Les techniques du NLP pour la recherche en sciences de gestion

Sophie Balech, C. Benavent

NLP text mining V4.0 - une introduction - cours programme doctoral

Preprint · December 2019

DOI: 10.13140/RG.2.2.34248.06405

2 authors:

Sophie Balech
IAE Amiens
3 PUBLICATIONS **1** CITATION
SEE PROFILE

Christophe Benavent
University Paris Nanterre
54 PUBLICATIONS **243** CITATIONS
SEE PROFILE

Les techniques du NLP pour la recherche en sciences de gestion.

Sophie Balech – Université d'Amiens

Christophe Benavent – Université Paris Nanterre

Résumé :

L'objet de ce chapitre est d'introduire aux techniques de traitement du langage naturel, à l'analyse textuelle, telles que les développements de la fouille de données et de la linguistique la définissent, automatisant en prenant avantage des propriétés distributionnelles du langage. Largement automatisées, les techniques de traitement du langage naturel enchaînent des séries d'opérations qui vont de la constitution du corpus, à son annotation, pour se traduire par des modèles de représentations et de qualifications. Ces méthodes sont désormais largement accessibles par les librairies des langages r et python. Elles permettent d'exploiter les larges corpus que la digitalisation permet de constituer : commentaires consommateurs, bases de news, rapports d'activité, compte-rendus d'entretiens L'objet de ce texte est essentiellement technique, cependant sans donner de mode opératoire. Il indique des méthodes génériques exploitables via r et leur contexte d'utilisation. C'est un bref manuel de l'analyse textuelle moderne pour la recherche en sciences de gestion.

Key-words : tsne, word2vec, tdidf, LSA, LDA, Sentiment, Dépendances syntaxiques, Diversité lexicale, texte, lisibilité, readability, glove, textmining, tal, corpus,ML

Abstract

The purpose of this chapter is to introduce natural language processing techniques and textual analysis, such as the developments of data mining and linguistics define it, automating it by taking advantage of the distributional properties of language. Largely automated, natural language processing techniques sequence a series of operations from the constitution of the corpus to its annotation, resulting in representation and qualification models. These methods are now widely available through the r and python language libraries. They make it possible to exploit the large corpus that digitisation makes it possible to build: consumer comments, news bases, activity reports, interview reports. The purpose of this text is essentially technical, however without giving any operating method. It indicates generic methods that can be used via r and their context of use. This is a short manual of modern textual analysis. For business research.

Key-words : tsne, word2vec, tdidf, LSA, LDA, Sentiment, syntaxic dependency, Lexical diversity, Readability, NLP, glove, textmining, corpus, ML

Les techniques du NLP pour la recherche en sciences de gestion.

Le texte connaît une double révolution. La première est celle de son système de production, la seconde est celle de sa lecture. Il se produit désormais tant de textes que personne ne peut plus tout lire, même en réduisant l'effort à sa sphère d'intérêt et de compétence. La production primaire de texte se soumet ensuite à ceux qui en contrôlent les flux et en exploitent les contenus, qui les mettent en avant ou les écartent, définissant la composition de ce que chacun va lire. La révolution de la lecture est venue avec les moteurs de recherche et les pratiques de curations (Fotopoulou et Couldry 2015), c'est une lecture sélectionnée et digérée par les moteurs de recommandation. ((Aggarwal 2016)).

Pour le chercheur qui étudie les organisations et les marchés, cette révolution textuelle offre de nouvelles opportunités d'obtenir et d'analyser des données pour vérifier ses hypothèses. La production abondante d'avis de consommateurs, de discours de dirigeants, de compte-rendus de conseils d'administration, d'articles techniques, de brevets rencontre une multiplication de techniques d'analyse, provenant de la linguistique computationnelle, de la fouille de données, de la traduction automatique, de la psychologie, pour traiter cette abondance. Elles permettent d'aller plus loin que l'analyse lexicale traditionnelle en incorporant des éléments syntaxiques, sémantiques, et pragmatiques, proposés par l'ensemble des outils des techniques de traitement du langage naturel.

Il se dessine surtout une nouvelle approche méthodologique qui prend place entre l'analyse qualitative, et les traditionnelles enquêtes par questionnaires capables de traiter des corpus d'une taille inédite. Le travail de (Humphreys et Wang 2018) en donne une synthèse dans le cadre d'un processus qui s'articule autour des différentes phases d'une recherche : la formulation de la question de recherche, la définition des construits, la récolte des données, l'opérationnalisation des construits, et enfin l'interprétation, l'analyse et la validation des résultats obtenus. Dans le champ du management, on trouvera des synthèses pour la recherche en éthique (Lock et Seele 2015), en management public (Anastasopoulos, Moldogaziev, et Scott 2017) ou en organisation (Kobayashi et al. 2018).

Ces développements sont favorisés par un environnement fertile dont trois facteurs se renforcent mutuellement.

- Le premier est l'**expansion de deux langages,** proprement statistiques pour r et plus généraliste pour python. Le propre de ces langages est, prenons le cas de r, de permettre d'élaborer des fonctions, dont un ensemble cohérent pour réaliser certaines tâches peuvent être rassemblées dans une bibliothèque. Coder une analyse revient ainsi à jouer avec un immense jeu de lego, dont de nombreuses pièces sont déjà pré-assemblées. D'un point de vue pratique, les lignes d'écriture sont fortement simplifiées permettant à un chercheur sans compétence de codage d'effectuer simplement des opérations complexes. L'autre conséquence est la croissance exponentielle du nombre de packages disponibles : près de 20 000 pour r. Dans le domaine de l'analyse du texte, on citera l'ancêtre "tm", le sophistiqué "quanteda"(Benoit et al. 2018), l'indispensable "cleanNLP"(Arnold 2017a), ou l'astucieux "tidytext"(Arnold 2017b)dimension langagière se traduit aussi dans l'édition : le rôle du markdown et des carnets Jupyter[1]. De plus de nouvelles familles de techniques se généralisent et sont accessibles en open-source au travers de langages tels que python ou r, qui est privilégié dans ce chapitre.
- Le second, intimement lié au premier, est la constitution d'une large **communauté** de développeurs et d'utilisateurs qui se retrouvent aujourd'hui dans des plateformes de dépots (Github, Gitlab), d'Arxiv, de plateformes de type Quora (StalkOverFlow), de tutoriels, de blogs (BloggeR) et de journaux (*Journal of Statistical Software*). Des ressources abondantes sont ainsi disponibles et facilitent la formation des chercheurs et des data scientistes. Toutes les conditions sont réunies pour engendrer une effervescence créative.

1Au-delà des modèles et des méthodes de manipulation des données, il y a aussi une conception du flux de tâches et de sa gestion qui s'est mise en place. Les carnets Jupiter ou les documents Markdown jouent un rôle essentiel : mettre dans un document le code de traitement, le commentaire organisé, les références bibliographiques, la représentation élaborée des résultats. Les méthodes de base sont une chose, leur arrangement dans un environnement stable, intuitif, souple, et rigoureux une autre. On consultera https://bookdown.org/yihui/rmarkdown/ pour en approfondir la technique.

- Le troisième est la **multiplication des sources de données** et leur facilité d'accès. Les données privées, et en particulier celles des réseaux sociaux, même si un péage doit être payé pour accéder aux APIs, popularisent le traitement de données massives. Des plateformes de concours telles que Kagle proposent de nombreux sets de données massives. Le mouvement des données ouvertes (*open data*) facilitent l'accès à des milliers de corps de données : retards de la SNCF, grand débat, le formidable travail de l'Insee, le European Social Survey etc.

Dans ce chapitre, on choisit de présenter les différentes facettes de ce qui s'appelle TAl, NPL, Text Mining, dans une approche procédurale qui suit les principales étapes du traitement des données. On rendra compte à chaque étape des techniques disponibles, et on illustre d'exemples. Nous suivrons ici une approche plus fidèle au processus de traitement des données, lequel peut connaître une stratégie inférentielle et exploratoire, tout aussi bien qu'une stratégie hypothético-déductive. Nous resterons agnostiques sur cette question, restant délibérément à un niveau technique et procédural.

1. Construire le corpus
    a. Corpus primaire ou secondaire
    b. Construire un corpus à partir du web : *scraping* et APIs
2. Décrire le corpus
    a. Travailler le corpus brut
        i. Pré-traitement et opérations de normalisation
        ii. Tokenisation
        iii. Matrice termes-documents et analyse de fréquence
    b. Travailler le corpus pré-traité
        i. Les indicateurs de qualité
        ii. L'annotation du langage naturel
        iii. La détection des sentiments
        iv. La vectorisation des termes
3. Découvrir le sens du corpus
    a. Cooccurrences et similarités
    b. L'analyse de graphe
    c. Les ancêtres : l'analyse factorielle des correspondances et le *clustering*
    d. La méthode T-sne
    e. L'identification des topics latents par la méthode LDA
4. Aller plus loin avec le machine learning
    a. Le processus de modélisation
    b. Une application à l'analyse des sentiments

1. CONSTRUIRE LE CORPUS

La première étape est la constitution d'un corpus : un ensemble de documents numériques ou numérisés. Les documents traditionnels nécessitent d'être numérisés, tels les romans pour un spécialiste de littérature, ils peuvent résulter de transcription d'entretien sonore ou vidéo. Ils sont de plus en plus directement collectables sur le web ou accessibles via les APIs. La nouveauté apportée par ces dernières est le volume : des millions, voire des dizaines millions de textes peuvent ainsi être disponibles dans des formats standardisés.

1. Corpus primaire ou secondaire

Un corpus est l'ensemble des documents qui se prêtent à l'analyse. Pour l'historien c'est un ensemble de cotes, pour le professeur de littérature c'est un ensemble d'œuvres, pour le sociologue des transcriptions d'entretiens, pour le linguistique un ensemble de paroles. Ce sont des matériaux classiques qui bénéficient de la numérisation. Pour la recherche en gestion, le corpus peut-être issu d'entretiens, de questions ouvertes dans des questionnaires, de documents internes aux entreprises, des commentaires des consommateurs sur les réseaux sociaux. La nouveauté apportée est que si une analyse thématique manuelle est possible sur un corpus limité

(une trentaine d'entretiens par exemple), l'analyse automatique vient au secours pour un grand nombre de documents.

Les corpus primaires sont ceux constitués par le chercheur lui-même, dans une interaction avec son terrain en en employant des dispositifs d'échantillonnage et de recueil des données qu'il a lui-même construits.

- Les transcriptions d'entretiens sont une source de données évidente, qu'il s'agisse d'analyse lexicale ou même thématique. La possibilité d'automatiser cette tâche, par des techniques de "*speech-to-text*" de plus en plus élaborées, ouvre aux chercheurs des approches plus extensives de la pratique de l'interview. Si la doctrine courante d'échantillonnage qualitatif s'appuie sur un principe de saturation dont on s'aperçoit qu'il ne nécessite que quelques dizaines d'entretiens, les chercheurs peuvent être intéressés par une connaissance plus précise de la distribution des opinions
- Les questionnaires ouverts peuvent devenir une source précieuse dans les enquêtes et associer de manière nouvelle le large nombre des répondants et la liberté d'expression.
- Mais on peut imaginer d'autres dispositifs et milles variantes : l'enregistrement de conversation en mode chat, toutes sortes de dispositifs « gamifiés » tel

Les corpus secondaires sont ceux rassemblés dans des bases de données suffisamment structurées pour en extraire des échantillons de documents nombreux par recherche, filtrage et échantillonnage.

- Les bases de dépêches et annuaires d'entreprises : Factiva ou Techcrunch pour ne prendre que deux exemples fournissent d'abondantes ressources sur les comportements des organisations, leur historisation, et les commentaires et analyses qu'ils suscitent.
- Les bases de données bibliographiques, et leur extrême structuration en abstract, keywords, références, ont donné naissance à la bibliométrie, dont les développements seront certainement textuels avec s'être largement appuyées sur l'analyse de réseaux.
- Une source de plus en plus importante est celle de l'*open data public (Mabi 2015)*ba. Les données du grand Débat, forment un exemple intéressant de plateforme *civic tech(May et Ross 2018)*may. L'*open data* public : carte, cadastre, fichiers de transaction immobilières, log de prise charge de taxi se développent fortement.
- Les réseaux sociaux, forum et plateformes de reviews sont des sources précieuses pour suivre l'opinion, ce qu'on dit d'une marque, les opinions qui se construisent sur des événements, les plaintes et les encouragements.

Le problème du biais de sélection est le talon d'Achille de ces données. Il résulte de l'interaction des dispositifs de collectes de données avec les comportements des sujets d'observation et la question du degré de participation est cruciale. Aussi vastes soient-elles, elles ne sont ni exhaustives (même vastes, ces données ne couvrent qu'une partie du champ), ni représentatives (les populations ne sont pas bien définies et les auteurs pas forcément bien identifiés). Et c'est sans doute un futur axe de développement méthodologique. D'autres méthodes statistiques devront être élaborées(Meng 2018).

2. Construire un corpus à partir du web : *scraping* et API

Une nouveauté apportée par internet est la possibilité de construire le corpus directement à partir des données accessibles en ligne. Deux approches de constitution des corpus sont possibles : par *scraping* ou grâce aux APIs (*Application Programming Interface*).

Le *scraping* est l'activité qui consiste à moissonner les informations disponibles sur le net en simulant et en automatisant la lecture naturelle d'un site ou d'une page web par un "butineur". Elle consiste à construire un robot capable de lire et d'enregistrer les informations disponibles sous forme html puis à les distribuer (*parsing*) dans des tableaux structurés, selon une stratégie d'exploration du web préalablement définie.

De nombreuses ressources sont disponibles, mais pour en rester à r, le package `rvest` permet de réaliser des extractions simples mais suffisantes pour de nombreux usages.

Les caractéristiques clés du *scraping :*

- La nécessité de programmer de manière ad hoc, en fonction des spécificités de chaque site.
- Des stratégies mécaniques, en boules de neige.
- Le risque de *deny of service*, c'est-à-dire de saturer ou de parasiter un système et de s'exposer à ses contre-mesures.

- Les conditions légales ne sont pas homogènes et relèvent de différents droits : de la propriété intellectuelle, du respect de la vie privée, du droit de la concurrence. Cependant des facilités et tolérances sont souvent accordées quand c'est dans un objectif de recherche et que des précautions minimales d'anonymisation ou de pseudonymisation sont prises, et que les règles de conservation et de destruction des données sont précisées.
- La question éthique va au-delà du droit, elle concerne les conséquences de cette action sur l'évolution d'ensemble. On notera qu'elle participe à la "robotisation" du web (plus de 50 % du trafic résulterait de la circulation de bots et qu'elle fait l'objet de contre-mesures.

L'utilisation d'API lève l'ambiguïté légale qui accompagne le *scraping* et peut ainsi paraître comme plus civilisée. Elle nécessite naturellement que le gestionnaire de la base de données fournisse les moyens de s'identifier et de requêter, elle peut avoir l'inconvénient d'un accès payant. Pour n'en donner qu'un exemple l'API de Twitter, il en coûte 399 $ dollars par mois pour moissonner jusqu'à 5 millions de tweets en au plus 500 requêtes.

