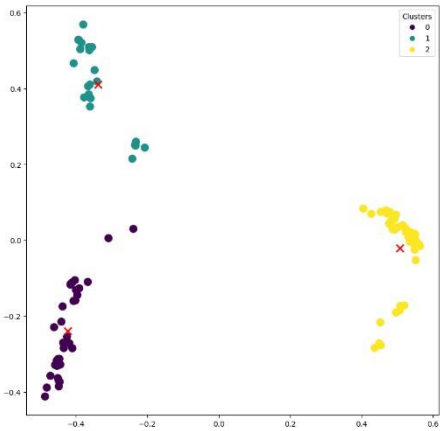
Dans le traitement du langage naturel la reconnaissance d'entité nommée fait partie. Avec pour objectif premier de traiter données structurées et non structurées et classer ces entités nommées dans des catégories prédéfinies. Certaines catégories courantes comme notre résultat ci-dessus se rapporte au nom, au lieu, aux organisations et d'autres comme les entreprises, l'heure, les valeurs monétaires, les événements, etc.

En quelques mots, NER s'occupe de : détection d'entités nommées – Identification d'un mot ou d'une série de mots dans un document. Et de la classification des entités nommées – Classement d'une entité détectée dans des catégories prédéfinies.

Et comme vous pouvez le constater les résultats sont pertinents.

Après il y a aussi flair avec un F1-Score: 90,61 (WikiNER) qui prédit tout aussi bien.

6. Clustering - Non supervisée



Our regroupement est un machine non supervisée qui permet de faire des dans un corpus sur base de similarité de documents. Les mots font un prétraitement avec TF-IDF qui les vectorise puis …

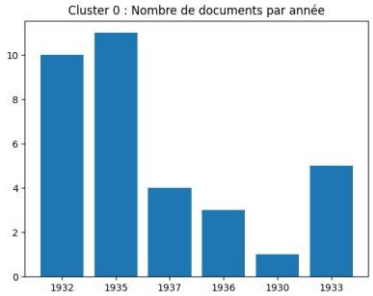
Le clustering, objet d'apprentissage non supervisé. Des méthodes du machine Learning, qui permettent à un ordinateur d'en apprendre luimême sur les données qui lui sont fourni.

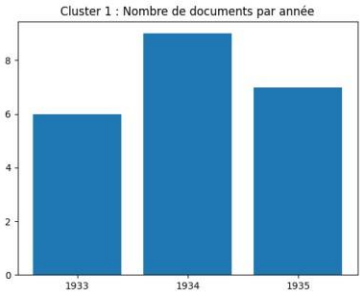
D'entrée de jeu le clustering : avec la KMeans, une des méthodes centroïdes très simple dès l'abord il faut seulement choisir une variable K qui détermine le nombre de groupe souhaité. Bien que ce n'est pas toujours optimal, on peut faire appel à ***Elbow***, une technique pour trouver le meilleur K. Un bon clustering nécessite de centres très distancés et des données concentrées autour de ces points.

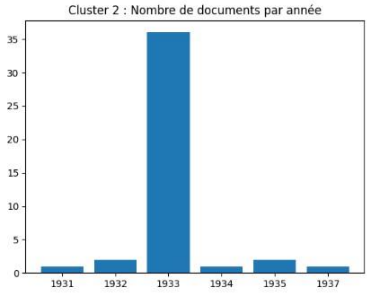
Pour ce faire à partir d'une décennie « 1930 car contient plus de documents » un nombre important de fichiers textes on applique un fonction de prétraitement et on les instancie avec tfidf pour les vectoriser. Puis on les transforme en matrice pour mieux représenter leur dimensions. Puis on applique la fonction fitpredict pour générer les **cluster**. A la base le nombre de dimension est énorme, pour le visualiser on applique une autre fonction APC pour les réduire en 2 dimensions, mieux interprétable pour l'être humain.

Pour la décennie que j'ai choisie, la fonction Elbow valide approximativement un K=3, acceptable avec des centres éloignés et des données concentrée autour des centres. Pour les 3 cluster, le visuel reste concluant.
A noter que TF-IDF est ici pour voir dans quelle mesure le terme représente bien le document. Dont (max_df et min_df) sont à paramètre pour éliminer les stop-words. Un max_df de 0.7 tend à éliminer une quantité importante de stop-words.