Exemple 1 : Requête à l'API Twitter via le package 'twittr'

```
library (twittr) # on appelle la librairie twittr qui permet les requêtes
consumerKey<-"Xq..."   #paramètres requis par l'API de twitter (Ouvrir un compte au préalable)
consumerSecret<-"30l..."
access_token<-"27A..."
access_secret<-"zA7..."

#fonction d'initialisation des requêtes
setup_twitter_oauth(consumerKey, consumerSecret, access_token,access_secret)

#recherche des tweets récents depuis une date donnée.
tweets1 <- searchTwitter("#IA", n = 2000, lang = "fr", resultType = "recent", since = "2019-08-01")

#transformer en data frame exploitable par r
tweets_df1 <- twListToDF(tweets1)
```

2. DÉCRIRE LE CORPUS

Le texte brut se prête rarement et directement à l'analyse. Il va falloir le malaxer, le brasser, le filtrer, le préparer, le purifier. C'est le but du pré-traitement (*pre-processing*) des données textuelles. Il s'agira autant de simplifier le texte, de le nettoyer, de le réduire, que de l'enrichir en annotant ses termes d'attributs particuliers relatifs à leur nature lexicale ou syntaxique. Il faut un texte bien tempéré pour en écouter la musique. Ensuite, on pourra appliquer différents traitements aux textes pour une analyse plus précise de leur composition.

1. Travailler le corpus brut

a. Pré-traitement et normalisation du texte

Cela commence par des opérations élémentaires de normalisation du texte qui réduisent le nombre de formes, mais peuvent aussi détruire de l'information. Le package classique dans l'environnement r est 'tm', les packages 'tidytext' ou 'quanteda' proposent eux des grammaires simplifiées. Les principales opérations sont les suivantes :

- Mettre le texte en minuscule, mais au risque de perdre les capitales initiales qui indicent les noms communs.
- Supprimer la ponctuation, mais du même coup déstructurer les phrases.
- Éliminer les nombres et dates.
- Repérer (et éliminer) les liens URL et les mentions de personnes.
- Éliminer aussi les mots sans signification que l'anglais dénomme par "*stopwords*" avec des dictionnaires.
- Repérer et traiter les émoticônes. Le langage numérique a la particularité de réintroduire des éléments iconiques dans une écriture culturellement alphabétique.
- Dans le cas d'un langage vernaculaire la correction orthographique est indispensable. Le package

‘hunspell’ permet de repérer les fautes.

b. Tokenisation

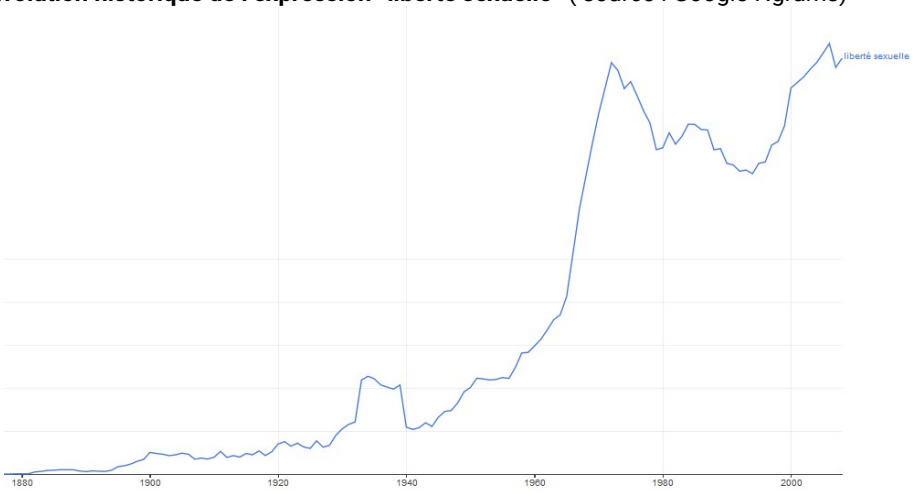
Cette première phase de lissage, de correction, de nettoyage, des données textuelles laisse la place à une seconde série d’opérations qui vise à définir l’objet de l’analyse : c’est la tokenisation des documents qui consiste à identifier les unités de textes élémentaires qui peuvent être des mots, mais aussi des lettres, des syllabes, des phrases, ou des séquences de ces éléments.

Chaque document devient alors une liste ordonnée (ou non) de termes élémentaires : les tokens. Nous passons d’un plan de données qui liste des documents (et les associe éventuellement à des auteurs), à un plan qui associe un document à une série d’attributs qui sont ses éléments unitaires (lettres, mots, syllabes, phrases).

Si les mots sont des unités de sens évidentes, les paires, les triplets de mots le sont aussi. Ainsi dans l’analyse du corpus du Grand Débat National, l’expression « mille feuille administratif » apparaît fréquemment, il est à lui seul une unité signifiante. Les n-grammes sont ainsi des suites de 1,2, … n lettres, syllabes ou mots dont on va mesurer la fréquence d’apparition dans le corpus.

L’utilisation des n-grammes peut être directe, comme l’illustre l’exemple du bigramme “liberté sexuelle” dans la base Google Ngrams, dont nous laissons au lecteur le soin de l’analyse.

Exemple 2 : Évolution historique de l’expression “liberté sexuelle” (source : Google Ngrams)

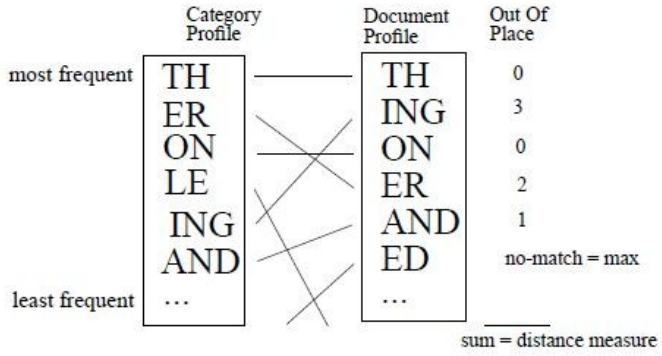


Il peut être plus sophistiqué comme l’illustre celui de l’utilisation des n-grammes pour détecter la langue d’un texte.

Exemple 3 : Détection des langues et comparaison des spectres de n-grammes de 'textcat'

Comment reconnaître une langue sans la connaître ? Une méthode simple consiste à compter ses éléments. Un mot est une série de lettres, et si on dispose d'un corpus de cette langue, on peut calculer la fréquence des n-grammes et obtenir leur distribution par ordre de fréquence, ce qui constitue leur signature. En comparant la distribution des n-grammes d'un texte particulier à celle de différentes langues, par un simple calcul de distance, on peut attribuer le texte à sa langue. Dans r la library 'textcat' réalise cette tâche.

FIGURE 3. Calculating The Out-Of-Place Measure Between Two Profiles



Note: These profiles are for explanatory purposes only and do not reflect real N-gram frequency statistics.

Dans la pratique le nombre des n-grammes est explosif : n uni-grammes forment potentiellement n^2 bi-grammes et n^3 tri-grammes. Mille mots, ce qui est faible, génèrent 1 milliard de trigrammes dont seule une très petite fraction se manifeste dans le corpus. Une sélection des bigrammes est donc nécessaire (a fortiori pour les n-grammes), c'est l'objectif des techniques dites de **collocation**. Elles consistent simplement à calculer des indices de liaison entre des termes spécifiques. On trouve cette fonction notamment dans 'quanteda'. Celle-ci consiste à calculer des indices de liaisons entre deux, trois ou plus termes spécifiques. La méthode proposée dans 'quanteda' utilise le lambda et le test z de Wald.

c. Matrice termes-documents et analyse de fréquence

La représentation du texte se traduit par un tableau fondamental : le tdm (*term-document matrix*) ou dtm (*document-term matrix*), un tableau document*termes ou termes*document. Dans sa forme élémentaire chaque unité est binaire : présence ou absence du terme dans le texte.

Cette opération permet de construire un dictionnaire, et de filtrer les tokens sur la base de la fréquence des éléments. Le plus souvent on écrête les termes très peu fréquents et ceux trop fréquents qui se retrouvent dans l'ensemble des documents. Les premiers apportent une information singulière, les seconds n'en apportent plus.

Il est parfois d'usage d'utiliser une transformation tf-idf, pour *term frequency-inverse document frequency*, qui pondère les fréquences d'apparition des termes par leur nombre d'occurrences dans l'ensemble des documents. La pondération tf-idf permet de contrebalancer l'importance d'un mot utilisé très fréquemment dans tous les documents du corpus par rapport aux termes plus spécifiques à certains documents. Une mesure possible est la suivante (Jones 1973).

$$tfidf = \frac{occurence\ du\ mot\ dans\ le\ document}{nombre\ de\ mots\ du\ document} * \log\left(\frac{nombre\ de\ documents\ dans\ le\ corpus}{nombre\ de\ documents\ dans\ lequel\ le\ terme\ apparait}\right)$$

Quand le nombre de documents est important (20 à 100 000), le nombre de termes l'est aussi (de l'ordre de

15 000) ce qui génère un tableau de grande taille, pouvant atteindre le giga octet, caractérisé par la rareté des éléments présents, en anglais *sparcity*.

Généralement on transformera ce tableau en un tableau de cooccurrences ou de corrélations qui se prête à différents types de représentation des similarités (voir section “Découvrir le sens du corpus”).

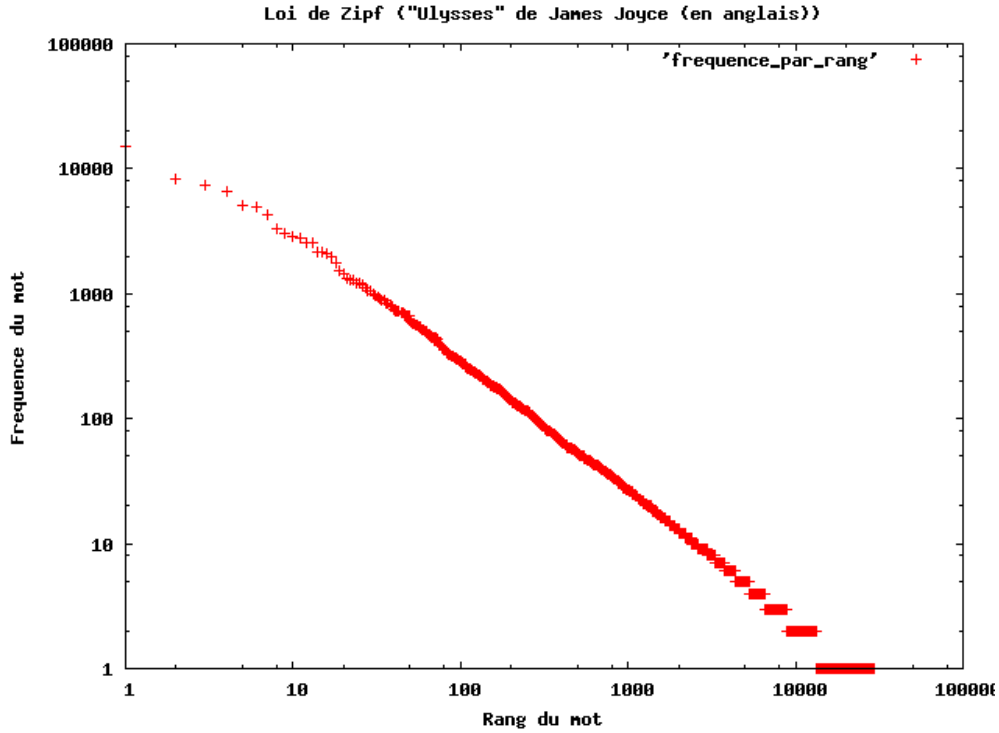
Cette transformation permet de produire des premiers éléments d’analyses, au premier titre duquel apparaissent les nuages de mots (*wordcloud*) qui représentent la fréquence relative des termes et qui donnent une première idée des significations contenues dans le corpus. On pourra produire des nuages de mots comparatifs, mais on se méfiera des proximités qui résulte d’un calcul de pavage (remplir l’espace avec des objets de tailles variées).

Exemple 4 : Nuage de mots

Les nuages de mots ont un intérêt rhétorique et se prêtent à la créativité esthétique, ils sont le degré zéro de la statistique textuelle. On leur reconnaîtra une valeur heuristique et herméneutique.

Des représentations plus classiques ordonnent les mots par fréquence et par rang de fréquence, révélant la structure profonde de la distribution des termes, celle identifiée par Zipf (B. et Zipf 1951) qui suppose que la fréquence de n’importe quel mot est inversement proportionnel à son rang dans la table de fréquence. Les mots classés par ordre de fréquence, suivent une distribution puissance : f(p)=1/r(p).

Exemple 5 : Une distribution de Zipf
Graphique log/log de la fréquence des mots par leur rang dans "Ulysse" de James Joyce (CC BY-SA 4.0, https://commons.wikimedia.org/w/index.php?curid=7982062)



2. Travailler le corpus pré-traité

a. Les indicateurs de qualité

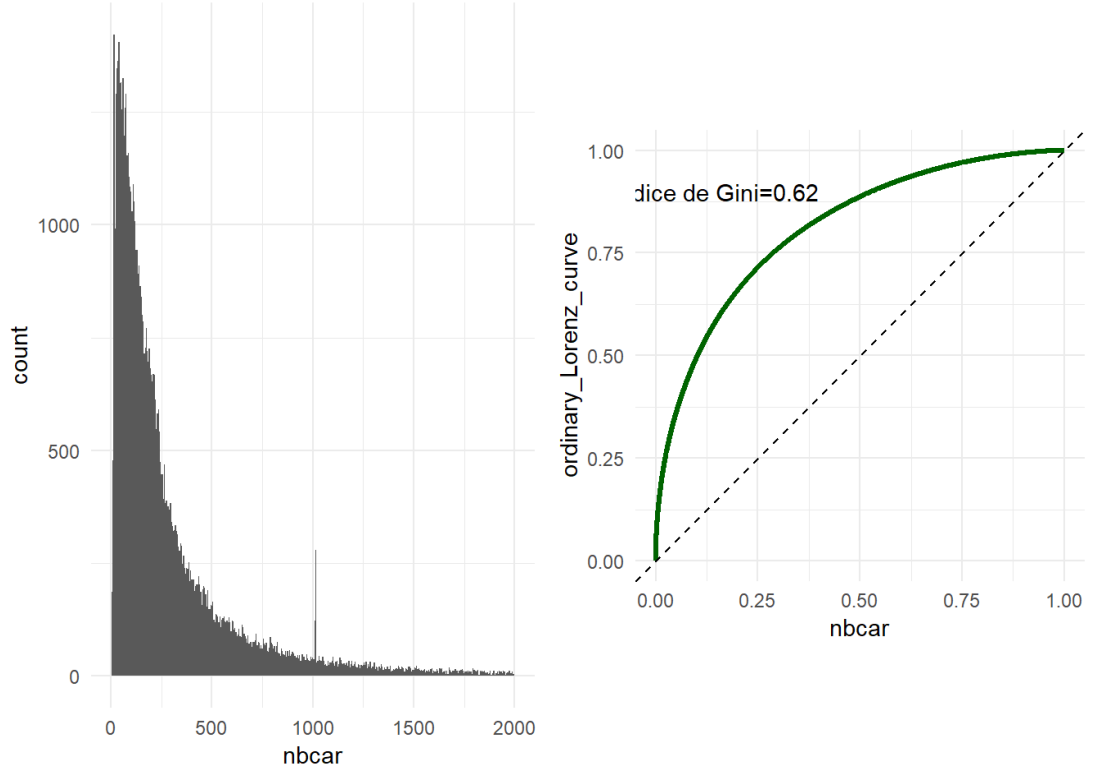
En préalable à toute analyse, il est utile de décrire le corpus de manière quantitative et d'en apprécier les dimensions :

- Nombre de textes ;
- Nombre de tokens, total et moyen par texte ;
- Taille des textes et concentration en nombre total et moyen de caractères, de syllabes, de mots et de phrase.

Ces informations peuvent suffire dans des tâches de comparaison : les femmes ont-elles une écriture différente de celle des hommes ? Les contenus négatifs et positifs se distinguent-ils sur ces critères ?

Exemple 6 : Distribution cumulée de la longueur de textes et concentration du texte

Les données sont celles des contributions au Grand Débat organisé à l'hiver 2019 relatives au thème de l'organisation de l'État et des services publics. La distribution en nombre de caractères montre une très forte asymétrie : la proportion de texte de plus de 500 caractères est faible. On remarque un pic qui correspond à des "fakes" : des textes stéréotypés et copiés-collés. À droite, on représente la distribution cumulée (diagramme de Lorenz) : les 25 % de contributions les plus longues représentent près de 75% du volume de texte.



D'autres indicateurs plus riches peuvent aussi être calculés :

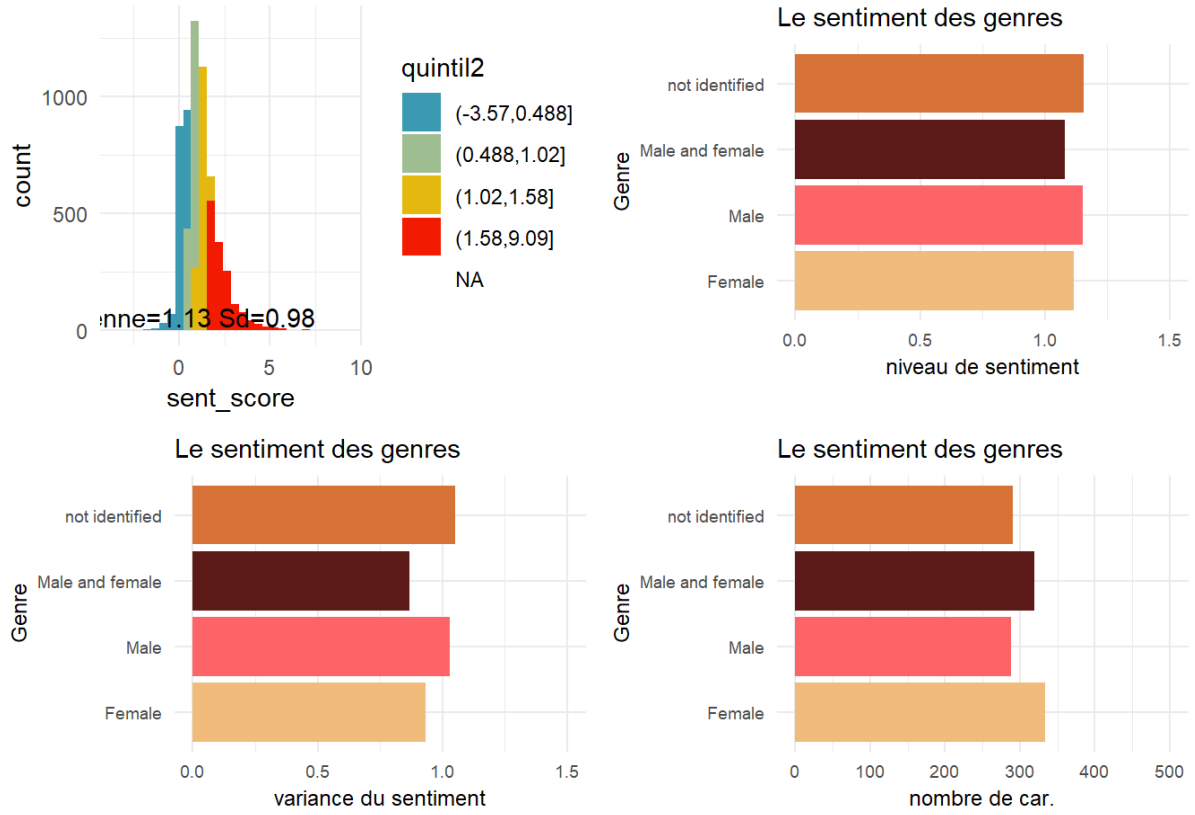
- Une mesure de lisibilité. Ces indicateurs ont été développés dans les années 60 (Senter 1967) pour standardiser le niveau de lecture (en termes de niveaux de classe d'apprentissage). Ils s'appuient sur une combinaison de nombre de mots par phrase et de syllabe par mots pour quantifier la facilité de lecture.
- La diversité lexicale, qui reflète la richesse du langage.
- Le nombre de mots uniques, les hapax.

Un outil propose une liste extensive : le LIWC créé par (Tausczik et Pennebaker 2010) dont 'quanteda' reprend des éléments. C'est un ensemble de dictionnaires qui permettent de distinguer près de 80 traits, allant de la mesure quantitative des textes à celle de catégorie topicales et affective. Récemment (van Laer et al. 2018) en font la démonstration du plein intérêt dans le champ des commentaires de consommateurs.

Une autre très belle utilisation de ces indicateurs est celle de (Šuster 2015) qui tente de répondre à la question de savoir si les gamers utilisent un langage plus simple que l'ordinaire des gens. En observant que le jeu vidéo massivement parallèle exige des capacités de coordination élevées, il analyse le contenu des échanges et montre que sur différents critères, ils sont aussi sophistiqués que la langue standard rejetant l'hypothèse d'une dégradation de la qualité du langage induite par l'usage des technologies de l'information.

Exemple 7 : Comparaison du langage entre des commentaires des hommes et des femmes sur Airbnb

On compare la production langagière des hommes et des femmes (identifiés par l'analyse du prénom donc de genre féminin, masculin, mixte ou non identifié), dans un corpus de commentaires Airbnb. On s'aperçoit de différences minimes, femmes et hommes parlent le même langage)



b. L'annotation du langage naturel

C'est une étape essentielle qui traduit une découverte importante des sciences des données. Les données brutes se prêtent rarement aux modèles, ces derniers sont d'autant plus efficaces que les données ont été filtrées et transformées de manière telle à ce qu'elles correspondent à leur structure. Si les n-grammes et les simplifications de bases ne sont pas suffisantes, il va être nécessaire de mieux caractériser les éléments du texte en apportant d'autres informations, lexicales et syntaxiques. C'est le rôle de l'annotation.

Exemple 8 : Annotation
Annotation de la phrase "je lis cet article"

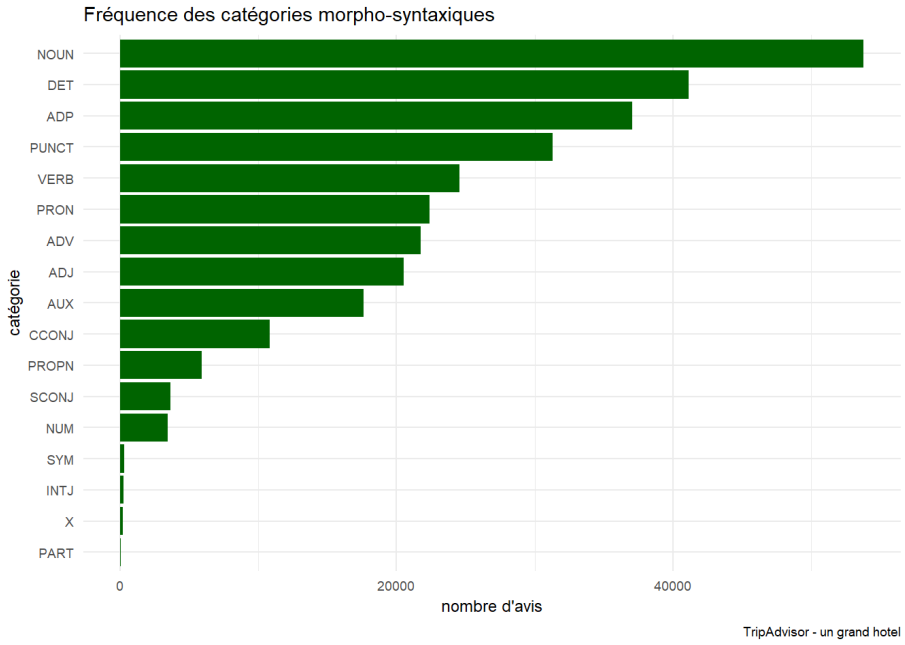
Phrase tokenisée	" Je"	"lis"	"cet"	"article"
Annotations				
Lemme	je	lire	ce	article
Position	1	2	3	4
Part Of Speech	Pronom	Verbe	Déterminant	Nom commun

Quand les tokens sont des mots, au moins 6 types d'annotations sont couramment employées, elles font souvent l'objet de normes internationales dans la définition de leurs modalités :

- La lemmatisation consiste à retrouver pour chaque flexion le terme primaire ; on distingue la stemmatisation qui consiste à prendre les premières lettres d'un mot pour trouver la racine mais qui échoue dans le traitement des affixes.
- Les Parts *Of Speech* consiste à identifier les éléments du langage et donc à attribuer des formes morphosyntaxiques aux mots en fonction de leur environnement : "rose" est-il la fleur, le prénom ou l'adjectif de couleur ?
- Annoter les entités nommées consiste à caractériser les noms communs par leurs catégories génériques : des lieux, des personnages, des marques, des institutions, des institutions.
- Le sentiment du terme s'il est disponible dans un dictionnaire des sentiments, la valence est le critère le plus fréquent, mais on peut aussi annoter l'émotion.
- Les dépendances syntaxiques établissent les relations syntaxiques entre les mots quelle que soit leur position dans la phrase. Le Stanford NLP en distinguent 57. Une solution étendue est le package 'CleanNLP'(Arnold 2017b). Ces méthodes s'appuient sur des méthodes d'IA et des corpus tels que le *Penn Treebank (Taylor, Marcus, et Santorini 2003)*.
- Les coréférences qui consistent à trouver les différents termes relatifs à un même objet (par exemple dans la phrase : « le manager encourage ses équipes, il est leur soutien plus que leurs chefs » , « il » et « leur » sont les coréférences de « manager »)
- Mais on peut aussi aller plus loin avec l'identification des synonymes les plus courants, leurs antonymes (termes opposés : succès-échec), les hyperonymes et hyponymes (relation du tout à la partie : entreprise-marketing), les troponymes (manières : chuchoter, parler) introduite par WordNet (Miller 1995)

Pour traiter ces structures complexes, les méthodes du deep-learning offrent de belles perspectives. À partir de ces annotations on peut imaginer fournir les éléments pour qu'un outil de machine-learning prédise un certain caractère du texte : le degré de sarcasme, la toxicité, le registre de langage (argotique, bureaucratique…), un certain sentiment de justice. Ce sera l'objet de la dernière partie du chapitre.

Exemple 9 : Analyse morphosyntaxique d'un échantillon de commentaires Trip Advisor.
La figure indique la distribution des formes grammaticale selon la classification UPOS.



Exemple 10 : Arbre syntaxique et ses annotations

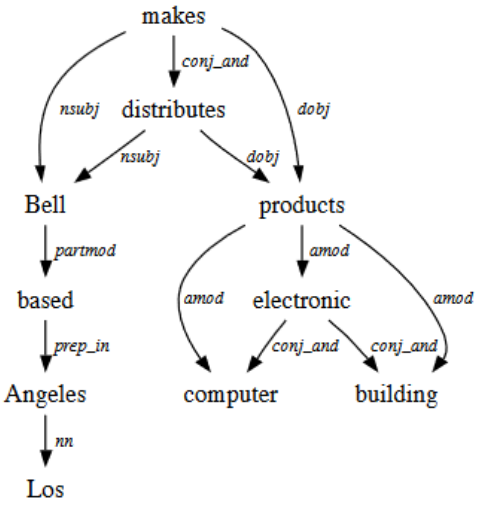


Figure 1: Graphical representation of the Stanford Dependencies for the sentence: *Bell, based in Los Angeles, makes and distributes electronic, computer and building products.*

Source : Stanford NLP

c. La détection des sentiments

L'analyse du sentiment s'est bien installée dans le monde pratique mais aussi peu à peu dans celui de la recherche : en marketing, pour la gestion de la relation client et de la réputation des marques, mais aussi en finance, pour prédire les cours de bourse par exemple.

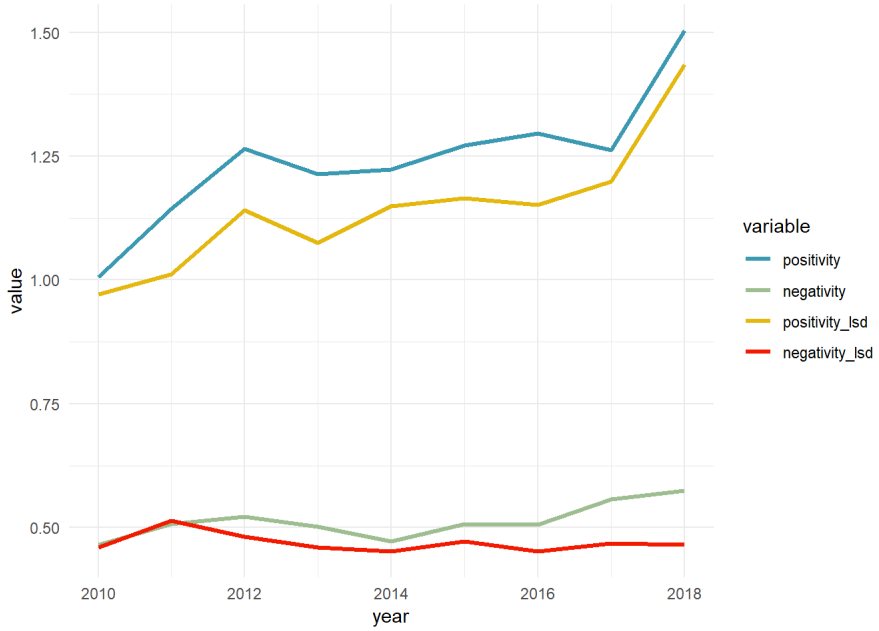
L'objectif est de mesurer, au niveau d'un mot, d'une phrase ou d'un document; la polarité du sentiment : l'idée proposée est-elle négative ou positive ?

Deux grands type d'approches se sont développée, la première s'appuie sur des dictionnaires, la seconde sur les techniques de machine learning, sur lesquelles nous reviendrons à la fin de ce chapitre.