Nombre de documents dans le cluster
cluster no 0 : 34 cluster no 1 : 22 cluster no 2 : 43





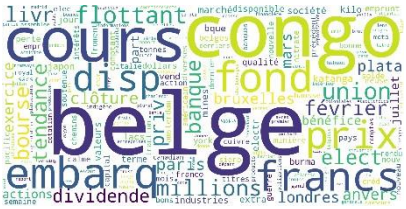


Les mots clés pour le cluster 0

Keywords, cluster n° 0 : [('Banque Nationale', 4.0989138730633755e05), ('Banque Générale Belge', 6.843825937662579e-05), ('Dette belge', 7.373734618719133e-05), ('Cours Cours Cours', 9.628585228334842e05), ('Gouvernement', 0.00013990752842995497)]

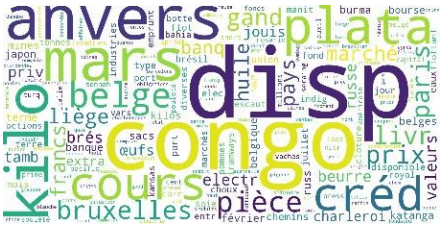
Bigrammes récurrents, cluster n° 0 : ['Banque Nationale', 'Dette belge']
Trigrammes récurrents, cluster n° 0 : ['Banque Générale Belge', 'Cours Cours Cours']

Les mots clés pour le cluster 1



Keywords, cluster n° 1 : [('Bque Congo Belge', 3.50317740077729 45e05), ('Congo Belge Bque', 4.2038128809327536e-05), ('cap', 8.385710585941606e-05), ('Congo Belge', 9.025099762965271e-05), ('Nord Belge cap', 0.00010369927017678926)]
Bigrammes récurrents, cluster n° 1 : ['Congo Belge']
Trigrammes récurrents, cluster n° 1 : ['Bque Congo Belge', 'Congo Belge Bque', 'Nord Belge cap']

Les mots clés pour le cluster 2



Keywords, cluster n° 2 : [('Plata Plata Plata', 4.378259534200089e-07), ('Cie Congo Belge', 9.813129763387104e-07), ('Plata avril Plata', 2.1186479916005434e-06), ('avril Plata Plata', 2.2699799910005826e06), ('Plata mars Plata', 2.470935534589287e-06)]
Bigrammes récurrents, cluster n° 2 : []
Trigrammes récurrents, cluster n° 2 : ['Plata Plata Plata', 'Cie Congo Belge', 'Plata avril Plata', 'avril Plata Plata', 'Plata mars Plata']

Pour plus d'analyse l'application de l'algorithme pour l'extraction de mots clés et de nuage de mots, nous porte à apprécier les termes dominants des différents cluster. Une très bonne méthode pour résumer l'information.
Et une remarque très importante , les nuages de mots effectués sur cluster faites ressortir les mêmes termes pour ne dire qu'ils ont une ressemblance frappante.

7. Word Embbeding avec Word2vec - Non supervise

Le Word embeddings ou plongement lexical prend forme dans le sens que chaque mot d'un corpus de textes est représenté par un vecteur, des nombres dans un espace vectoriel de dimension 'x' en se basant sur différents critères de contextualité. Il désigne un ensemble de méthode d'apprentissage visant à représenter les mots d'un texte par des vecteurs de nombres réels. Effectivement en se débarrassant du « fléau de la dimension » (curse of dimensionality) : on réduit les milliers de dimensions d'un mot à un nombre fixe moins élevé (dizaines/centaines) intuitivement, les mots aux sens similaires seront proches dans l'espace vectoriel.

Pour ce faire on fait appel à Word2Vec .
Pour apprendre des termes similaires dans un corpus non supervisé donné, Word2Vec est très utilisé. Que ce soit la sémantique, ou pour créer des incorporations de mots par

des représentations sémantiques distributionnelles dans les applications NLP, Semantic Analysis, Text Classification et bien d'autres.
Mise à part, sa limitation uni-gramme. Word2Vec est très performant, pour prédire un mot donné en fonction de son contexte (CBOW), ou de prédire un contexte environnant en fonction d'un mot donné (SkipGram). Sur notre corpus plus restreint les résultats ne semblent pas trop concluants.

Au point de départ on se sert d'un objet **phrases** pour extraire les N-grammes de mots formant un dictionnaire.
Des 2730290 du corpus 953417 (bi-grammes) clés qui sont des termes observe entre autres, possible grâce avec l'objet phrases. Et n'importe clé pris au hasard tombe sur un mot exemple la clé 6754 est **la_grace**, et son score d'occurrence est de « 5 ». Mais dans un contexte plus poussé l'outils **phraser** produit des clés sous forme groupe soit de tri-grammes concaténées pourvu que cela fasse du sens. De ce fait en réduisant la dimension de la matrice on capture le contexte, la similarité sémantique et syntaxique (genre, synonymes, …) d'un mot.