L'approche par dictionnaires permet au chercheur d'utiliser des lexiques de sentiment pour évaluer la tonalité du texte étudié. Ces lexiques (ou dictionnaires) listent des mots et le sentiment qui leur est rattaché. La création du lexique est un enjeu essentiel pour le chercheur. C'est l'approche retenue par le National Research Council of Canada pour constituer grâce à Amazon Mechanical Turk un dictionnaire des sentiments, traduit dans plus de 140 langues, le lexique NRC(Mohammad et Turney 2013)adapté en français avec l'outil FEEL de (Abdaoui et al. 2017) Différentes approches des sentiments co-existent, qui se traduisent par différents types de lexique : le lexique AFINN (Nielsen 2011) propose une échelle de -5 à +5, le lexique BING(Ding, Liu, et Yu 2008) utilise un codage binaire positif/négatif, le lexique NRC associe pour chaque mot un sentiment positif ou négatif et/ou une émotion, le lexique du LIWC utilise des listes de mots associés à des états émotionnels, d'autres dictionnaires sont disponibles comme le Lexicoder (Duval et Pétry 2016)

La problématique actuelle repose sur la disponibilité des outils en langue française. Le dictionnaire NRC est accessible, mais suite à un processus de traduction automatique, qui ne tient pas compte des spécificités liées aux langues et aux cultures (le terme communiste n'a pas la même valeur négative aux États-Unis ou en France).

Exemple 11 : Comparaison de deux dictionnaires de sentiment
Convergence de deux dictionnaires de sentiment sur le corpus d'un hôtel de Polynésie (n= 2800 sur 10 ans) FEEL et frLSD. Les deux indicateurs de positivité co-varient dans le temps (r=0,53 au niveau des avis), ceux de négativité sont moins associés même s'ils ne divergent pas excessivement,(=0,50).



Cette approche par les dictionnaires, un peu sèche (un mot pour un mot, sans contexte), s'enrichit aujourd'hui par la prise en compte des "modificateurs" : négation, double négations, amplificateurs…

La force de cette approche est la simplicité d'application, quand sa faiblesse est de ne pas tenir compte des spécificités du corpus. Prenons par exemple le terme "félicité" qui dans le sens commun décrit un état de grâce d'un bonheur qu'aucune joie n'encombre, mais qui en philosophie du langage désigne les conditions pour qu'un énoncé soit performatif. Dans des recherches importantes (en moyens) il est nécessaire de compléter ces dictionnaires par des termes spécifiques au domaine ou au contexte d'études. Des dictionnaires ad hoc peuvent (doivent ?) être créés par les chercheurs.

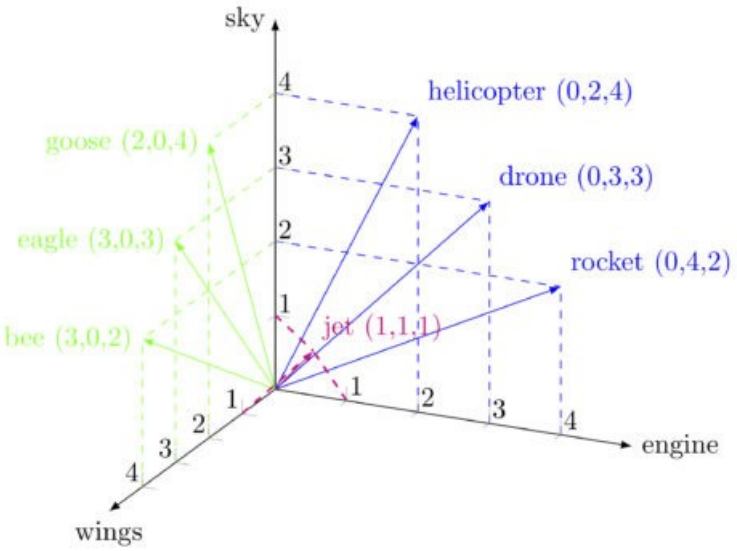
d. La vectorisation des termes (word embedness)

L'hypothèse de distribution en linguistique est dérivée de la théorie sémantique de l'usage de la langue, c'est-à-dire que les mots utilisés et utilisés dans les mêmes contextes ont tendance à avoir des significations similaires. Le *word embedding* repose sur une découverte : la régularité de certains rapports qui permettent de jouer le sens par un calcul vectoriel.

Le monde du machine learning apporte une innovation remarquable : la vectorisation. Le principe est simple. À partir du tdm, on peut aisément construire des tables de corrélations entre les termes. L'idée est de représenter ces corrélations le plus précisément possible pour garder toute l'information dans un espace de dimension arbitrairement grande. L'idée est moins de réduire que de conserver l'information et le choix est de représenter chaque terme dans un espace de 100 à 1000 dimensions.

L'astuce est le sac de mot (*bag of words*). Un petit ou un grand ? Considère-t-on que deux termes sont similaires s'ils font partie d'un paquet de 5 mots ou de 40 mots ? C'est bien plus qu'une astuce, cela devient le paramètre de l'analyse, favoriser de petits sacs, c'est chercher les termes qui sont directement associés, de plus grands sacs permettent de mieux prendre en compte le contexte des termes. Le paramètre de la taille du sac traduit la stratégie de recherche : les compléments ou les contextes.

Exemple 12 : La sémantique comme calcul vectoriel
On représente ici un micro corpus composé de 7 mots (bee, eagle, goose, helicopter, drone, rocket, and jet) et de trois contextes(wings,engine, and sky). Chaque mot est caractérisé par trois coordonnées qui correspondent au nombre de fois où le mot apparaît dans chacun des contextes. Par exemple, *helicopter* n'apparaît pas dans le contexte *wings*, mais apparaît respectivement 2 et 4 fois dans les contextes *engine* et *sky*. Ses coordonnées sont par conséquent (0,2,4). Chaque mot occupe une position spécifique dans l'espace vectoriel, comme représenté sur la figure suivante(Guillaume Desagulier 2018)



Pour cette tâche, d'un point de vue technique au moins deux approches sont disponibles, notamment sur r :

- *Word2Vec* proposé par(Mikolov et al. 2013)miko Mikolov (2013)[2] de Google. Il propose d'apprendre la structure en résolvant simultanément deux problèmes : prédire à partir d'un environnement (une série de mots) celui qui leur est le plus associé, et réciproquement pour un mot de prédire ceux qui constituent leur environnement. Un réseau de neurones à une couche cachée est employé (avec K, le nombre de dimension, inférieur à N, le nombre de termes).
- *Glove(Pennington, Socher, et Manning 2014)* qui présente l'avantage de construire deux vecteurs : un spécifique et un de contexte. Il vient du MIT, et n'a pas recours à une méthode de réseaux de neurones.

Une innovation importante dans le traitement des données textuelles est sans doute celle de Word2Vec. L'idée est finalement simple et tient sur deux éléments essentiels :

- Le premier est que la similarité entre deux mots dans un corpus peut non seulement être mesurée par une corrélation, mais surtout être calculée non sur l'ensemble du corpus des cooccurrences, mais sur des fenêtres glissantes de texte, de l'ordre de 5 à 10 mots consécutifs. Les termes seront similaires parce qu'ils apparaissent dans les mêmes séquences.
- Le second est que l'on peut représenter les termes par des vecteurs dont les angles (leur cosinus) correspondent aux corrélations. L'idée clé est de représenter cette matrice de corrélations dans un espace vectoriel de grande dimension (de 50 à 1000). À l'inverse de l'ACP qui cherche une représentation vectorielle réduite, on cherche une représentation vectorielle étendue pour conserver toute l'information.

De manière métaphorique un modèle Word2Vec peut être imaginé comme un oursin, chaque épine est un vecteur (et donc un mot), les épines proches sont celles qui marchent ensemble.

La méthode Word2Vec est, on le comprendra, gourmande en calcul, mais ce n'est pas un grand problème sur des corpus de dizaines de milliers de termes sur les machines actuelles.

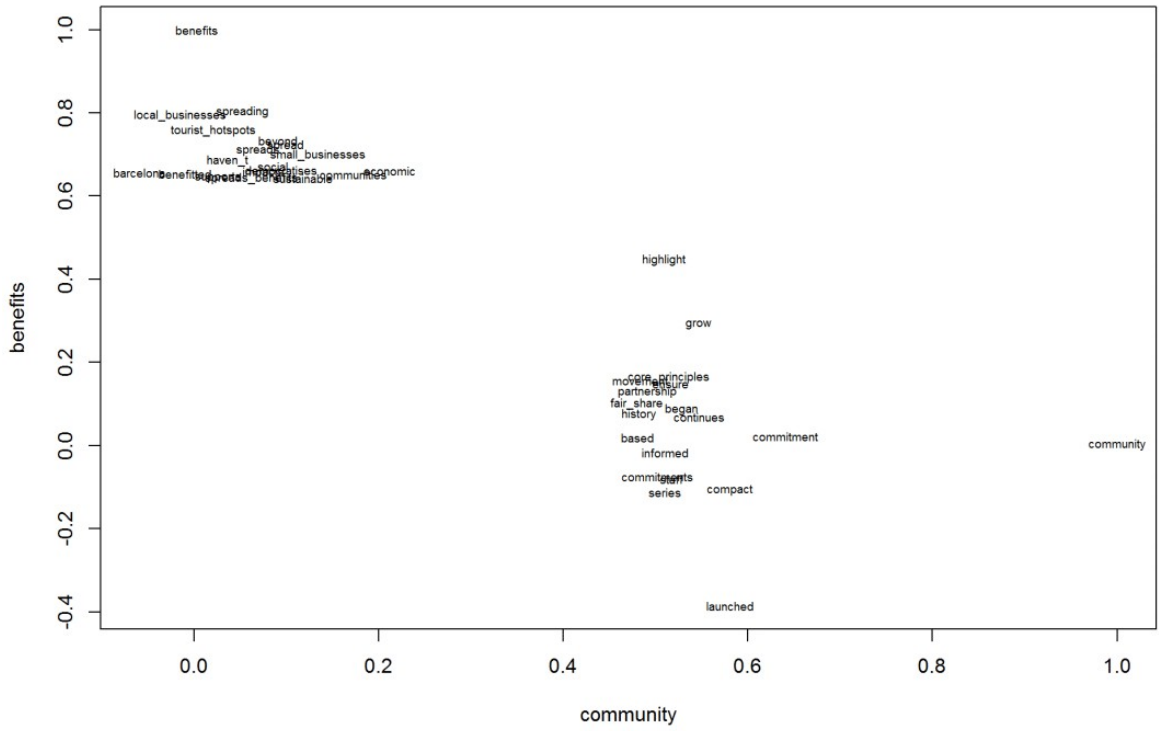
2

Exemple 13 : Analyse de concept

La vectorisation ne demande avec 'text2vec' que deux lignes de commande. On analyse ici un petit corpus des textes de communication institutionnelle de Airbnb en 2016.

```
prep_word2vec(origin="airbnbcitizen",destination="airbnbcitizen.txt",lowercase=T,bundle_ngrams=2)
train_word2vec("airbnbcitizen.txt","airbnbcitizen_vectors.bin",vectors=200,threads=4,window=12,
iter=500,negative_samples=0)
```

On représente ici les termes les plus corrélés à deux termes cibles : *community* et *benefits* qui sont les plus fréquents dans le corpus, marquant le but du discours : les avantages de la communauté. La solidité de l'engagement caractérisant cette dernière, les avantages se traduisant par un impact diffusé dans l'économie locale.



3. DÉCOUVRIR LE SENS DU CORPUS

Une fois que les mots importants, redondants ou qui posent question ont été identifiés, il est utile de pouvoir les appréhender dans leur contexte naturel, pour bien saisir le sens dont ils sont porteurs. La fonction *kwic* (*key word in context*) de 'quanteda' permet de représenter un mot-clé dans son contexte, c'est-à-dire au sein des phrases où il apparaît, en laissant le choix quant à la fenêtre de mots à présenter avant et après le terme cible.

Cette manière d'interroger le texte est essentielle au chercheur pour donner tout son sens aux découvertes de régularités présentes dans le corpus.

Exemple 14 : *Key word in context*
Extrait des contextes d'occurrence du terme "minutes" dans le discours de la société Allocab lors du conflit opposants les VTC et les taxis.

docname	from	to	pre	keyword	post
text2	6	6	Taxi-VTC : Décret des 15	minutes	suspendu suite au recours d'AlloCab.com
text2	59	59	. Le décret des 15	minutes	imposait aux véhicules de tourisme
text2	71	71	chauffeur un délai de 15	minutes	entre la réservation et la
text3	7	7	12 août 2013 Les 15	minutes	de la discorde entre VTC
text3	118	118	pratique le délai de 15	minutes	? Prenons le scénario suivant
text3	129	129	intégrant ce délai de 15	minutes	: 1 . Un client
text3	158	158	de la course en 7	minutes	, 3 . Le client
text3	167	167	Le client doit attendre 8	minutes	avant de monter dans le
text3	183	183	7 + 8 = 15	minutes	« légales » . .
text3	194	194	. 4 . Les 8	minutes	s'écoulent , le client monte
text3	275	275	contrôler le respect des 15	minutes	requises rend logiquement la sanction

1. Cooccurrences et similarités

L'opération la plus simple pour analyser les données textuelles consiste à identifier pour un mot clé ou mot cible ceux qui lui sont le plus fréquemment associés. On pourra utiliser les corrélations ou les cooccurrences et ordonner les termes selon la métrique choisie, la différence entre ces deux mesures étant de tenir compte ou non de la possibilité pour un terme d'apparaître sans l'autre.

Exemple 15 : Du tdm aux cooccurrences

Corpus	Documents	Tokens
1	ceci est une rose rouge	"être", "rose"
2	le jardin est fleuri de roses	"être", "jardin", "fleur", "rose"
3	un bouquet de fleur pour sa mère	"bouquet", fleur", "mère"
4	des ombres dans le jardin	ombre, jardin

dtm	être	rose	jardin	fleur	bouquet	mère	ombre
1	1	1					
2		1	1	1			
3				1	1	1	
4				1			1

Co-occurences	être	rose	jardin	fleur	bouquet	mère	ombre
être		1					
rose		2	1				
jardin		1	1	1			
fleur		1	1	1	1		
bouquet					1	1	
mère		1			1	1	1
ombre				1			1

Il existe aussi d'autres mesures de similarités, comme la *keyness*, qui consiste à comparer les apparitions des termes similaires aux apparitions des mots-cibles. Pour cela, après avoir délimité une fenêtre d'analyse (de 5 à 40 mots en pratique), on va comparer l'apparition des termes présents dans la fenêtre avec leur apparition dans le corpus auquel on a soustrait le mot-cible et les termes l'entourant. Des mots présents de manière quasi-

systématique avec le mot-cible et peu présent sans que le mot-cible soit présent nous informe sur le sens du discours.

Exemple 16 : *Keyness*

Mesure de similarité du terme “minutes” extrait du corpus d’Allocab (cf. exemple 14). La similarité entre les mots est mesurée par un test de chi-deux. Les résultats montrent aussi les occurrences des termes dans le corpus cible (entourant le mot-cible) et celles dans le corpus de référence (excluant le corpus entourant le mot-cible).

feature	chi2	p	n_target	n_reference
minutes	172.118398	0.0000000000	19	0
courses	13.425079	0.0002482825	4	3
sens	10.012298	0.0015549841	2	0
respect	10.012298	0.0015549841	2	0
impossible	10.012298	0.0015549841	2	0
dont	10.012298	0.0015549841	2	0
journée	10.012298	0.0015549841	2	0
chaque	10.012298	0.0015549841	2	0
pourquoi	10.012298	0.0015549841	2	0
chauffeur	9.373637	0.0022012829	6	12
client	9.373637	0.0022012829	6	12

2. L’analyse de graphe

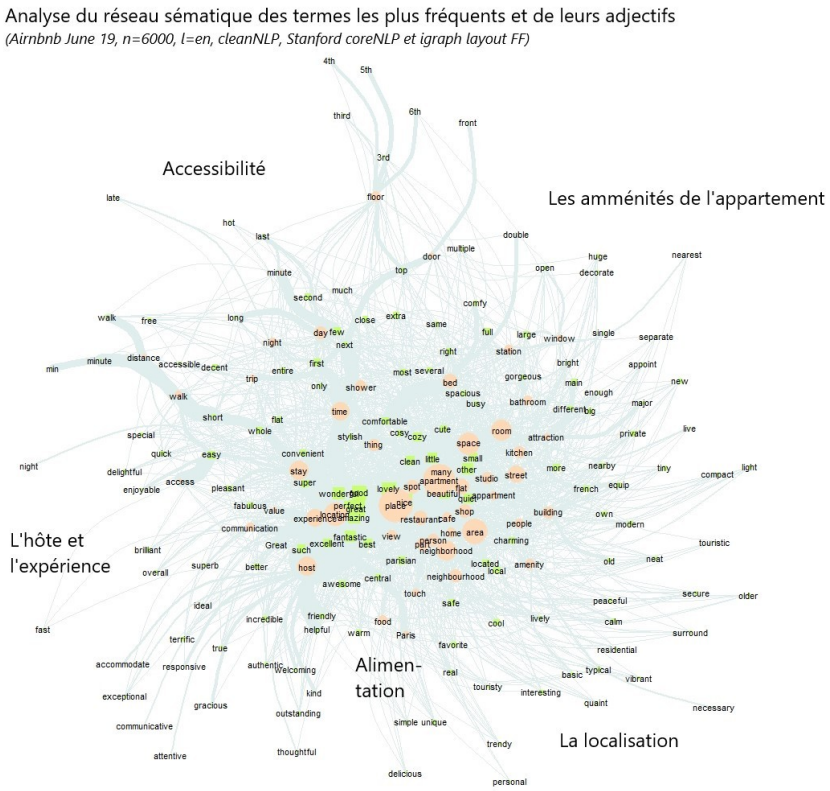
À partir des corrélations ou des collocations des termes, il est possible de représenter les relations entre les termes sous la forme d’un graphe. Si nous prenons un mot cible, et que nous obtenons la liste et les valeurs des 30 mots qui lui sont le plus corrélés, on peut répéter l’opération pour chacun des mots associés et construire alors leur table de réseau.

Pour la représentation dans le plan on emploie souvent la méthode de (Fruchterman et Reingold 1991), un algorithme de force qui va calculer la distance entre les nœuds par des mécanismes d’attraction/répulsion suivant la force des liens et le poids des nœuds. Naturellement, d’autres solutions d’analyses des similarités sont possibles, notamment via une analyse en composante principale (ACP).

Le choix des liens est alors juste une question de seuil : quel niveau de corrélation choisit-on de représenter ?

Exemple 17 : Un réseau sémantique

Ce réseau sémantique représente la relation entre les adjectifs et les termes qu'ils qualifient. C'est un réseau bipartite qui associe deux types d'objets.



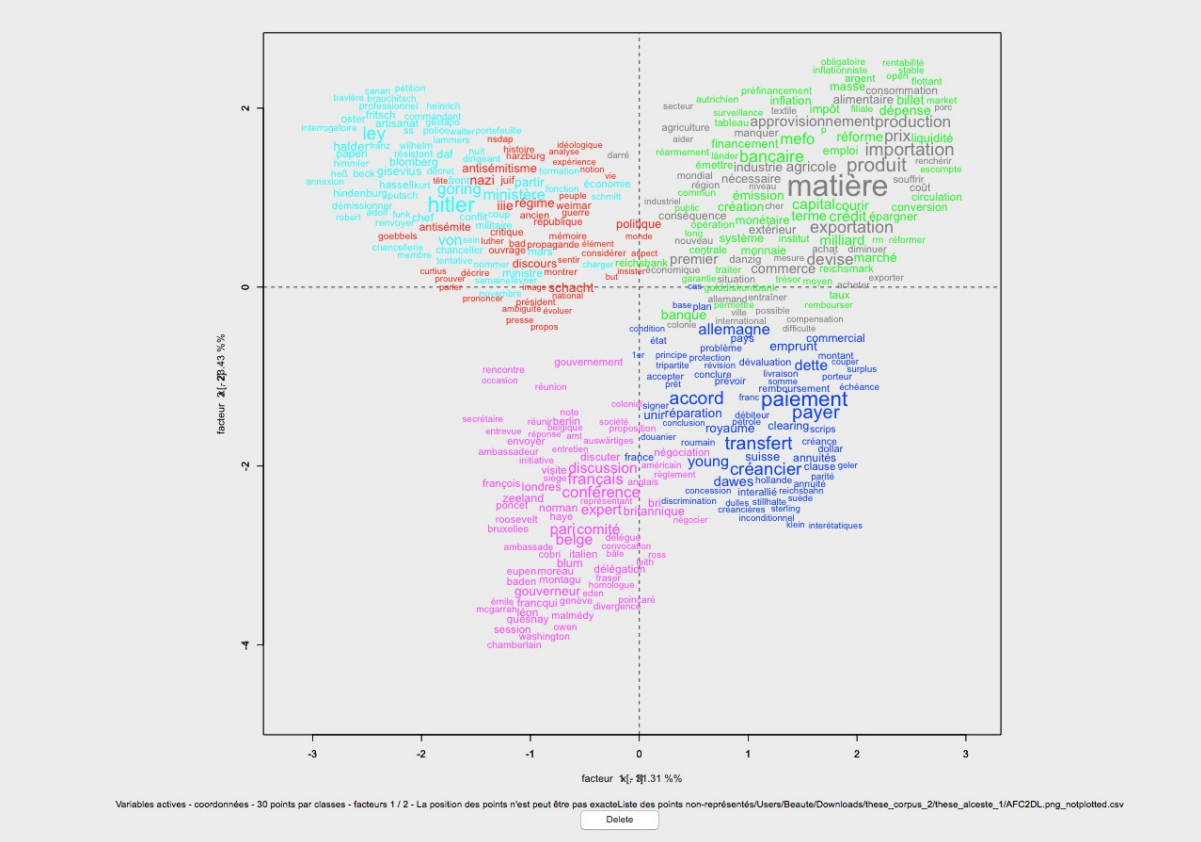
3. Les ancêtres : l'analyse factorielle des correspondances et le *clustering*

La matrice termes-documents se prête bien aux méthodes classiques d'analyse des données, que ce soit l'analyse factorielle multiple des correspondances (AFCM) ou la classification, hiérarchique ou non.

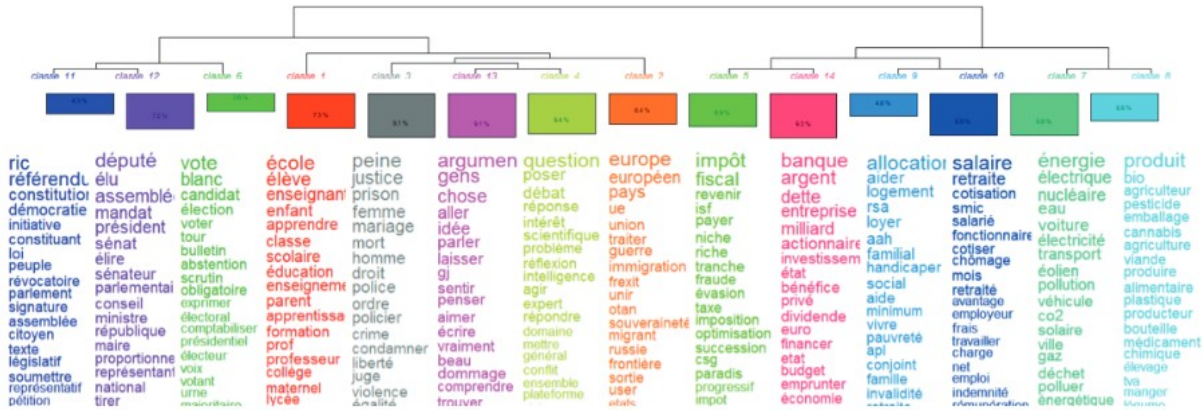
L'analyse factorielle des correspondances est une longue tradition française d'analyse des données, grâce notamment aux travaux de Benzecri (1973) ou de Bouroche (1977) dans les années 1960. La technique originelle de l'AFCM appliquée aux données textuelles a connu de nombreuses implémentations, dont le package 'themis' sur r, les solutions intégrées d'Alceste, et maintenant l'interface Iramuteq qui exploite pleinement les ressources de r/python, un bel exemple d'application est donné par (Ratinaud et Marchand 2015)ratin. Le principe est le même que pour le traitement plus classique de questionnaires à variables qualitatives.

Les méthodes de classification hiérarchique ou non hiérarchique s'appliquent également très bien aux données textuelles. Nous renvoyons les lecteurs désireux d'en savoir plus aux manuels d'analyse de données pour la gestion.

Exemple 18 : Résultats d'une AFCM
Utilisation de l'AFCM via Iramuteq pour l'analyse du Grand Débat par l'équipe du LERASS de Toulouse.



Exemple 19 : Résultat de la classification descendante hiérarchique
Classification effectuée sur le corpus du Vrai Débat, plateforme participative alternative au Grand Débat National, analyse effectuée par l'équipe du LERASS à Toulouse à l'aide d'Iramuteq.



4. Analyse des similarités avec t-SNE

Les méthodes précédentes permettent d'explorer et d'identifier les thématiques principales qui traversent les documents. Mais elles laissent place à des méthodes plus adaptées à des espaces de grande dimension.

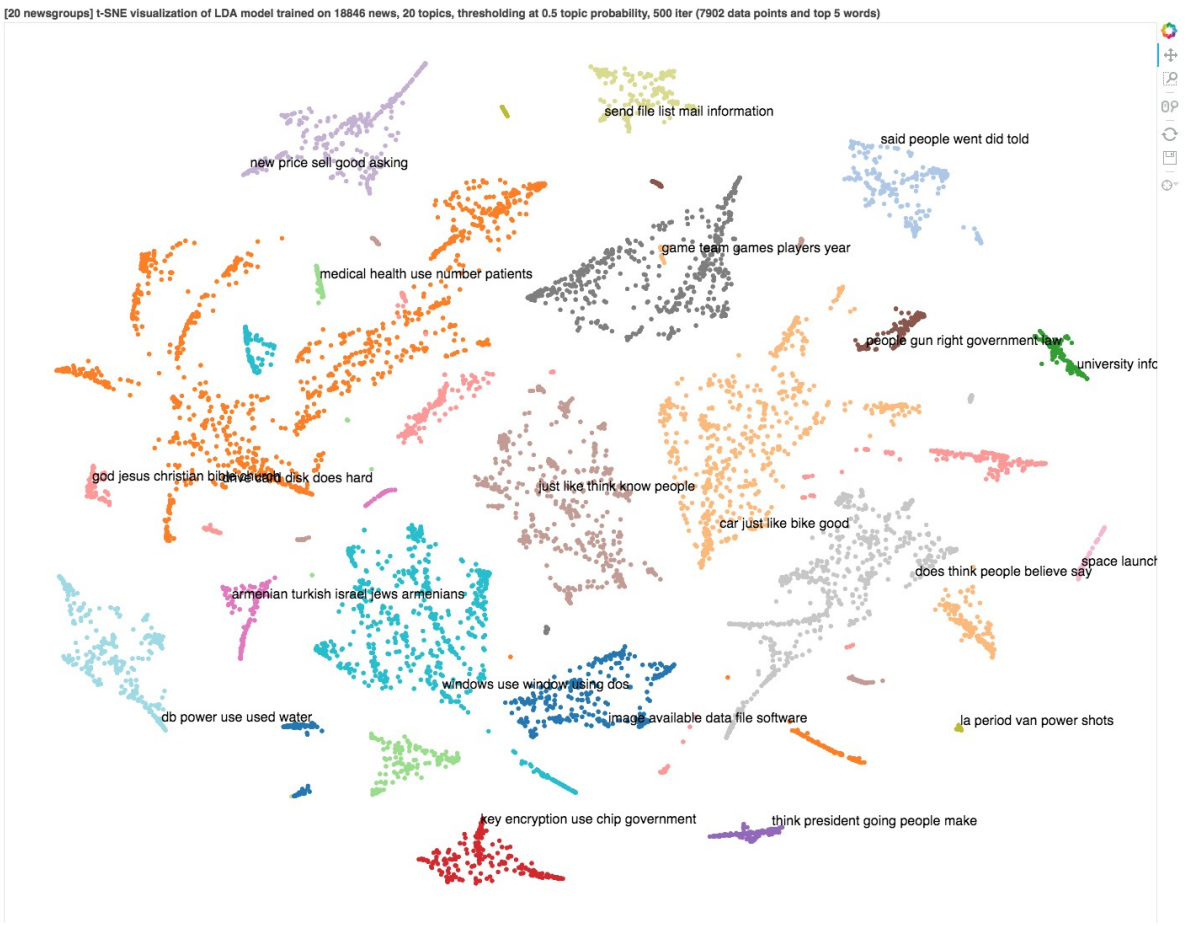
L'algorithme t-SNE, pour *t-Distributed Stochastic Neighbor Embedding*, permet de représenter les similarités présentes dans un corpus, par nature de grande dimension, dans un plan à deux dimensions seulement. Par rapport aux méthodes classiques de réduction des données, cette approche est non-linéaire et repose sur la

transformation de la distance euclidienne entre les points en probabilités conditionnelles, la similarité. Elle a été développée pour gérer des jeux de données de très grande dimension. En pratique, le modèle cherche à garder les points proches, le plus proche possible et à accentuer l'éloignement entre des points très éloignés.

La représentation en deux dimensions se prêtent facilement à l'interprétation, une fois le modèle ajusté. L'ajustement se fait par la détermination par le chercheur d'un paramètre crucial : la perplexité, dont la valeur varie en général de 5 à 50. Ce paramètre peut être vu comme un variateur qui définit le nombre de voisins effectifs à prendre en compte dans les calculs. La définition de la bonne perplexité passe par une méthode d'essai-erreur, jusqu'à obtenir une représentation visuelle interprétable et qui a du sens pour le chercheur.