Observation :

1) Calcul de la similarité entre 2 termes données.

corpustp4.model vector_size=32, window=4, min_count=5		corpusdb.model vector_size=32, window=5, min_count=7	
model.wv.similarity("cours", "obligations") model.wv.similarity("obligations", "obligation") model.wv.similarity("france", "belgique")	0.51527447 0.6349034 0.79148525	model.wv.similarity("cours", "obligations") model.wv.similarity("obligations", "obligation") model.wv.similarity("france", "belgique")	0.50232005 0.6350837 0.84518313

Des deux models, même en jouant sur les paramètres on arrive à des résultats quasi semblables.

2) Recherche de mots les plus rapproches d'un terme donnée.

corpustp4.model	corpusdb.model

model.wv.most_similar("soir", topn=5) [('pre', 0.949506938457489), ('matin', 0.9382753968238831), ('siecle', 0.9382133483886719), ('capitaine', 0.9332357048988342), ('cinema', 0.9332219362258911)]	model.wv.most_similar("soir", topn=5) [('matin', 0.9528449773788452), ('siege', 0.9342163801193237), ('drame', 0.9251589179039001), ('courrier', 0.9147465229034424), ('roi', 0.9114739894866943)]
model.wv.most_similar("fer", topn=5) [('des_chemins', 0.830240786075592), ('chemin', 0.825129508972168), ('actions_des_chemins', 0.7695838212966919), ('societe', 0.7685903310775757), ('du_chemin', 0.7618299126625061)]	model.wv.most_similar("fer", topn=5) [('chemin', 0.7993881702423096), ('du_congo', 0.7872655391693115), ('des_chemins', 0.778247058391571), ('du_chemin', 0.7752842307090759), ('charleroi', 0.7616026401519775)]
model.wv.most_similar("esclaves", topn=5) [('prunes', 0.9437491297721863), ('academiciens', 0.9427877068519592), ('bonifications', 0.9412000179290771), ('accueillir', 0.9410020709037781), ('opinions', 0.9402633309364319)]	model.wv.most_similar("esclaves", topn=5) [('plus_belles', 0.9268314242362976), ('moindres', 0.9179300665855408), ('cours_actuels', 0.9111868143081665), ('yeux', 0.9066286087036133), ('incidents', 0.9065471887588501)]
model.wv.most_similar("actions", topn=5) [('obligations', 0.8504869937896729), ('banques', 0.8275479078292847), ('jouissance', 0.8054082989692688), ('depots', 0.7919761538505554), ('divers', 0.7897666692733765)]	model.wv.most_similar("actions", topn=5) [('obligations', 0.8504869937896729), ('banques', 0.8275479078292847), ('jouissance', 0.8054082989692688), ('depots', 0.7919761538505554), ('divers', 0.7897666692733765)]

Je m'attendais pour soir et matin des négatifs. Mais l'explication est de mise puis qu'on parle de similarité. Les models répondent pas, dans le cas de sous corpus, avec des textes trop homogènes le wordembedding semble ne pas être trop approprié. Après un autre problème que je constate c'est qu'il renvoie quand même des réponses bien que cela ne colle pas avec la réalité.

En somme la recherche sur la question très précise de la dette belge est concluante. Qu'il s'agit du nombre de document constituant le corpus, des mot clés, d'entités nommées et de clustering. Et ce, malgré l'importance du nombre de documents les algorithmes ont montrés une forte homogénéité de ces derniers. Bonne pour certains outils et moins bonne pour d'autres. Attrayant et très pratique, trois (3) cluster distinctes résument une bonne partie de l'information. L'entrainement de deux modèles m'a semblé nécessaire, bien que les résultats ne soient pas trop satisfaisants. Néanmoins on peut constater l'efficacité de ces méthodes dans l'extraction d'information. Les outils de Tac sont de grandes portées et peuvent aider à retracer ou à extraire de l'information dans des données structurées ou non. Toutes fois il faut à la base user de bonne pratique de prétraitement dans la numérisation et d'océrisation pour garantir la qualité de ces données.

Web/Bibliographie

https://www.kbr.be/fr/projets/camille/
https://stackoverflow.com/questions/44159645/error-cat-is-not-recognized-as-an-internal-or-external-command Tac document de cours https://www.camille-ulb-kbr.be/
https://huggingface.co/flair/ner-french https://spacy.io/api