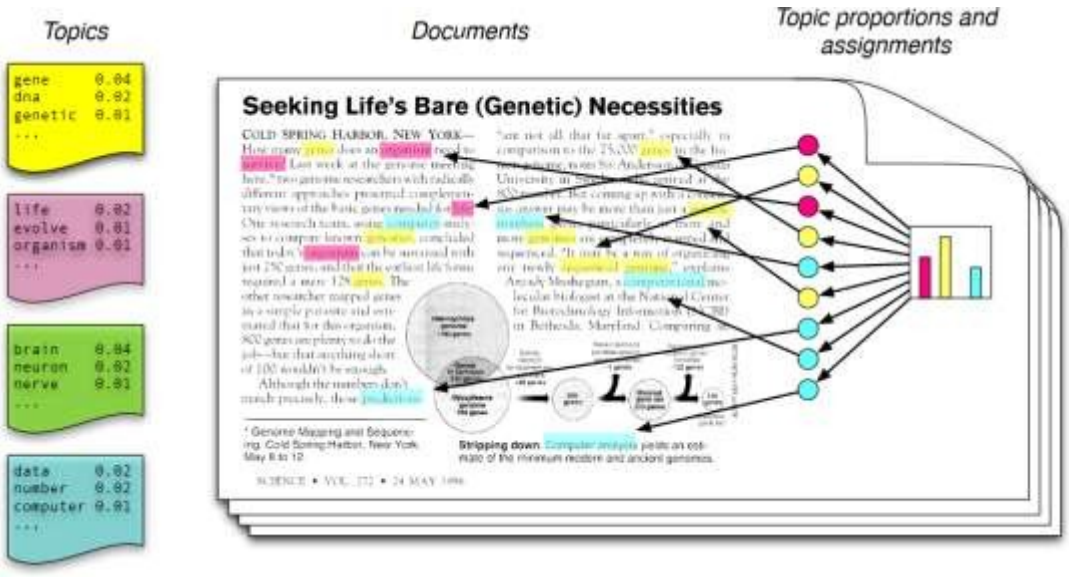
Exemple 20 : Résultats de la procédure t-SNE

https://www.mathworks.com/help/textanalytics/ug/visualize-word-embedding-using-text-scatter-plot.html



5. L'identification des *topics* latents par la méthode LDA

En 2003, Blei et ses collègues (Blei, Ng, et Jordan 2003) introduisent un modèle probabiliste, *Latent Dirichlet Allocation* (LDA) qui décrit k sujets (*topics*) définis par la probabilité que les termes appartiennent à chacun des k sujets, et par les probabilités que chaque document soit relatif aux sujets. C'est une forme d'analyse factorielle latente. Le modèle s'appuie sur une distribution de Dirichlet naturellement adaptée aux événements multinomiaux de paramètre alpha.



Cette méthode est algorithmique : on cherche à améliorer le résultat (*topic model*) généré aléatoirement en initialisation. Pour cela, dans chaque document, on prend chaque mot et on met à jour le thème auquel il est lié. Ce nouveau thème est celui qui aurait la plus forte probabilité de le générer dans ce document. On fait donc l'hypothèse que tous les thèmes sont corrects, sauf pour le mot en question.

Plus précisément : pour chaque mot (w) de chaque document (d), on calcule deux choses pour chaque thème (t) :

p(thème t | document d): la probabilité que le document d soit assigné au thème t

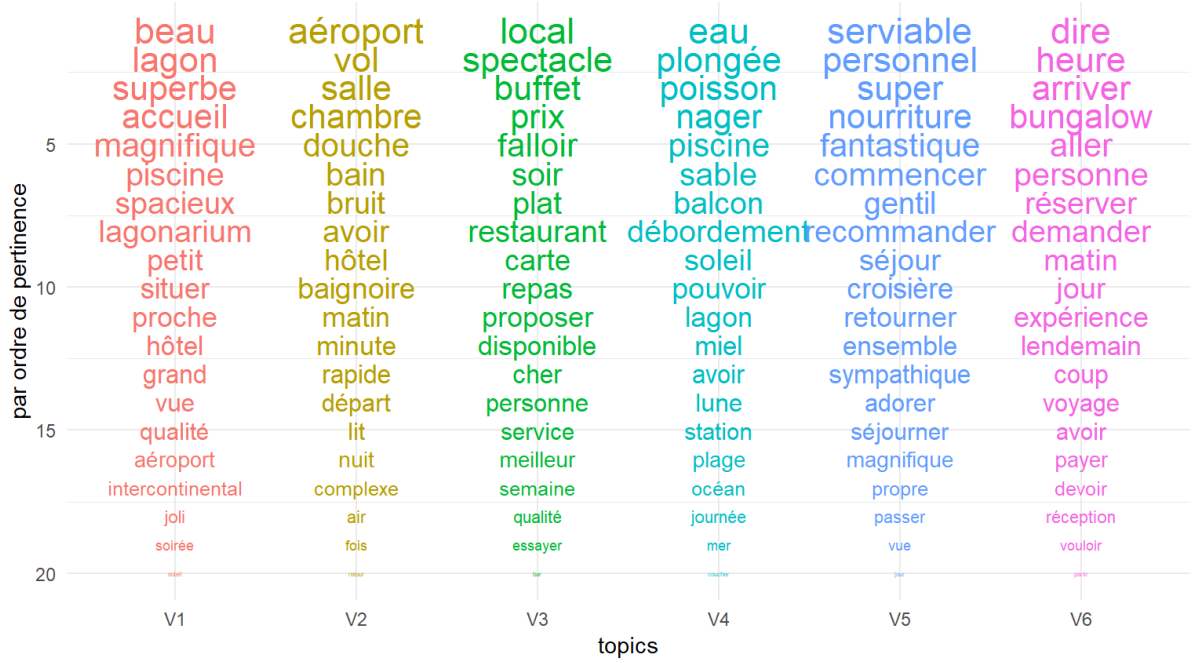
p(mot w | thème t): la probabilité que le thème t dans le corpus soit assigné au mot w

On choisit alors le nouveau thème t avec la probabilité p(thème t | document d) * p(mot w | thème t). Ceci correspond à la probabilité que le thème t génère le mot w dans le document d.

En répétant les étapes précédentes un grand nombre de fois, les assignations se stabilisent. On obtient le mélange de thème présent dans chaque document en comptant chaque représentation d'un thème (assigné aux mots du document). On obtient les mots associés à chaque thème en comptant les mots qui y sont associés dans le corpus.

Les résultats du modèle dépendent d'un paramètre fixé à l'avance par le chercheur : le nombre de *topics* latents présents dans le corpus. La question est donc de savoir quel est ce nombre optimal pour décrire au mieux le corpus. Des indicateurs existent à calculer sur chaque itération du modèle avec un nombre k de *topics* différent, et dont la comparaison permet d'estimer le nombre adéquat de *topics* à retenir pour décrire le corpus. Cette procédure est très gourmande en temps de calcul, mais représente un passage nécessaire pour le chercheur en quête de sens.

Exemple 21 : Résultats du modèle LDA
Application à l'échantillon des commentaires relatifs à un grand hôtel du pacifique.



4. ALLER PLUS LOIN AVEC LE *MACHINE LEARNING*

Jusqu'à présent nous nous sommes largement appuyé sur les propriétés distributionnelles du langage (dont la philosophie date de (Firth 1957) et largement sur le niveau lexical, même si on l'enrichit d'une approche en n-grammes, que l'on traite les cooccurrences au niveau de "sacs de mots", et qu'on aura fait une incursion dans le niveau syntaxique avec l'analyse des dépendances. L'analyse de contenu traditionnelle favorise un niveau sémantique plus abstrait celui de thématiques, de catégories, éventuellement de descriptions stylistiques.

La philosophie du *machine learning* (ML) s'y prête volontiers et peut créer de nouvelles méthodes d'analyse, généralisant ce que l'on sait faire manuellement. Un logiciel tel que Tropes se prête à cet exercice en codant les termes associés dans un dictionnaire des catégories. Mais une telle approche devient difficile à mettre en œuvre quand le nombre de documents excède des centaines d'exemplaires.

Une approche alternative peut alors consister à entraîner un algorithme de ML à reconnaître de telles catégories. C'est une approche employée industriellement pour mesurer par exemple la toxicité d'un contenu (Aroyo et al. 2019), détecter le sarcasme (Xiong et al. 2019)Pr, deviner les mensonges (Afroz, Brennan, et Greenstadt 2012), niveau de langage, l'identification du style (Gomez Adorno et al. 2018) .

1. Le processus de modélisation

L'ensemble du processus de machine learning est résumé dans le schéma suivant.

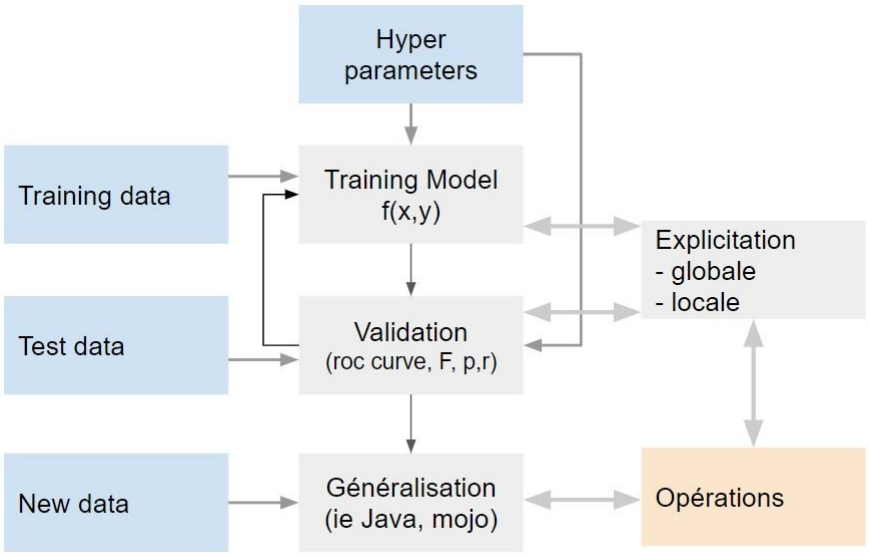


Figure ? : le processus de machine learning

- Les hyperparamètres sont les valeurs de réglage du modèle : le nombre d'itérations, les valeurs de *seed* (valeur aléatoire initiales), la solution initiale et les autres paramètres spécifiques des différents modèles testés.
- L'échantillon d'entraînement comprend les prédicteurs *x* (qui dans le cas du texte peuvent être les transformations du corpus) et l'annotation *y* que l'on veut généraliser. La taille des corpus, qui peut atteindre des milliards de mots, autorise des modèles à de nombreux paramètres, ils peuvent se compter en millions.
- Les modèles les plus couramment employés sont les modèles Random Forests, SVM, Naïves Bayes, et, plus sophistiqués, des réseaux de neurones et autres réseaux de neurones récursifs (*deep learning*). Faute de place nous ne les développons pas dans ce chapitre et renvoyant à (Goldberg 2017) pour des applications aux données textuelles ou (Hastie, Tibshirani, et Friedman 2009). Ils sont désormais d'usage standard.
- La phase de validation établit la performance du modèle en termes de taux de faux positifs (les fausses alertes) et de faux négatifs (les ratés) que l'on doit réduire simultanément. Différents critères de précision, de rappel, de F1 ou d'AUC sont employés dans la manière maintenant standardisée du Machine Learning. La validité de ces approches dépend largement de cette performance et de ses applications. Dans un certains nombre de tâches la précision atteinte dépasse celle d'un traitement humain.
- La généralisation consiste à intégrer le modèle dans un processus de *big data*, et dépasse l'horizon méthodologique concernant l'industrialisation des processus de plus en plus souvent assurée par une distribution du calcul via des Apis ce que proposent les grands opérateurs du cloud.

2. Une application à l'analyse des sentiments

Dans l'exemple 25 on présente la mise en œuvre d'un modèle extrêmement simple mais adapté au traitement du texte. C'est le modèle Naive Bayes qui s'appuie sur une idée simple et fondamentale : la probabilité qu'un texte possède un certain caractère dépend de la probabilité que ses mots soient associés à ce caractère. L'exemple le plus simple d'application est celui des anti spam qui nettoient nos boites mails.

On peut exprimer la probabilité qu'un mail entrant soit du spam, connaissant un mot particulier (par ex «Gratuit !») comme p(S/w), en termes bayésiens on obtient donc :

P(S/w)=p(W/C)*p(S)/p(w).

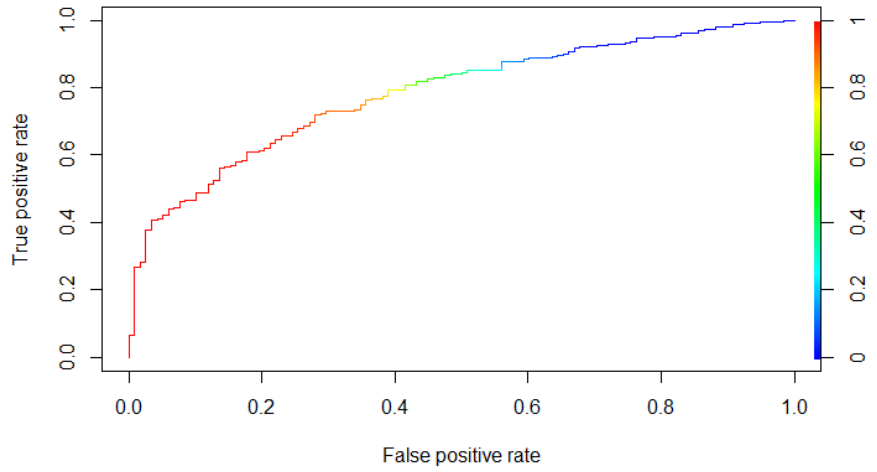
Autrement dit cette probabilité se calcule en connaissant la fréquence du mot parmi les spams, celle du spam parmi les courriers, et la fréquence relative du mot dans l'ensemble du corpus traité. Le modèle naive bayes généralise ce calcul quand un ensemble de mots est observé, il se réduit après quelques manipulations algébriques à :

$$\begin{aligned} p(\text{Spam} \mid w_1, \cdots, w_n) &\propto p(\text{Spam}) \prod_{i=1}^{n} p(w_i \mid \text{Spam}) \\ &= p(\text{Spam})\, p(w_1 \mid \text{Spam})\, p(w_2 \mid \text{Spam})\, p(w_3 \mid \text{Spam}) \cdots \end{aligned}$$

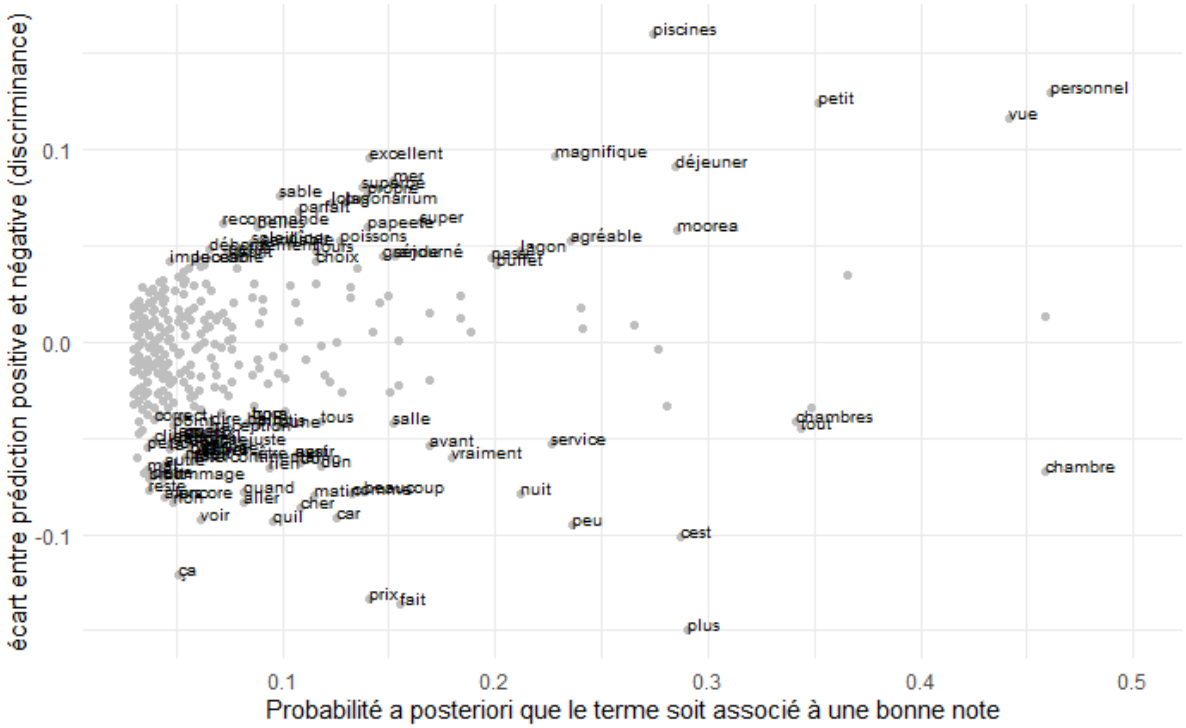
L'avantage du modèle est naturellement sa simplicité de calcul qui s'appuie sur la fréquence relative du spam et des mots dans ce corpus. On s'attend à ce qu'il soit d'autant meilleurs que le corpus est grand, il est flexible car à mesure que les spams sont tagués, le modèle peut être facilement recalculé. Il possède cependant une faiblesse en considérant a priori que les termes introduits dans l'analyse sont indépendant, en dépit de cette faiblesse il s'avère souvent efficace.(Eyheramendy, Lewis, et Madigan 2003)

Une fois le modèle testé et validé, il ne restera plus qu'à le généraliser à l'ensemble du corpus pour l'annoter complètement et séparer le grain de l'ivraie. Naturellement dans les applications la catégorie à classer peut prendre des formes multiples : distinguer les commentaires positifs des négatifs, distinguer les sujets de discussion, le caractère personnel ou impersonnel de l'expression… La principale difficulté reposera dans l'annotation nécessaire pour engager la phase d'apprentissage.

Exemple 25 : On ajuste pour un échantillon de 3700 commentaires d'un hôtel, un modèle naïve bayes pour prédire les meilleures notes. 60 % de l'échantillon sont utilisés pour calibrer le modèle, les autres 40 % pour le tester. La



précision est de 0,78 et l'AUC de 0,78. On reproduit la fonction ROC d'évaluation du modèle. Ce n'est pas un modèle extrêmement précis sauf à accepter une détection d'un fort taux de positifs (pour identifier 80 % de vrais positifs il va falloir accepter 40 % de faux négatifs.
Dans une optique de prédiction il faudrait nous tourner vers des modèles plus sophistiqués, mais dans un objectif d'analyse, ce modèle nous apprend des choses, en particulier il permet d'identifier les mots qui contribuent aux bonnes notes. Si on retrouve des adjectifs comme « magnifique » ou « agréable » qu'un détecteur de sentiment utilise certainement. D'autres qui n'ont pas forcément de valence, se retrouve dans ce contexte



comme étant des prédicteurs du sentiment positif : « piscines », « déjeuner » « personnel » ou du sentiment négatif : « chambre », « service » ou « prix ». Ils sont utilisés plus fréquemment quand l'expérience est positive ou négative.

1. Expliquer le modèle

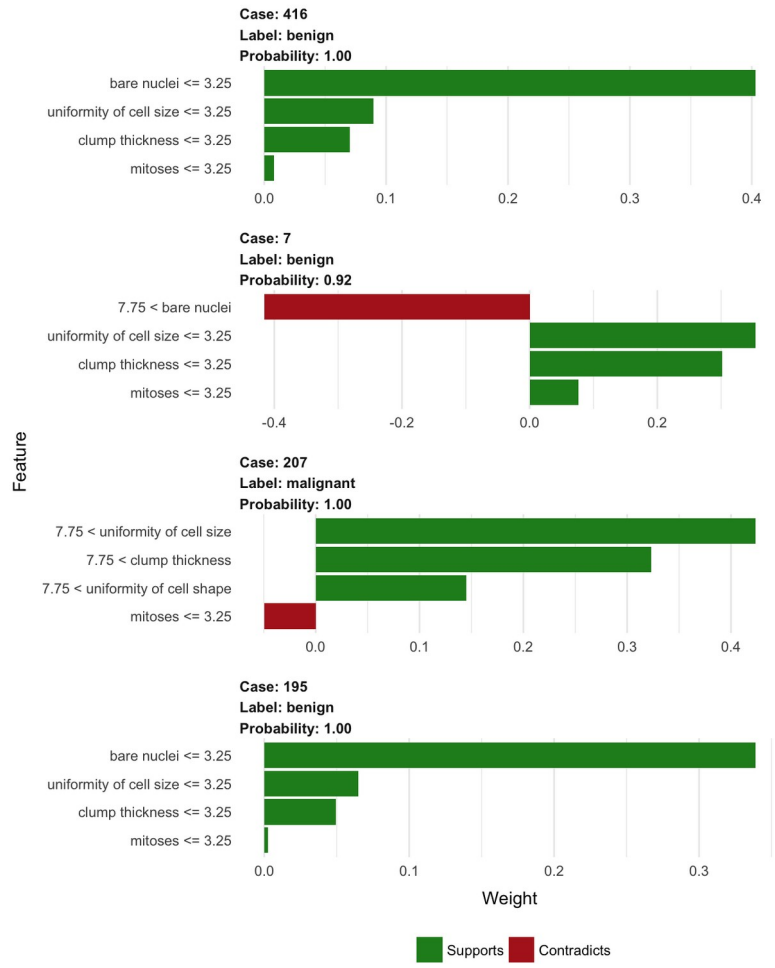
Deux points méritent un développement dans le cadre de l'analyse textuelle :

- La question de l'explicitation, car il est nécessaire de comprendre pourquoi l'algorithme fait telle ou telle prédiction et avec quelle fiabilité. C'est un domaine nouveau et actif du ML destiné à contrecarrer l'inconvénient du caractère de boîte noire de ces méthodes.
- La question du pré-apprentissage qui permet de pallier la faiblesse de la taille de certains corpus.

Dans la dernière étape il s'agit de comprendre comment le modèle calcule les probabilités d'avoir le caractère étudié pour une prédiction donnée. Il s'agit donc localement de comprendre comment les caractéristiques (*features*) expliquent la prédiction réalisée. C'est l'objet de la méthode Lime ou *Local Interpretable Model-agnostic Explanation*, développée par(Ribeiro, Singh, et Guestrin 2016)ribeir (Ribeiro, Singh, et Guestrin 2016) dont le principe général est relativement simple : utiliser un modèle local pour approximer de manière interprétable un modèle global.

Il consiste à identifier un certain nombre de cas voisins (permutations), et, sur cet ensemble (les observations sont pondérées en fonction de leur distance au point étudié), à estimer un modèle interprétable qui associe les prédicteurs à leurs prédictions (obtenues par le classificateur global qu'on cherche à évaluer). Un modèle interprétable est par exemple une régression Lasso(Tibshirani 2011) dont l'intérêt est d'obtenir des paramètres pour un nombre réduit de critères grâce à la normalisation (la somme des paramètres est contrainte à être inférieure à une valeur fixée).

Exemple 23 : Application de Lime à une analyse textuelle
Le modèle global identifie dans des dossiers médicaux les cas de cancer à partir de la mention d'un certain nombre de valeurs physiologiques.



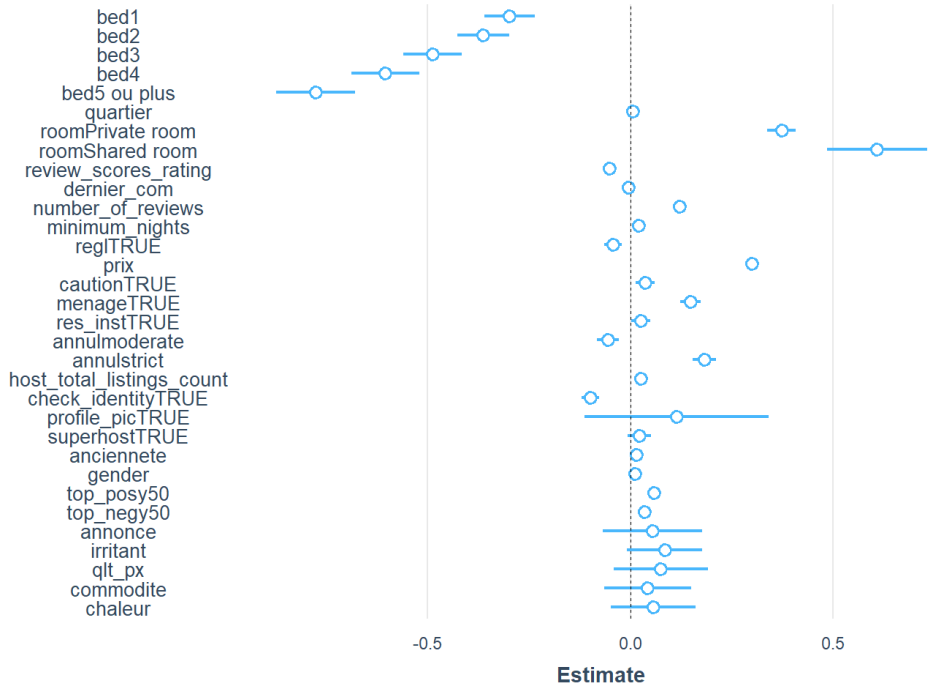
PROLONGEMENTS

Les techniques examinées dans les sections précédentes n'épuisent pas le travail de l'analyse textuelle, elles représentent une phase intermédiaire dans le processus d'analyse. Elles fournissent des matériaux qu'il s'agira d'introduire dans des modèles statistiques plus traditionnels (comparer deux groupes, analyser l'influence de variable exogènes…), mais aussi de représenter de manière stimulante la synthèse des données et des modèles, la visualisation est nécessaire pour la compréhension mais aussi de proposer des versions condensées du corpus de sa structure pour favoriser une approche qualitative des données.

Vers une économétrie textuelle

Un intérêt principal est de quantifier le texte pour introduire ces éléments dans des modèles explicatifs, par exemple pour caractériser les auteurs (compétences langagières, tonalité du discours, genre, catégorie sociale…) ou pour caractériser les contenus (thématiques, sentiments, lisibilité…).

Exemple 25 : Comprendre le taux de réservation
L'idée du modèle est d'ajouter aux caractéristiques du logement touristique (on s'aperçoit ainsi que la taille des appartements est déterminantes : plus c'est grand, plus c'est demandé), des variables construites à partir du texte des commentaires associés aux annonces : deux indice de sentiments (positif et négatif) et la composition en termes de sujets (*topics*) des 50 premiers commentaires associés aux logements. Lorsque la valeur du paramètre est positive, c'est que la variable considérée tend à prédire une plus grande disponibilité, lorsqu'elle est négative la disponibilité est moindre et donc le logement se loue plus facilement.



L'importance de la visualisation

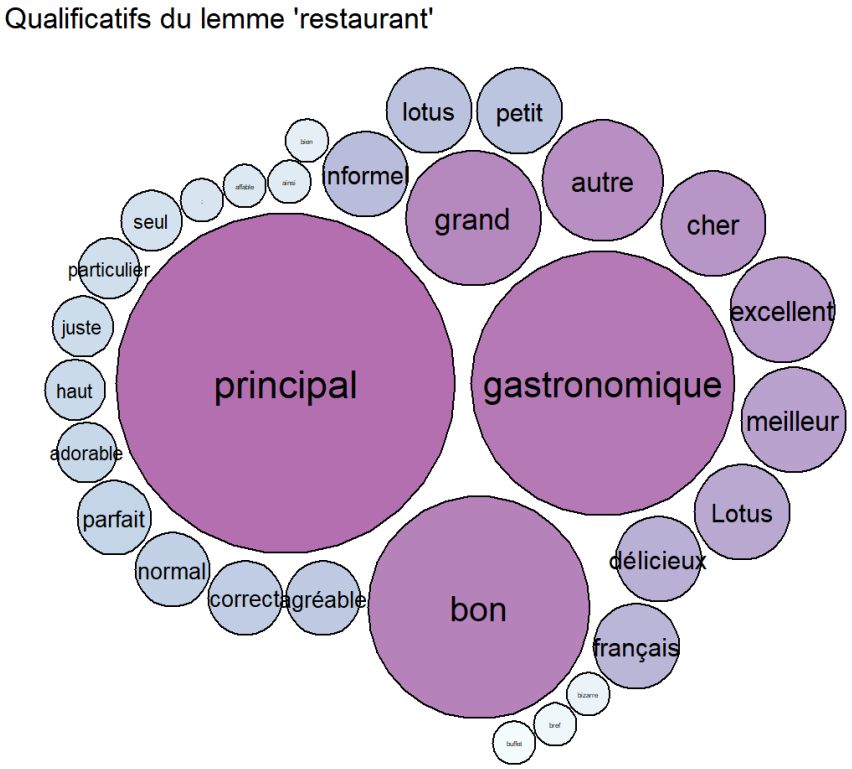
Le succès des méthodes dépend moins de leur qualité à rendre compte d'une réalité qu'à susciter de nouvelles intuitions et hypothèses.

Les données textuelles se condensent difficilement, même si on le veut représenter qu'une fraction des termes (par exemple dans une carte de similarité), ce sont d'emblée des centaines d'éléments qui doivent être représentés. Leur représentation est un enjeu majeur pour l'inférence et la communication.

Pour rendre intelligible de telles représentations, il peut être utile de zoomer sur certaines parties du graphique, ou d'en faire varier les projections s'il est tri-dimensionnel. Les méthodes interactives de représentation seront vite indispensables, elles permettent au lecteur de contrôler une partie des paramètres. Un outil tel que "Shiny" permet la navigation dynamique dans les cartes sémantiques.

Les éléments de grammaire des graphiques proposés par (Wilkinson 2005) et mise en œuvre par (Wickham 2010)r avec 'ggplot2' et ses dépendances, renouvellent la conception de représentations visuelles simples et lisibles. Le site https://www.r-graph-gallery.com/index.html donne une excellente idée des possibilités offertes.

Exemple 26 : Visualisation
Dans cet exemple, on utilise une méthode de *bubble chart* dont chaque bulle est relative aux effectifs d'un terme du corpus des commentaires d'un hôtel de Polynésie. Ces termes sont les adjectifs relatifs au lemme "restaurant" obtenue par l'analyse des dépendances syntaxiques. L'hôtel propose deux restaurants : le "principal" et un "gastronomique" – le Lotus, dont la faible fréquence résulte sans doute de sa faible identification comme qualificatif et plutôt comme nominatif. On dénombre peu d'hyperlatifs, la qualification est "bonne", plus rarement "excellent" ou "parfait". Certains termes sont comparatifs ("meilleur").



Vers un NLP génératifs

Au-delà de l'analyse garder en tête que les outils du *deep learning* peuvent être génératif. Une grande partie des recherches est consacrée à des tâches qui ne sont pas l'étude du discours mais à des tâches opérationnelles : comprendre une quête dans un moteur de recherche, répondre par un chat bot à une question, traduire un texte, et même en créer de nouveaux. Et l'ensemble des outils présentés n'ont pas forcément été conçus pour l'étude, mais pour exécuter de manière acceptable des tâches spécifiques en grande masse.

Elle donne incidemment une opportunité pour les méthodes de recherche, celle de réduire de grand corpus à un petit nombre de formes élémentaires. Des arguments, des séquences logiques, le mot et ses corrélations ont fait l'objet d'une très vaste recherche, la phrase et ses articulations y compris ses significations reste un enjeu majeur. Mais les techniques permettent de faire face à cet enjeu.

Deux outils sont à portée de main du chercheur en sciences de gestion. Il permettrait de coder certains concepts clés des discours managériaux : l'incertitude perçue en premier lieu, puisque c'est la variable clé, la compétition puisqu'elle est omniprésente. Mais on peut aussi envisager d'annoter l'indignation

- Extraction de triplets noms-verbes-objet n'a pas encore d'applications dans r, mais des solutions sont proposées ailleurs. Ils peuvent être représentés par des graphes bipartites comme dans l'exemple 2X.
- mais aussi les méthodes de résumés (faire des résumés de compte rendu d'entretien).
- Et les méthodes de Q&A.

Exemple	27	BERT:

Conclusion

On assiste depuis quelques années à un bouillonnement d'idées, de techniques, de ressources résultant dela convergence de plusieurs champs : linguistique, psychologie, informatique, et nourri par la production de corpus de grande taille. Cette créativité bénéficie aussi largement des lingua franca que sont r et python, dans lesquelles les bibliothèques d'algorithme se remplissent rapidement de nouvelles méthodes.

Nous n'avons pas développé les nouvelles perspectives offertes par le deep learning et des structures appropriées aux données séquentielles. Elles sont cependant l'avenir immédiat qui va offrir des solutions testées sur des corpus de plusieurs milliards de mots. Une démarche de plus en plus fréquente d'ailleurs consiste à entraîner des modèles à partir de modèles déjà entraînés sur des corpus génériques. On imagine d'ailleurs aller plus loin. À un certain niveau d'abstraction les modèles deviennent transférables d'une langue à l'autre. Dans la somme du discours il semble qu'il y ait assez de régularité pour passer d'une langue à l'autre. La langue change, les émotions restent. La langue de la chanson populaire le traduit dans ses formes variantes et son discours oscillant.

On restera cependant prudent. Mieux associer le discours aux actes, c'est le propre du numérique où on enregistre des déclarations, des commentaires et des actions en continu. Mais le sujet peut tromper, se taire, jeter un voile de fumée. S'il parle beaucoup et qu'on sait en connaître chaque parole, il sait aussi se taire. L'analyse du discours reste une analyse du discours. Même si le potentiel des techniques face à des corpus géants, dont l'avantage linguistique est de repérer des régularités rares. est vaste, on gardera à l'esprit que tout ne se verbalise pas, et qu'on ne dispose pas d'outil pour entendre ce qui ne se prononce pas. L'agence est présente dans le verbe guidant ses silences, ses nuances, ses stratégies scripturales.

Cependant quand il est abondant et qu'on peut l'associer au déroulement des événements, qu'on peut y associer des actes et enregistrer les changements d'environnement. Il devient une matière précieuse pour comprendre la causalité sociale. La nouveauté c'est que désormais la physique peut mémoriser les pensées et les actions et leur mémorisation. Dans le cloud reposent nos intentions, nos déclarations,

Mais il y a la langue. Comprendre l'autre nécessite d'entendre ce qu'il dit tout autant que ses modulations. On glissera sans doute d'un point de vue linguistique (lexical, syntaxique, sémantique, pragmatique, phonologique, prosodique) vers celui de la littérature si on prend aussi en compte les formes de composition. Le sens n'est pas seulement dans ce qu'on dit mais aussi dans le comment on le dit. Le style compte.

Un horizon prochain, sera l'analyse des genres du discours : l'avis, la réponse, le dialogue, le chat, le tweet. Dans chacun de ces genres il faudra identifier des styles : argumentatif, narratif, descriptif, récriminatoire, poétique, diplomatique, revendicatif. Il faudra apprécier aussi comment la combinaison des styles et de ce qu'on dit est capable de persuasion. D'autres options seront sans doute engagées, la langue et ses productions vont redevenir au centre de l'attention. Il va falloir apprendre à les lire massivement.

Références

Abdaoui, Amine, Jérôme Azé, Sandra Bringay, et Pascal Poncelet. 2017. « FEEL: A French Expanded Emotion Lexicon ». *Language Resources and Evaluation* 51 (3): 833–855. https://doi.org/10.1007/s10579-016-9364-5.

Afroz, Sadia, Michael Brennan, et Rachel Greenstadt. 2012. « Detecting Hoaxes, Frauds, and Deception in Writing Style Online ». In *2012 IEEE Symposium on Security and Privacy*, 461-75. San Francisco, CA, USA: IEEE. https://doi.org/10.1109/SP.2012.34.

Aggarwal, Charu C. 2016. *Recommender Systems*. Cham: Springer International Publishing. https://doi.org/10.1007/978-3-319-29659-3.

Anastasopoulos, L Jason, Tima T Moldogaziev, et Tyler A Scott. 2017. « Computational Text Analysis for Public Management Research: An Annotated Application to County Budgets », 47. https://doi.org/10.2139/ssrn.3178287.

Arnold, Taylor. 2017a. « A Tidy Data Model for Natural Language Processing Using CleanNLP ». *The R Journal* 9 (2): 248. https://doi.org/10.32614/RJ-2017-035.

———. 2017b. « A Tidy Data Model for Natural Language Processing Using CleanNLP ». *The R Journal* 9 (2): 248. https://doi.org/10.32614/RJ-2017-035.

Aroyo, Lora, Lucas Dixon, Nithum Thain, Olivia Redfield, et Rachel Rosen. 2019. « Crowdsourcing Subjective Tasks: The Case Study of Understanding Toxicity in Online Discussions ». In *Companion Proceedings of The 2019 World Wide Web Conference on - WWW '19*, 1100-1105. San Francisco, USA: ACM Press. https://doi.org/10.1145/3308560.3317083.

B., M., et George Kingsley Zipf. 1951. « Human Behavior and the Principle of Least Effort; An Introduction to Human Ecology ». *The American Journal of Psychology* 64 (1): 149. https://doi.org/10.2307/1418618.

Benoit, Kenneth, Kohei Watanabe, Haiyan Wang, Paul Nulty, Adam Obeng, Stefan Müller, et Akitaka Matsuo. 2018. « quanteda: An R package for the quantitative analysis of textual data ». *Journal of Open Source Software* 3 (30): 774. https://doi.org/10.21105/joss.00774.

Blei, David M., Andrew Y. Ng, et Michael I. Jordan. 2003. « Latent Dirichlet Allocation ». *J. Mach. Learn. Res.* 3 (mars): 993–1022.

Ding, Xiaowen, Bing Liu, et Philip S. Yu. 2008. « A Holistic Lexicon-based Approach to Opinion Mining ». In *Proceedings of the 2008 International Conference on Web Search and Data Mining*, 231–240. WSDM '08. New York, NY, USA: ACM. https://doi.org/10.1145/1341531.1341561.

Duval, Dominic, et François Pétry. 2016. « L'analyse Automatisée Du Ton Médiatique : Construction et Utilisation de La Version Française Du *Lexicoder Sentiment Dictionary* ». *Canadian Journal of Political Science* 49 (2): 197-220. https://doi.org/10.1017/S000842391600055X.

Eyheramendy, Susana, David D. Lewis, et David Madigan. 2003. *On the Naive Bayes Model for Text Categorization*.

Firth, J. R. 1957. « A synopsis of linguistic theory 1930-55. » *Studies in Linguistic Analysis (special volume of the Philological Society)* 1952-59: 1–32.

Fotopoulou, Aristea, et Nick Couldry. 2015. « Telling the Story of the Stories: Online Content Curation and Digital Engagement ». *Information, Communication & Society* 18 (2): 235-49. https://doi.org/10.1080/1369118X.2014.952317.

Fruchterman, Thomas M. J., et Edward M. Reingold. 1991. « Graph Drawing by Force-Directed Placement ». *Software: Practice and Experience* 21 (11): 1129–1164. https://doi.org/10.1002/spe.4380211102.

Goldberg, Yoav. 2017. « Neural Network Methods for Natural Language Processing ». *Synthesis Lectures on Human Language Technologies* 10 (1): 1-309. https://doi.org/10.2200/S00762ED1V01Y201703HLT037.

Gomez Adorno, Helena Montserrat, Germán Rios, Juan Pablo Posadas Durán, Grigori Sidorov, et Gerardo Sierra. 2018. « Stylometry-based Approach for Detecting Writing Style Changes in Literary Texts ». *Computación y Sistemas* 22 (1). https://doi.org/10.13053/cys-22-1-2882.

Guillaume Desagulier. 2018. « Word embeddings: the (very) basics," in Around the word, 25/04/2018 ». *https://corpling.hypotheses.org/495*, avril.

Hastie, Trevor, Robert Tibshirani, et Jerome Friedman. 2009. *The Elements of Statistical Learning*. Springer Series in Statistics. New York, NY: Springer New York. https://doi.org/10.1007/978-0-387-84858-7.

Humphreys, Ashlee, et Rebecca Jen-Hui Wang. 2018. « Automated Text Analysis for Consumer Research ». Édité par Eileen Fischer et Linda Price. *Journal of Consumer Research* 44 (6): 1274–1306. https://doi.org/10.1093/jcr/ucx104.

Jones, Karen Sparck. 1973. « Index Term Weighting ». *Information Storage and Retrieval* 9 (11): 619-33. https://doi.org/10.1016/0020-0271(73)90043-0.

Kobayashi, Vladimer B., Stefan T. Mol, Hannah A. Berkers, Gábor Kismihók, et Deanne N. Den Hartog. 2018. « Text Mining in Organizational Research ». *Organizational Research Methods* 21 (3): 733-65. https://doi.org/10.1177/1094428117722619.

Laer, Tom van, Jennifer Edson Escalas, Stephan Ludwig, et Ellis A van den Hende. 2018. « What Happens in Vegas Stays on TripAdvisor? A Theory and Technique to Understand Narrativity in Consumer Reviews ». Édité par Gita V Johar, J Jeffrey Inman, et Paul M Herr. *Journal of Consumer Research*, août, ucy067. https://doi.org/10.1093/jcr/ucy067.

Lock, Irina, et Peter Seele. 2015. « Quantitative Content Analysis as a Method for Business Ethics Research ». *Business Ethics: A European Review* 24 (juillet): S24-40. https://doi.org/10.1111/beer.12095.

Mabi, Clément. 2015. « La plate-forme « data.gouv.fr » ou l'open data à la française ». *Informations sociales* 191 (5): 52-59.

May, Andrew, et Tracy Ross. 2018. « The Design of Civic Technology: Factors That Influence Public Participation and Impact ». *Ergonomics* 61 (2): 214-25.

https://doi.org/10.1080/00140139.2017.1349939.
Meng, Xiao-Li. 2018. « Statistical Paradises and Paradoxes in Big Data (I): Law of Large Populations, Big Data Paradox, and the 2016 US Presidential Election ». *The Annals of Applied Statistics* 12 (2): 685-726. https://doi.org/10.1214/18-AOAS1161SF.
Mikolov, Tomas, Ilya Sutskever, Kai Chen, Greg S Corrado, et Jeff Dean. 2013. « Distributed Representations of Words and Phrases and Their Compositionality », 9.
Miller, George A. 1995. « WordNet: a lexical database for English ». *Communications of the ACM* 38 (11): 39-41. https://doi.org/10.1145/219717.219748.
Mohammad, Saif M., et Peter D. Turney. 2013. « Crowdsourcing a Word-Emotion Association Lexicon ». *Computational Intelligence* 29 (3): 436–465.
Nielsen, Finn Årup. 2011. « A New ANEW: Evaluation of a Word List for Sentiment Analysis in Microblogs. » In *#MSM*, édité par Matthew Rowe, Milan Stankovic, Aba-Sah Dadzie, et Mariann Hardey, 718:93-98. CEUR Workshop Proceedings. CEUR-WS.org. http://dblp.uni-trier.de/db/conf/msm/msm2011.html#Nielsen11.
Pennington, Jeffrey, Richard Socher, et Christopher Manning. 2014. « Glove: Global Vectors for Word Representation ». In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 1532-43. Doha, Qatar: Association for Computational Linguistics. https://doi.org/10.3115/v1/D14-1162.
Ratinaud, Pierre, et Pascal Marchand. 2015. « Des mondes lexicaux aux représentations sociales. Une première approche des thématiques dans les débats à l'Assemblée nationale (1998-2014) ». *Mots*, nº 108 (octobre): 57-77. https://doi.org/10.4000/mots.22006.
Ribeiro, Marco Tulio, Sameer Singh, et Carlos Guestrin. 2016. « "Why Should I Trust You?": Explaining the Predictions of Any Classifier ». *arXiv:1602.04938 [cs, stat]*, février. http://arxiv.org/abs/1602.04938.
Šuster, Simon. 2015. « An investigation into language complexity of World-of-Warcraft game-external texts ». *arXiv:1502.02655 [cs]*, février. http://arxiv.org/abs/1502.02655.
Tausczik, Yla R., et James W. Pennebaker. 2010. « The Psychological Meaning of Words: LIWC and Computerized Text Analysis Methods ». *Journal of Language and Social Psychology* 29 (1): 24-54. https://doi.org/10.1177/0261927X09351676.
Taylor, Ann, Mitchell Marcus, et Beatrice Santorini. 2003. « The Penn Treebank: An Overview ». In *Treebanks*, édité par Anne Abeillé, 20:5-22. Dordrecht: Springer Netherlands. https://doi.org/10.1007/978-94-010-0201-1_1.
Tibshirani, Robert. 2011. « Regression Shrinkage and Selection via the Lasso: A Retrospective: Regression Shrinkage and Selection via the Lasso ». *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 73 (3): 273–282. https://doi.org/10.1111/j.1467-9868.2011.00771.x.
Wickham, Hadley. 2010. « A Layered Grammar of Graphics ». *Journal of Computational and Graphical Statistics* 19 (1): 3–28. https://doi.org/10.1198/jcgs.2009.07098.
Wilkinson, Leland. 2005. *The Grammar of Graphics*. New York, NY: Springer Science+Business Media, Inc. http://0-dx.doi.org.fama.us.es/10.1007/0-387-28695-0.
Xiong, Tao, Peiran Zhang, Hongbo Zhu, et Yihui Yang. 2019. « Sarcasm Detection with Self-Matching Networks and Low-Rank Bilinear Pooling ». In *The World Wide Web Conference on - WWW '19*, 2115-24. San Francisco, CA, USA: ACM Press. https://doi.org/10.1145/3308558.3313735.